Mitigating Relation Completion Problem in Database Application
Suwarna Sanap¹, Madhuri Gaikwad², Amruta Ratnaparkhi³, Prashant Kolekar⁴, Sandeep Gore⁵
Student¹,²,³,⁴, Assistant professor⁵
Department of Computer Science
G.H. Raisoni College of Engineering & Management, Affiliated to SPPU, Pune, Maharashtra, India

Abstract:
Document retrieval and information retrieval is task of retrieving relevant documents according to user query. Given relation X, relation completion occurs at linking entity pairs between two entity set under the relation X. In Pattern based search, it may produce results that are unspecific or ambiguous when taken out of context i.e. it gives more results relevant as well as partially relevant. That means it produces a relation completion problem when extracting any relation. This implementation solves a major problem in context aware relation extraction method using text based information recovery, using the limited set of query terms given by user and extracting small set of relevant documents in a large document set. We propose search queries formulation for each query entity based on some information so that to detect target entity from the set of retrieved documents. Proposed method utilizes contextual information formed by word relationships contained in the document collection. And auxiliary information is given by context terms evaluated through the expression of a relation. If the retrieved document is less than top K documents KNN algorithm is applied on tree structure, to retrieve top K relevant documents. Document relevance is determined by comparing the similarity of search and document tree structure, the disclosed system attains greater search precision by retrieving relevant documents based on context and facilitating easier way search specification.

Keywords: Context aware relation extraction, Pattern based search, Relation completion, Rel term, Data mining, Information extraction, Big data.

1. INTRODUCTION
In this work, we identify Relation Completion is one recurring problem that is central to the success of the novel application such as information extraction and relation extraction [1]. This motivated novel frameworks that incorporate information extraction (IE) tasks such as named entity recognition (NER) and relation extraction (RE). Those frameworks have been used to enable some of the emerging data linking applications such as object rebuild and data enrichment. Main contribution in this project work is summarized as follows:
1) We propose context based document retrieval Using Taxonomic tree structure
2) We propose an integrated model to learn high quality context relation terms and expand methods that are based on term frequency.
3) We propose taxonomic tree based query formulation method which selects a small subset of search queries to be issued.
4) We propose a confidence aware method that estimates the confidence that a candidate target entity is correct one.
5) We propose K-nearest neighbor algorithm (KNN) to traverse to the brother nodes in taxonomic tree structure to fetch relevant documents when top K documents are not returned.

2. Background and CoRE Review
Relation completion is rapidly becoming one of the fundamental tasks underlying many of the emerging applications that capitalize on the opportunities provided by the abundance of big data (e.g., entity reconstruction [9], [13], data enrichment [5], [16], etc.) We formally define the relation completion task as follows:

2.1 Architecture
2.1.1 Pattern Based Search:
In this scheme to provide secure data storage three contributors are involved. These contributors are Data Multiple RelQueries each of which is based on the query entity a in conjunction with one of the patterns extracted by the PaRE method. In pattern based method we must know the more information about the relation between the two entities.
2.1.2 Real Term Search:

CoRE makes use of existing set of linked pairs towards learning Relation Expansion Terms (i.e., RelTerms) for any relation $R$. CoRE generates a set of Relation Queries (i.e., RelQueries) for each query entity based on the set of learned RelTerms.

There are two types of searching Method:

(i) learning a set of high-quality candidate RelTerms from each existing connect pair

(ii) Consolidating and removing those individual candidate sets into a minimal global set of RelTerms that are used in the formulation of RelQueries.

2.1.3 Context aware relation extraction:

This creation is based on our proposed CoRE, in which multiple RelQueries are present, each of which is based on the query entity $a$ in conjunction with several RelTerms extracted by the CoRE method (e.g., (“Jon” $\triangleright$ “Staff”), (“Jon” $\triangleright$ “Designation”), etc.). By using RelTerms, a limited number of documents are retrieved, in which some are RelDocs that contain the correct target entity. Context-based creation tries to strike a fine balance between a very strict RelQuery formulation (i.e., pattern based) and a very relaxed one (i.e., query-based). Towards this, CoRE make use of Rel Term towards a flexible query creation in which a RelQuery is created based on the query entity $a$ in conjunction with one or more RelTerms.

2.1.4 Relation Completion:

RelTerms from each of the existing individual linked pair, CoRE selects a set of general RelTerms from those candidates. The goal is to select a set of high-quality RelTerms for effective query formulation, and in turn accurate relation completion (i.e., finding target entities). In CoRE, this task takes place in two steps:

I. CoRE uses a local pruning strategy to eliminate the least effective RelTerms i.e. least relevant to the input entity,

II. CoRE uses a global selection strategy to choose the most effective RelTerms i.e. more relevant terms to the relation

2.2 Relation Creation

![Flowchart of Relation Creation](image)

1. Relation Creation: Create relation with word with respect to relations
   **Eg. Word:** college, **Relation:** College City, College state etc.

2. Staff Data: All Staff Data with respect to college

3. Department: department details with respect to colleges

4. Category: add word and insert categories with respect to that word

5. Place: Add Place and address details of colleges

6. Books: add books with their categories, authors and book names

7. Definition: create definitions

8. Search: search using context aware relations

9. Logout

3. LEARNING RELATION EXPANSION TERMS

The main aim of query driven document retrieval system is to find relevant documents based on input query which is given by users. Since the typical query from the user comprises only few keywords and existing techniques failed to discriminate between documents having only query terms and documents that are actually relevant, and it give more document it includes relevant as well as partially relevant. There are two categories of conventional search engines for unstructured data that is first one is keyword based in which documents are ranked based on the occurrence of input keywords provided by the user, and the second one is categorization based, where documents and information contain within the documents to be searched are pre classified. Context aware Relation Extraction i.e. CoRE method gives relevant documents from document sets than the existing pattern based method which we are using now a days, by recognizing and exploiting a particular context within the document.

3.1 Learning Relational Expansion Terms

Contextual document retrieval utilizes existing set of linked pairs of a relation towards learning relational expansion terms.

3.2 Learning Candidate Context Terms

Frequency based model is considered in selecting good expansion terms in the straight Query expansion models. Frequency: The context term is mentioned frequently across a
number of different documents that are relevant to given linked pair.

3.3 Tree Based QF model
To facilitate the specification of search queries and document comparison at retrieval time, user key words are compared against taxonomic tree structure computed using context database.

1) provide the input query as alpha
2) generate the possible term set beta
   \{e1, e5, e7\}
3) if CETs contains \{e1, e5, e7\}, the RelQuery will be e1+e5+e7+α
   \{if(rootnode==empty)
   \{then unexpanded query of alpha\}
   else \{newquery=newquery+newterm
   (new query=e1+alpha,e1+e2+alpha) submit real query to web page
   set nsr=top k result
   nsr indicate the number of search result \}

Situation 1: There are at least one candidate target entities whose confidence is higher than a threshold which is given. According To the Confidence-Aware Termination condition, we will exit the QF process. When several target entity candidates are found, only the one have the highest confidence will be taken as the target entity.

Situation 2: Otherwise, assume there are ND documents returned by the current Rel Query, and ND ≤ K, and then we have already gone through all returned documents without finding good candidates. We skip over all the descendant nodes under the node (such as e7 under e5), and the Rel Term in this node (e5) will be removed from CETs. Next, we move to its first brother node (e6) on the right. If there is no brother node on the right, we skip to the first un-traversed brother node of its parent node on the right, and the RelTerm in its parent node will also be removed

Situation 3: Otherwise, we have ND > K (most of the time ND >> K), for efficiency issues, we won’t go through the documents after top-K. Instead, we move to the first of its un-traversed child node (e7) without touching CETs. However, if there is no un-traversed child, we skip to the next un-traversed brother of its father node. Meanwhile, we remove the RelTerms in this node and its parent node. Note that in situation 2, although it cannot be guaranteed that the SubCoverTerms of a failed RelTerm will also fail

3.4 Confidence aware termination
This algo or technique is used for which estimates the confidence that a candidate target entity is (generated possible real term query) is the correct target entity.

The following query types become important:

- Threshold: Return only result tuples with confidence above a threshold \( \tau \).
- Top-k: Return k result tuples with the highest confidence values.
- Sorted: Return result tuples sorted by confidence.
- Sorted-Threshold: Return result tuples with confidence above a threshold \( \tau \), sorted by confidence.

Our algorithms use the following operations:

- load(S, os1, os2): Loads into memory tuples from relation S (outer) starting at offset os1 and ending at os2. It may construct a hash table on the joining attributes to enable efficient look ups in mem when R is scanned, if the join condition can benefit, but doing so does not affect retrieval cost of document). The cost of a load is os2 – os1.

- scan(R, or1, or2): Scans relation R (inner) starting at offset or1 and ending at or2. While scanning, the join condition is evaluated on each scanned tuple of R and all tuples of S residing in mem (possibly using the hash table if the join condition allows it). The cost of a scan is or2 – or1.

- conf(R, S, or, os): Returns the result of the combining function f on the tuple from R at offset or and tuple of S at offset os. A negative value (flag) is returned if the offset for either relation exceeds the size of the relation. This function is always called such that the data required is in memory.

- explore(R, S, or1, or2, os1, os2): Combines load(S, os1, os2) followed by scan(R, or1, or2).

Algorithm 3 Top-k

\[ oR[e] \leftarrow 0; oS[e] \leftarrow 0 \]
\[ \text{confidence[e]} \leftarrow 1 \]
\[ Q \leftarrow e \]
\[ \text{while 1 do} \]
\[ i \leftarrow \text{ExtractMax}(Q) \]
\[ \text{if confidence[i]} \leq \text{Bottom}(K) \text{then} \]
\[ \text{break} \]
\[ or1 \leftarrow oR[i] \]
\[ os1 \leftarrow oS[i] \]
\[ \text{explore}(R, S, or1, \min(L + or1, or2), os1, \min(M + os1, os2)) \]
\[ \{os2 \leftarrow \minp \text{conf}(R, S, or1, p) \leq \text{Bottom}(K)\} \]
\[ \{or2 \leftarrow \minp \text{conf}(R, S, p, os1) \leq \text{Bottom}(K)\} \]
\[ oR[r] \leftarrow or1 + L; \]
\[ oS[r] \leftarrow os1 \]
\[ \text{confidence[r]} \leftarrow \text{conf}(R, S, oR[r], oS[r]) \]
\[ Q \leftarrow r \]
\[ oR[u] \leftarrow or1; oS[u] \leftarrow os1 + M \]
\[ \text{confidence[u]} \leftarrow \text{conf}(R, S, oR[u], oS[u]) \]
\[ Q \leftarrow u \]

4. ACKNOWLEDGMENTS

We hereby wish to take this opportunity to express our gratitude to our teachers and friends and all who have helped toward the completion of our project. We take a great honor in presenting this Project Report to our Director, Prof. Amol Pote. We also like to give thanks to our H.O.D. Dr. K. S. Wagh for helping us and guiding us throughout our endeavor. We are very grateful to our Guide Prof. Sandeep Gore for his guidance through the project. We are very grateful to our teaching staff for guiding us all over the duration of the degree. They were very helpful to us, as and when we required their help. We are also very grateful to non-teaching staff to help us in the laboratory in various ways. We would also like to extend our gratitude to those friends whose knowledge and time helped us in many different ways.
5. CONCLUSION

In this work, we identify relation completion as one recurring problem that is central to the success of novel big data applications when extracting the relation of input term. We then propose a Context-Aware Relation Extraction method, which is particularly designed for the Relation Completion i.e. RC task. The experimental results based on several real-world web data collections and bid data applications shows that CoRE could reach more than 40 percent higher accuracy than a Pattern-based method in the context of RC. As future work, we will further study the RC problem under the many-to-many mapping, and investigate techniques for maintaining the high precision and recall achieved under the many-to-one case.

6. REFERENCES


[8]. Yankaj Lin, Zhiyuan Liu, Maosong Suni,Yang Liu and Xuan Zhu, “ Learning Entity and Relation Embeddeddings for Knowledge Graph Completion” In proceedings of AAAI’15 conference, feb 2015


