Query Expansion based on Word Embeddings and Ontologies for Efficient Information Retrieval

Namrata Rastogi[1], Parul Verma[2]
Amity Institute of Information Technology
Amity University Uttar Pradesh
Lucknow, India

Pankaj Kumar[3]
Dept. of Computer Science
Sri Ram Swaroop College of Engineering and Management
Lucknow, India

Abstract—**Information retrieval has been an ever-going process for end users to fetch relevant data at one go. The problem intensifies more with unstructured data in a semantic web environment. It is also a promising area for researchers to dive in and refine it from time to time. Expanding the user query and reformulating it is one probable solution to increase the efficiency of the information retrieval system. In this paper we propose "WeOnto", a novel two-level query expansion algorithm that utilizes the combination of web ontologies and word embeddings for similarity calculation. In the first level, the Real estate Ontology (REO) is created using Protégé and Sparql queries are passed to retrieve probable semantic words from the given ontology for each inputted user query. The first level gave significant results and improved the information retrieval by 18%. The second level of algorithm uses word embedding enhanced with the domain knowledge that helps to retrieve similar meaningful words based on cosine similarity for the same user query. Word embeddings are implemented using Word2Vec method that follows two architectures namely CBOW or Skip Gram. Most similar semantic words are retrieved using the CBOW word embeddings method in the proposed algorithm and concatenated with the semantic keywords generated from the real estate ontology to form a powerful reformulated query that gives promising relevant results. Finally, two topmost words as per their similarity index are taken to reformulate the original user query. Experimental results depict that proposed algorithm has given distinct results and has showcased significant improvement of 93% over the initial user query.**

Keywords—CBOW; Information retrieval; ontology; query reformulation; semantic web; skip gram; word embeddings; word2vec

I. Introduction

Internet is a deep ocean of information and efficient information retrieval has been a constant desire of users. Researchers have been continually working towards achieving this goal with various methodologies and algorithms being designed to give easy and quick access of information to the intended users. The main aim has been to work at the basic level and frame the user query such that the expanded query gives better results with increased precision.

Traditional IR methods were based on TF-IDF, Boolean, vector space models (VSM) or BM25 methods based on document frequency to solve the problem. But they all suffered from word mismatch issues called lexical gap problem [1] while at times the queries were not formulated correctly or were having ambiguous words that led to poor retrieval. This leads to following problems that needs to be catered eventually:

- With ever-growing data over Internet, improving the efficiency of information retrieval system has always been an issue.
- How good a user query be formulated such that it increases the efficiency of retrieved web results.
- How to measure the effectiveness of the user query formulated that retrieves required relevant results.

Thus, the research objective is to increase the efficiency of information retrieval systems and focusing on query expansion such that the performance evaluation of reformulated user queries gives effective desired web results.

Finally, our research work focusses on the problem of information retrieval and low web results and proposes WeOnto algorithm, a novel algorithm that works on increasing the efficiency of information retrieval by incorporating the latest concept of word embeddings combined with web ontologies in a semantic web environment. The algorithm suggests a solution of reformulating the user query by expanding the user query with most similar new words, thereby giving better retrieved results.

Such expanded queries include the original query and some additional keywords that are found relevant to the given query keywords. These additional keywords are derived using the concept of Word embeddings amalgamated with ontologies both help to extract the semantics of the given query words.

Web ontologies are stored in the form of triples, i.e., subject, object and predicate having the entire meaning or relation between the subject and object explained within the triple. The word embeddings on the other hand take the word with respect to its meaning from the surrounding context and give us most similar words based on embeddings that store the relations in the form of vectors calculate cosine similarity to draw the appropriate results.

Word embeddings is method from natural language processing that has gradually found its application in information retrieval also [2]. Pre trained word embeddings are applied on the user corpus to retrieve most similar word for the query words such that the given user query be expanded to give efficient information retrieval. Word2Vec, GloVe, FasText, Bert, Elmo are few methods that could be incorporated to do above mentioned tasks.

In this paper, Section II explains the background related work while Section III talks about method and material used and comprises of the description of the proposed algorithm, “WeOnto” used for query expansion to increase the efficiency of information retrieval. Section IV describes the result and discussion along with the analysis and result of the experiment done on user defined corpus. Section V is the conclusion.

II. Related Work

Information retrieval has always been a topic of concern for researchers worldwide and many experiments and methodologies have been devised to increase its efficiency from time to time. We even have traditional IR models like Boolean model, vector space (VSM) model, probability-based models, and fuzzy set models [3]. These models enhanced the workability of IR systems but still with ever growing information over the Internet, the need to improvise the efficiency continues. The keyword matching approach could not do better around problems like polysemy where semantics of the words was required instead of syntactical approach.

So, recent researchers evolved methods that focus more on semantics and the meaning of words based on the context used. For such purposes, a natural language processing feature called Word embeddings [4] for the purpose of information retrieval has come as a probable solution. Word2vec is a deep learning method under NLP that takes word embeddings with respect to the context learned from the given corpus and gives most similar words as output. Siriguleng [5] in the paper also used word2vec and LDA topic model to expand Mongolian query and improve retrieval. Even B. Wang, et.al. in their work had discussed about experimental results [6] they had in using six embedding models. They compared these models but could not find one universal method that would cater all possibilities. On the other hand, B. Mansurov and A. Mansurov [7] depicts the use of word embeddings on Uzbek language and used it to get semantic similar words. Farhan et.al also talks about taking top relevant results and calculating the average vector values using word embeddings in a deep neural network and improvise the IR system to an extent [8]. Various researchers have recently understood the power of using ontologies with word embeddings and have showcased their effectiveness in their works; some of them have been put here. WE-based Arabic IR models also use wordnet and embeddings and depict comparisons of working after incorporating embeddings as in [9]. QSST, a Quranic searching tool based on word embeddings gave a high performance with an average precision of 91.95% [10]. Jin Ren *et. al.* in his paper [11] also explained about the effective results obtained on the use of predicate expression related to ontology and combining it with word embeddings. The work of Jayawardana, *et. al.* also describes the use of word embeddings on semi-supervised ontology population [12]. Lastra-Díaz *et. al.* in their work [13] stated that taking an average of two models i.e., Word embedding models and ontology measures in an experimental survey gave better results.

A. Word Embeddings

Word embeddings are unsupervised learning applications that also talk about transfer learning as it is incorporated in the given user corpus. Embeddings can be character level or word level [14]. The word level embeddings use word2vec method where the basic construct of embeddings is converting words into vectors and then mathematically apply relations on them based on the corpus being used. The vectors having similarity are closer to each other and have similar values. Their threshold value is mostly greater than 0.6. The closer it is to 1, the higher the similarity index is considered and thus two vectors or words are considered most similar.

Word2vec is a deep learning method under NLP that takes word embeddings with respect to the context learned from the given corpus and gives most similar words as output. The similarity is calculated using Cosine similarity [15] such that:

$$\text{Cos}\ (\theta) = \frac{\mathbf{A * B}}{|\boldsymbol{A}||\boldsymbol{B}|} \qquad (1)$$

The similarity value of the vectors ranges from -1 to +1. The Gensim library in Python language gives all capabilities of running this model and check the output. This model talks about different vector dimension and window size.

Word2Vec model is further divided into two architectures: Continuous Bag of Words (CBOW) and Skip-Gram (see Fig. 1) used to calculate vectors in their own way and giving different results but closely like each other.

CBOW architecture projects ‘Current Word’ based on inputted context words whereas Skip-Gram works vice versa, i.e., it takes current word as input and gives contextual word before and after the current word [15].

The window size plays an important role in capturing the context of the corpus and giving similar words as output. In Fig. 1, window size =2 where W(t) is the target word while t-2, t-1, t+1, and t+2 are the neighboring words that form the contextual window because of which the meaning is understood. The more-closer words, the better they are related to each other.

All the researchers have ultimately tried to incorporate various ways of implementing ontologies or word embeddings method to achieve efficient information retrieval.

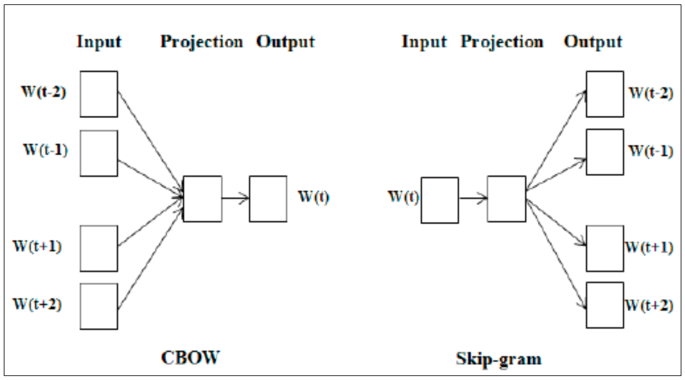


Fig. 1. Word2Vec (CBOW and Skip-Gram) Model Architecture [16]**.**

Few of them have achieved the precision of 91% using their own methods. Our proposed algorithm "WeOnto" depicts the power of ontologies along with word embeddings that doubly work on semantics of keywords rather than pattern matching alone and show an effective information retrieval system having a 93% precision.

The user query in the proposed algorithm goes through series of steps that works on the semantics by fetching most similar words and reformulating the user query such that it improves the efficiency of information retrieved during the web search.

The next section gives a detailed description of WeOnto algorithm: a novel query optimization approach that provides users with more meaningful similar words that upon implementation improves retrieval results.

III. Method and Material

With information explosion over the Internet, better information retrieval systems are always in demand. To increase its efficiency, query reformulation is one of the probable solutions.

For this purpose, we need to expand our query by preprocessing it first such that the stop words are removed and then we gather similar terms related to the major keywords left. The query after expansion will now have two set of words: i) keywords from query, ii) addition of new words.

A. Proposed Algorithm

Mathematically, let us suppose a query Q has n terms, Q = $\{t_1, t_2, t_3, \ldots t_n\}$. The reformulated query Q+ will have two set of words: i) keywords from expanded query Q' = {Q – ST} where ST is the list of stop words; ii) addition of new terms T' = $\{t_1{}'_{,}\ t_2, \ldots\ldots t_m'\}$. The reformulated query after expansion will look like [17]:

$$Q+ = Q' \cup T' \qquad (2)$$

$$= \{t_1, t_2, t_3, \ldots., t_{n,}, t_1{}'_{,} t_2, \ldots\ldots t_m'\}.$$

The question arises how to get these new terms.

"WeOnto" is the proposed algorithm that finds an answer in the form of applying a combination of Ontologies and word embeddings on the user query and reformulates it into a new query which will be more suitable in context to the domain and aims to give more relevant results with increased precision.

As per the WeOnto algorithm, there is an input user query Q in step 1 (see Fig. 2) that will be reformulated such that the expanded query Q' at the end of algorithm is suitable enough to retrieve more relevant web documents and has better precision.

To expand the given user query Q, the query is sent for pre-processing which includes processes like removal of stop words, lower their case and tokenization and q[] = {t1, t2, t3…………, t_n } is obtained. Here, q[] is the query after preprocessing having list of tokens $\{t_1,t_2,t_3\ldots t_n\}$.

```
Input Query: Q, Word2Vec Model: M

Output expanded Query: Q'

Step 1: Get User input query, Q and apply Pre-Processing.
        q [] = preprocess(Q) s.t. q ∈ {t1, t2, t3…………,
        $t_n$ } where t = tokens generated after
        preprocessing of Q.
Step 2A: q [] passed to Real Estate Ontology (REO) to
retrieve synsets. i.e., ∀ (t) ∈ q; ∃ (s) ∈ Ontology 'O'.
Step 2B: For each token '$t_i$'
        If $t_i = s_i$ then
 SW [] = add ($s_i$), SW is semantic words
 Else
        If $t_i \neq s_i$ then
 SW [] = add ($t_i$)
        End for
Step 3A: Num_tokens = len (q)
        For each token, ti,
        Sim_list [] = most similar vectors retrieved from
        Model 'M'.
Step 3B:
Threshold (th) = $\sum_{i=1}^{num\_tokens} sim\_list[\,] \Big/ num\_tokens$

        i.e., Calculate threshold Value = Average of
Vector values retrieved for 't'.
Step 3C: For each token '$t_i$'
        If sim_list[] > th then
 Act_sim_words = add (sim_list[])
        End for
Step 4: For each token '$t_i$'
         MSW[] = act_sim_words ∪ SW
        i.e., Combine both lists and retrieve two most
        suitable words from MSW[] to be used in
        expanded query, Q'.
Step 5: new words from MSW[] to initial query tokens, q[]
and get final expanded query,
        Q' = Q + { q[] ∪ MSW [] }
Step 6: Retrieve documents using the new query Q'.
```

Fig. 2. "WeOnto" Proposed Algorithm.

These tokens, t_i , i = 1…n are sent for a Two-level process of query expansion which can be seen in the algorithm. In the first level as shown in step 2, tokens t_i are passed to real estate ontology (REO) that was created to store the legal glossary terms used in case of real estate documentation during buying and selling of real estate properties [18]. The ontology was created using the WordNet vocabulary to capture all the syntactical and semantics of the English language as well as Legal terminology used for query reformulation. Sparql queries are issued in background that fetch semantically enriched keywords or synsets for each token t_i. For every token, t_i, if its synset exists, then it gets added to the semantic words list SW[], else the token itself gets added to SW[] giving us the list of semantic words fetched from REO as seen in Fig. 3.

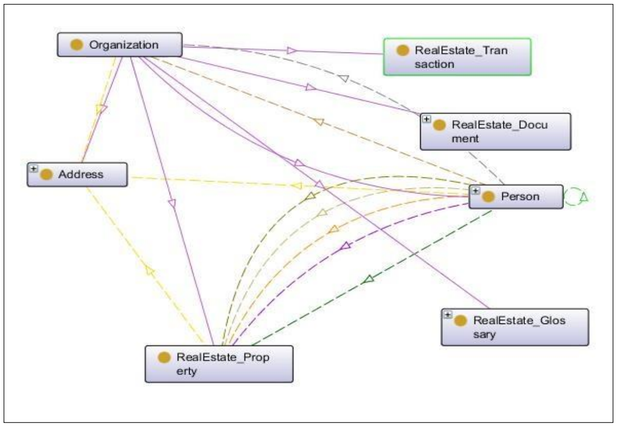


Fig. 3. Real Estate Ontology [18].

In the second level as depicted in step 3, a shallow learning NLP technique, Word2Vec model M that is using Continuous bag of words (CBOW) method to learn from the given corpus C. The corpus is made by scraping 2000 web documents related to real estate legal documentation domain. This word embedding model M has converted 1.05 million words into 12.5 thousand vectors using which most similar words would be derived for the same set of tokens, t_i.

These similar words are retrieved by calculating Cosine similarity for each pair and stored in sim_list[] list.

A threshold value is calculated as per step 3B from Fig. 2 to fetch the top k most similar words by taking average of the similarity index obtained for every token. If the similarity index is higher than this calculated threshold value, then such words are put into consideration and transferred to the actual similar words list, act_sim_words[].

Step 4 of WeOnto algorithm shows a union of above two lists obtained after step 2 & 3, i.e., SW[] and act_sim_words[] are combined on the basis of cosine similarity calculated for each pair of words and two most similar and suitable words are finally retrieved and stored to form the most suitable semantically enriched similar words, MSW[].

Ontology at first level is first incorporated to find the semantically enriched words for the tokens. Then word embeddings are also applied at second level to understand the context of real estate related queries with the help of the knowledge model that has learnt from the user defined corpus.

Step 5 of proposed algorithm shows the final step of union of original tokens from user query to most suitable semantically enriched similar words as in Eq. (2), MSW [], i.e., Q' = q[] ∪ MSW[]. Here, we need to remember that Q' will hold unique words only.

Hence, Q' becomes the final reformulated query that is deduced as the user query was expanded after applying the proposed algorithm where a list of semantic words retrieved from an ontology is concatenated to the list of words obtained from the word embeddings-based NLP model showing most similar words based on cosine similarity values.

The increase in the performance of information retrieval systems is calculated as defined in [19]:

$$Improvement = \frac{Reformulated\ Result - Baseline\ Result}{Baseline\ Result} \qquad (3)$$

This proposed novel algorithm is again tested, and it gives promising results showing a remarkable increase in the efficiency of IR system by incorporating a methodology that uses both ontology and word embeddings from NLP.

IV. Results and Discussion

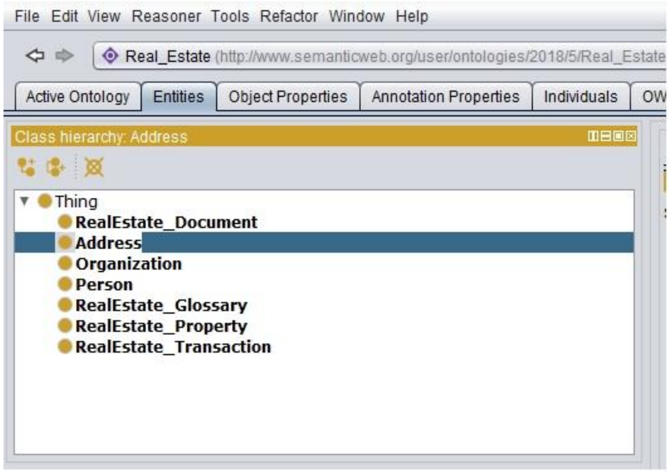
A. Experiment Setup

The proposed algorithm, "WeOnto" has a two-level procedure where the first level deals with the use of real estate ontology (REO) as defined in [20]. Real estate ontology has been created for a domain of real estate related legal documentation and has a glossary of legal terminology created using Wordnet Dictionary as seen in Fig. 4. The first level uses REO to retrieve semantically enriched keywords for the given user query and improved the reformulated query by 18%. However, the second level of algorithm is designed to further improve the information retrieval system and get more relevant results.

Hence, Step 2 of the WeOnto algorithm talks about the second level of the algorithm with the generation of similar words using word embeddings of natural language processing.

Word2Vec model of word embeddings is used with the aim that it will first train the model on real estate related dataset having data from 2000 web documents that were either government based or related to legal or real estate buying and selling. The implementation is done in Python language where its Gensim library was used to train the model that contains word vectors for a vocabulary of 12,462 words trained on around 1.05 million words from the user corpus and then apply various methods from it to derive similarity values.

Various parameters were set while training the model using Gensim library in python. Some of them like vector size = 100, initial learning rate, alpha = 0.025, window size = 5 which means two context words taken before and after the target word. Also, min_count = 1 which means that words having frequency <1 were avoided and lastly sg = 0 for CBOW and 1 for skip-gram method to be used.


File Edit View Reasoner Tools Refactor Window Help
Real_Estate (http://www.semanticweb.org/user/ontologies/2018/5/Real_Estate
Active Ontology
Entities
Object Properties
Annotation Properties
Individuals
OW
Class hierarchy: Address

Thing
- RealEstate_Document
- Address
- Organization
- Person
- RealEstate_Glossary
- RealEstate_Property
- RealEstate_Transaction

Fig. 4. Glossary Entity of Real Estate Ontology [19].

After the model has learnt and created vectors or embeddings from the corpus, the model is loaded to fetch similar words using cosine similarity (see eq. 1) for a given user query. The high cosine similarity between two words shows that the words are semantically similar and accordingly converted to vectors and are geometrically similar in the Euclidean space as well.

The ‘most_similar’ method returns the word vectors based on similarity for every token in the user query. To find the similarity between specific two words, i.e., to find similarity between user query and synsets retrieved from REO, ‘model.similarity()’ method from genism library in python is used and all values are stored in final_list.

Once the training is done, the test set includes 50 random user queries on which the entire algorithm is applied step wise. The result generated at each step is stored in Fig. 5 where every column defines a sub step of the algorithm.

Column No. 1 depicts the initial user query, Q. The query is pre-processed and converted into set of tokens, Q1 as shown in column 2 in Fig. 5 above. These tokens are passed to the real estate ontology (REO) and Synsets (named as set A) are retrieved for each token as in column 3.

The second step of algorithm talks about tokens being sent to Word2Vec model that produces most similar words (named as set B) as stored in column 4. Column 5 depicts the union of set A and B along with cosine similarity calculated for all paired vectors.

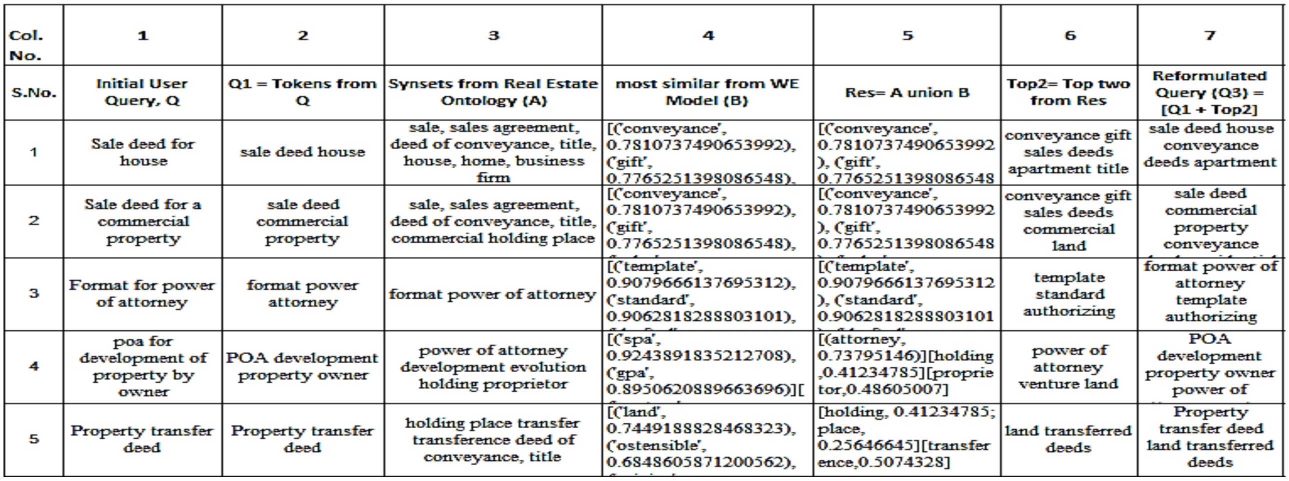
Column 6 holds the topmost two best words that are deduced using threshold value. Threshold value is first calculated taking the average of N vectors retrieved in column 4. If the similarity is greater than the average threshold value which keeps on changing with respect to every pair of word vector, then such words are counted as the best and stored in column 5.

Hence column 5 has the topmost words that has the highest cosine similarity. Column 6 shows the best two words derived for each token that will be added to the final expanded query.

Column 7 depicts the final expanded query, Q3 that holds the tokens after pre-processing and topmost two similar contextual words retrieved after implementation of the algorithm. This Q3 query is finally tested on test bed, www.google.com to retrieve the most relevant web documents against the initial user query.

B. Result Discussion

Table I shows the number of relevant documents retrieved at three levels i.e., at the initial query Q1, then Q2 is the query transformed by applying real estate ontology (REO) only and final expanded query, Q3 that shows implementation of combination of REO and word2vec method used in word embeddings.



Col. No.	1	2	3	4	5	6	7
S.No.	Initial User Query, Q	Q1 = Tokens from Q	Synsets from Real Estate Ontology (A)	most similar from WE Model (B)	Res= A union B	Top2= Top two from Res	Reformulated Query (Q3) = [Q1 + Top2]
1	Sale deed for house	sale deed house	sale, sales agreement, deed of conveyance, title, house, home, business firm	[('conveyance', 0.7810737490653992), ('gift', 0.776525139808654 8),	[('conveyance', 0.7810737490653992), ('gift', 0.776525139808654 8	conveyance gift sales deeds apartment title	sale deed house conveyance deeds apartment
2	Sale deed for a commercial property	sale deed commercial property	sale, sales agreement, deed of conveyance, title, commercial holding place	[('conveyance', 0.7810737490653992), ('gift', 0.776525139808654 8),	[('conveyance', 0.7810737490653992), ('gift', 0.776525139808654 8	conveyance gift sales deeds commercial land	sale deed commercial property conveyance
3	Format for power of attorney	format power attorney	format power of attorney	[('template', 0.9079666137695312), ('standard', 0.9062818288803101),	[('template', 0.9079666137695312), ('standard', 0.9062818288803101	template standard authorizing	format power of attorney template authorizing
4	poa for development of property by owner	POA development property owner	power of attorney development evolution holding proprietor	[('spa', 0.9243891835212708), ('gpa', 0.8950620889663696)][	[(attorney, 0.73795146)][holding ,0.41234785][proprie tor,0.48605007]	power of attorney venture land	POA development property owner power of
5	Property transfer deed	Property transfer deed	holding place transfer transference deed of conveyance, title	[('land', 0.7449188828468323), ('ostensible', 0.6848605871200562),	[holding, 0.41234785; place, 0.25646645][transfer ence,0.5074328]	land transferred deeds	Property transfer deed land transferred deeds

Fig. 5. Working of Proposed Algorithm.

TABLE I. NO. OF RELEVANT DOCUMENTS AND AVERAGE PRECISION

Query No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
No. of weblinks-Baseline query, Q1	7	7	23	8	1	10	0	15	0	1	0	6	8	2	5
No. of weblinks, Post Ontology, Q2	8	7	15	10	2	10	0	15	6	3	3	8	13	5	11
No. of weblinks- Post Word embeddings, Q3	7	7	20	5	8	5	1	8	8	9	7	22	17	10	6
Avg. Precision for Baseline query(Q1)	0.27	0.31	0.77	0.45	0.03	0.61	0	0.83	0	0.05	0	0.36	0.79	0.21	0.35
Avg. Precision for Post Ontology Query(Q2)	0.73	0.61	0.69	0.51	0.17	0.67	0	0.72	0.39	0.44	0.17	0.38	0.55	0.54	0.44
Avg. Precision for post Word Embeddings Query(Q3)	0.8	0.82	0.98	0.86	0.78	0.82	1	0.7	0.7	0.71	0.67	0.76	0.83	0.93	0.62

Table I also depicts the average precision of each query calculated at baseline, ontology, and embeddings level.

The graph in Fig. 6 is showing the average precision of getting relevant documents for given 50 queries.

Average precision after implementing the complete algorithm gives a substantially higher precision values for post word embeddings queries, Q3 as compared to its baseline, Q1 queries (see Fig. 6).

Another metric, Precision at 10 (P@10) is also used for performance evaluation of WeOnto algorithm and information retrieval at large. Table II depicts a sample of P@10 computed for all 50 queries. P@10 gives the number of relevant documents from the top 10 retrieved documents.

Fig. 7 shows the precision at 10 (P@10) metric of 50 queries together. This metric is used for performance evaluation of information retrieval systems. Here, values of P@10 have increased considerably for every query after implementation of word embeddings as compared to the baseline queries.

The results show a major increase in the number of relevant documents retrieved and hence depicts a higher mean average precision upon implementing the proposed algorithm. Table III displays mean average precision of 0.44 for base line queries that increased to 0.85 after implementation of the second stage of WeOnto algorithm showing remarkable improvement of 93% as compared to an improvement of 18% at first level of the algorithm as per Eq. 3 in the efficiency of information retrieval system. Even precision at 10 also depicts a clear increase and states that top 10 documents retrieved are 75% more relevant as compared to initial baseline queries.

The graph in Fig. 8 depicts a significant upgrade in the values of the metrics required for performance evaluation of REIR model calculated at each level as described in the paper. It clearly shows an increase in efficiency of information retrieval using the semantically enriched ontology and word embeddings model of NLP for quick retrieval of real estate related legal documents.

A trend showing usage of semantic ontology [21] for query expansion was already there. Its aggregation with word embeddings has proved to give better information retrieval results.

It is evident that WeOnto algorithm proposed in the paper includes the usage of the combination of web ontology and word embeddings as also mentioned in [22] for the purpose of query expansion has given significant results with respect to information retrieval of web documents as compared to the baseline user queries.

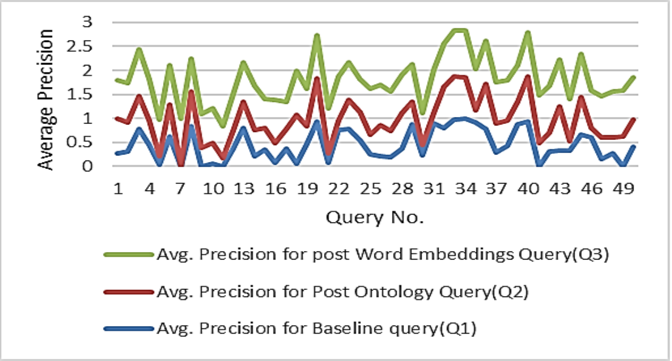


Fig. 6. Graph showing Average Precision.

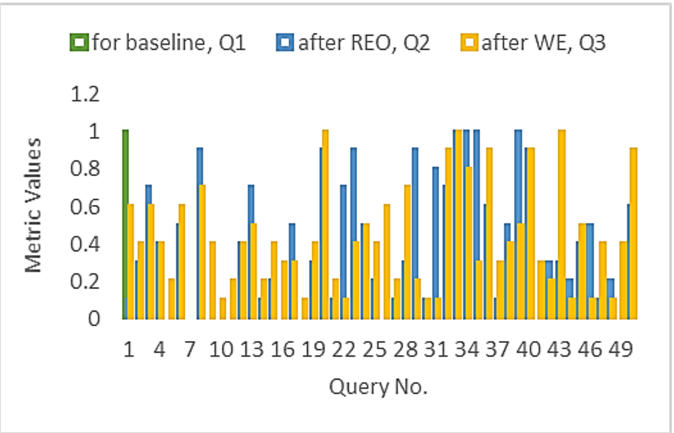


Fig. 7. P@10 of 50 Queries.

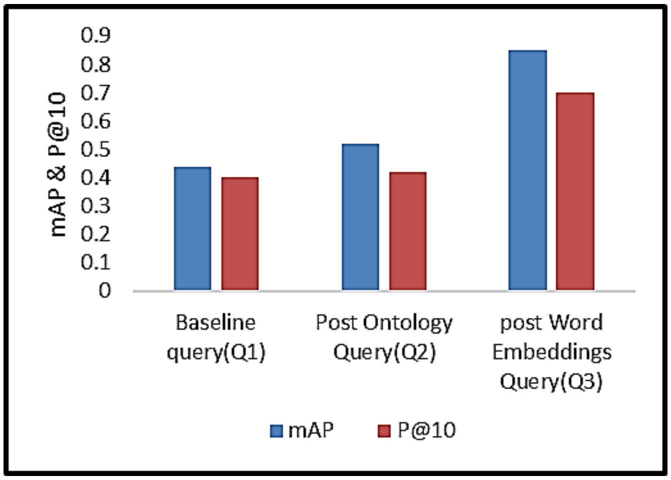


Fig. 8. Graph showing Calculated MAP & P@10.

TABLE II. SAMPLE OF P@10 METRIC FOR ALL 50 QUERIES

P@10	**Query No.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	15	16	17	18	19	20
	for baseline, Q1	0.1	0.3	0.7	0.4	0	0.5	0	0.9	0	0	0	0.4	0.7	0.1	0.2	0	0.5	0	0.3	0.9
	after REO, Q2	0.6	0.4	0.6	0.4	0.2	0.6	0	0.7	0.4	0.1	0.2	0.4	0.5	0.2	0.4	0.3	0.3	0.1	0.4	1
	after WE, Q3	0.7	0.6	1	0.5	0.7	0.5	0.1	0.8	0.6	0.6	0.6	0.7	0.8	0.8	0.4	0.6	0.5	0.3	0.8	0.8

TABLE III. IMPROVEMENT IN MAP & P@10

Metric Used	Baseline query(Q1)	Post Ontology Query(Q2)	post Word Embeddings Query(Q3)
mAP	0.44	0.52	0.85
P@10	0.4	0.42	0.7

V. CONCLUSION

Improving the process of information retrieval for efficient retrieval of web documents with high precision has been an ever-going process. Numerous methods have been developed from time to time be it traditional Boolean models or vector state models or even probabilistic models. Each of them was more concerned with queries having keyword matching and had very little understanding of the semantics or context of the query formed.

Query expansion that includes the reformulation of the user query showing better IR results has been a promising solution. The proposed algorithm, WeOnto works on same query expansion and suggests using a two-step procedure that uses ontologies and word embeddings. The ontology gives semantically enriched keywords for the user-query tokens whereas Word2Vec model learns from the given corpus and give most similar words for the said tokens. The best keyword from the entire set is extracted to form the final reformulated query that gave remarkable results and increased precision of the web documents retrieved. In future, instead of word embeddings, sentence-based embeddings can be devised. Also, as the embeddings are shallow unsupervised NLP techniques, the learning of the model can be improved by growing the size of the corpus.

REFERENCES

[1] Q. Liu, H. Huang, J. Lut, Y. Gao and G. Zhang, "Enhanced word embedding similarity measures using fuzzy rules for query expansion", 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2017, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2017.8015482.

[2] M. Zhang, Y. Liu, H. Luan, M. Sun, T. Izuha and J. Hao, "Building Earth Mover's Distance on Bilingual Word Embeddings for Machine Translation",AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Arizona, February 2016, pp. 2870–2876.

[3] Qiu D, Jiang H and Chen S., "Fuzzy Information Retrieval Based on Continuous Bag-of-Words Model", Symmetry. 2020; 12(2):225. https://doi.org/10.3390/sym12020225.

[4] Phan H.T., Nguyen N.T., Musaev J., Hwang D. , "A Method for Improving Word Representation Using Synonym Information", In: Paszynski M., Kranzlmüller D., Krzhizhanovskaya V.V., Dongarra J.J., Sloot P.M. (eds) Computational Science – ICCS 2021. ICCS 2021, Lecture Notes in Computer Science, 2021, vol 12744. Springer, Cham. https://doi.org/10.1007/978-3-030-77967-2_28.

[5] Siriguleng, "Mongolian Information Retrieval Method Based on Word2vec and Topic Model," 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019, pp. 1217-1220, doi: 10.1109/IAEAC47372.2019.8997588.

[6] Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C., "Evaluating word embedding models: Methods and experimental results.", APSIPA Transactions on Signal and Information Processing, 2019, 8, E19. doi:10.1017/ATSIP.2019.12.

[7] B. Mansurov and A. Mansurov," Development of Word Embeddings for Uzbek Language", Computation and Language (cs.CL), Sep2020, https://arxiv.org/abs/2009.14384v1.

[8] Y. H. Farhan, S. A. M. Noah, M. Mohd and J. Atwan, "Word Embeddings-Based Pseudo Relevance Feedback Using Deep Averaging Networks for Arabic Document Retrieval", Journal of Information Science Theory and Practice, vol 9, no. 2, pp. 1-17, June 2021, DOI: 10.1633/JISTaP.2021.9.2.1.

[9] El Mahdaouy, A., El Alaoui, S.O. & Gaussier, E., "Improving Arabic information retrieval using word embedding similarities", International Journal of Speech Technology, 21, 121–136 (2018). https://doi.org/10.1007/s10772-018-9492-y.

[10] E. H. Mohamed and E. M. Shokry, "QSST: A Quranic Semantic Search Tool based on word embedding", Journal of King Saud University - Computer and Information Sciences, Jan 2020, https://doi.org/10.1016/j.jksuci.2020.01.004.

[11] J. Ren, H. Wang, and T. Liu, "Information Retrieval Based on Knowledge-Enhanced Word Embedding Through Dialog: A Case Study", International Journal of Computational Intelligence Systems, Volume 13(1), pp 275-290, 2020 https://doi.org/10.2991/ijcis.d.200310.002.

[12] V. Jayawardana et al., "Semi-supervised instance population of an ontology using word vector embedding," 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2017, pp. 1-7, doi: 10.1109/ICTER.2017.8257822.

[13] J. J. L-Díaz, J. Goikoetxea, M. A. H. Taieb, A. G.-Serrano, M. B. Aouicha, and E. Agirre, "A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art", Engineering Applications of Artificial Intelligence, volume 85, pp 645-665, 2019, https://doi.org/10.1016/j.engappai.2019.07.010.

[14] Aubaid, M. Asmaa, and A. Mishra, "A Rule-Based Approach to Embedding Techniques for Text Document Classification" Applied Sciences, Vol. 10, no. 11: 4009, 2020, https://doi.org/10.3390/app10114009.

[15] D. Jatnika, M. A. Bijaksana, A. A. Suryani, "Word2Vec Model Analysis for Semantic Similarities in English Words", 4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), Procedia Computer Science, Volume 157, Pp.160-167, 12-13 September 2019.

[16] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient Estimation of Word Representations in Vector Space", Computation and Language (cs.CL), 2013, arXiv:1301.3781 [cs.CL].

[17] H. K. Azad and A. Deepak, "Query expansion techniques for information retrieval: A survey", Information Processing & Management, Volume 56, Issue 5, Pages 1698-1735, 2019, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2019.05.009.

[18] N. Rastogi, P. Verma and P. Kumar, "Evaluation of Information Retrieval Performance Metrics using Real Estate Ontology," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 102-106, 2020, doi: 10.1109/ICSSIT48917.2020.9214285.

[19] G. Besbes and H. Baazaoui-Zghal, "Fuzzy ontology-based Medical Information Retrieval," 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 178-185, 2016, doi: 10.1109/FUZZ-IEEE.2016.7737685.

[20] N. Rastogi, P. Verma, P. Kumar, "Ontological Design of Information Retrieval Model for Real Estate Documents"., In: Chaudhary A., Choudhary C., Gupta M., Lal C., Badal T. (eds) Microservices in Big Data Analytics. Springer, Singapore., 2020.

[21] N. Yusuf, M. A. M. Yunus, N. Wahid, A. Mustapha, and M. N. M. Salleh, "A Survey of Query Expansion Methods to improve Relevant Search Engine Results", International Journal on Advanced Science, Engineering and Information Technology, Vol 11, No 4, pp 1353-1359, 2021, http://dx.doi.org/10.18517/ijaseit.11.4.8868.

[22] S. D. Kok and F. Frasincar, "Using Word Embeddings for Ontology-Driven Aspect-Based Sentiment Analysis", SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, pp 834-842, Sep 2020, https://doi.org/10.1145/3341105.3373848.