Review on Information Retrieval for Desktop Search Engine

Jahnavi.P[1], Jammula Nikhil[2], Arindam Chowdhury[3], Karthik.S.A[4]

[1,2,3]Student, Dept. of Information Science and Engineering, Dayananda Sagar of Technology and Management, Karnataka, India
[4]Guide, Asst.Professor, Dept. of Information Science and Engineering, Dayananda Sagar of Technology and Management, Karnataka, India

---***---

Abstract - *Search is an important aspect of information management often taken for granted. Domain specific repositories are growing in both size and numbers calling for efficient search and retrieval of documents. This paper explores the possible techniques and necessary system components for a search engine charting several iterative optimizations over the last few years. This paper focuses on NLP models while retaining basic principles from other methods that assist in information search.*

Key Words: Information Retrieval, Query Understanding, Feature Extraction, Entity Recognition, Similarity Measures

1. INTRODUCTION

Data generation has increased exponentially with time as computing has grown ubiquitous. The internet is a great facilitator when it comes to knowledge discovery and dissemination. This property of the internet stems from extremely efficient data cataloguing, storage, and retrieval from large scale real time distributed system. Initially, rudimentary information retrieval systems constituted a simple query to access a particular document from a local database. Over time these databases have grown to span across networks and storage mediums and these gave rise to search algorithms designed for user convenience. Accessing documents from a database through a query mentioning document id or title proved to be difficult when one wanted to retrieve large number of documents. It was also an issue to actually give correct values for the document details such as title, author, keywords, domain and more.

Later this lead to rise of Information Retrieval based on ontology where documents were classified with respect to some entities they contain. It used inverted indexing of the classes they belong to and the keywords they contain [1].

However, information retrieval improved to the levels where user need not necessarily know the keywords to get desired documents. It was possible due to efforts to understand the intent behind the user's query and the concept the document entails.

2. METHODOLOGY

2.1 Feature Extraction – Entity Recognition

Extracting entities is a preprocessing technique either based on linguistic rules or n-grams word or phrase extraction [2][3]. Algorithms based on linguistic rules to tag parts of speech are popular for feature extraction. Their rules are manually written and updated with regards to standards of English Grammar.

Entities and their associative categories, either manually tagged [2][4] or classified by an algorithm, become important features that influence retrieval models. The possibility of including the synonyms, antonyms, acronyms and more with help of word embeddings enrich the extracted features to cover the meaning behind the context and help the model infer meaning behind the query.

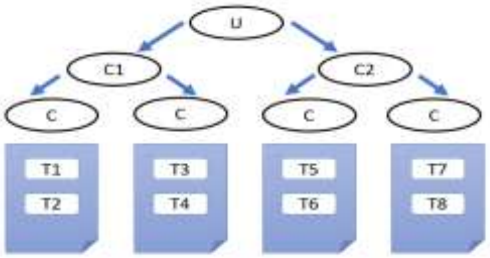


Fig -1: Universal concept graph of Agrawal et.al [4]

Many approaches to word-category disambiguation have been developed that give the most optimal results in context understanding. Ontology based entity classification [1][3] is the most primitive yet classical attempt to entity disambiguation where the documents or their paragraphs can be mapped to a particular category to enrich the meaning behind the group of entities. It is still the preferred technique for information retrieval in closed domain as there are clearly defined boundaries and categories to map the context to their categories. Knowledge Bases provide more varieties of categories to enrich extracted features in a non-conventional manner. Knowledge Base such as Wikipedia [2] is a store for structured data where the entities and their relationships can be mined. The facts are represented as attribute: fact pairs that gives flexibility in storing various types of relationships. This opens the door the inference in the highest levels of abstraction with regards to navigational queries. On the opposite end of this stream is ProBase which is a Knowledge base of isA relationships between entities that can be mined to build a concept graph in levels of abstractions [4] or a semantic network [9]. Mapping extracted entities of the text to ProBase results in a probabilistic model that classifies the text according to the probability based on Bayes theorem. Bayes theorem describes the probability of a term belonging to the particular concept given the prior information about its isA relationship occurrences.

Table -1: Example of Instance terms in queries which are mapped to ProBase [9]

Query	Non-Instance	Type	Instance
watch harry potter	watch	verb	harry potter
most dangerous python in the world	dangerous	adjective	python
population of china	population	attribute	china

Word Embedding is an alternative approach to entity disambiguation which uses a vector space model to retrieve semantically similar terms to the entities [6][8][10]. It is quite simple and provides an opportunity to involve synonyms and acronyms of the extracted entities of the text. VerbNet [10]and WordNet [6][8] are framework that provide reliable similar semantic representations to the entities and the possible categories they belong to.

2.2 Query Understanding

Query Understanding seem to have many approaches and perspectives which rather makes one be dispirited to get to the bottom of it.

Understanding the intent behind a user's query regardless of the technique is query understanding. Larger part of Query Understanding seems to be POS tagging and segmentation [9]. This works for all ontology based, graph based and vector space-based models of Information retrieval [1]. If one wants to avoid the issue of not retrieving any document due to absence of a key word of query in the document it is better to introduce some word embeddings as part of query expansion[2] or sub-query generation[4] that later hit these queries against the search engine and gets the results.

It is also imperative to carefully choose a representation model for the knowledge in the documents. One can choose Index [1], Concept Graphs [4], Semantic Networks [9] or Vector Space Models.

Indexing is the method creating a list of keywords for every document. It is a basic File Structures method of Information Retrieval.

Inverted Indexing is the method of Indexing that evaluates whether a keyword is present in the document or not by creating a list of documents containing that particular keyword. This is the primitive method to provide search functionality [1][7].

Later properties of graphs seem to be a suitable knowledge representation for the documents. The nodes contain a term or concept while the edges represent the relationship between the nodes in the graph. If the nodes and the edges have semantic relationship then the graph becomes a semantic network [9][11]. If the terms in the nodes have a lexical relationship then the graph becomes a concept graph [4].

However, it is still not possible to complete the process of understanding user intent if one doesn't consider the synonyms or alternatives for a concept. Kuzi et.al [2] specifically says to enrich the entity values of a query with alternative synonyms to help in retrieval.

2.3 Similarity Measure and Scoring/Ranking1

One of the simplest formulae for scoring is mentioned in [1]. Since it is based on ontology, list of keywords is associated to the class and are compared to the terms in the user's query. It uses Bag Of Words approach to classify the query according to user's Intent.

$$Score = \frac{\text{no of matches between query terms and keywords}}{\text{no. of query terms}}$$

Agarwal et.al [4] presents that search engine performs better in low dimensions hence it is necessary to deal with only few concepts of user's query at a time. User's query traverses the concept graph built and then sub queries are hit against the search engine. The results retrieved are aggregated and relevance scores are computed. The highly relevant docs are retained and retrieved a result to the user's query.

Gelbukh et.al [10] to compute max-common sub graph, to compare the similarity between graphs i.e. the document and the query. Vector mapping between the two graphs ensure to help score the max-common sub graph. Scoring of max-common sub graph is similar to vector scoring.

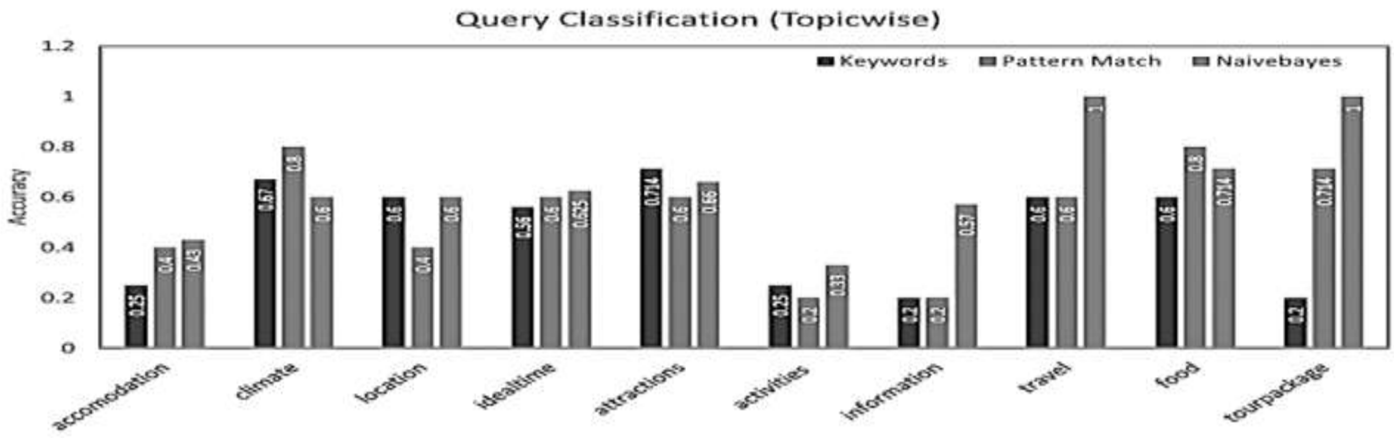
$$P_{qd} = \frac{\mathrm{P}(R_{qd}^{>}) + \mathrm{P}(R_{qd}^{\geq})}{2}$$

Gelbukh et.al [10] also deals with ambiguous equal ranking by using the above equation where P is the precision, $R^{>}_{qd}$ is the set of documents scored higher than d and $R^{>}_{qd}$ is the set of documents scored higher or equally as d.

2.4 Information Retrieval

In Agrawal's and et.al paper [4] the query and document representations need not be the same. The documents are processed to make a universal concept graph and the combination of the core concepts and directly fed to the search engine.

Chart -1: Query Classification Topic score provided by Prasath et.al [1]



Buttler et.al [5] presents a pseudo – feedback relevance feedback that helps to enrich the search functionality. A set o learned latent topics that are relevant to the query are provided as suggestion and if user chooses them, the query is augmented. Even Kuzi et.al [2] and Ganguly et.al [6] shows its results by using pseudo – feedback relevance model.

Techniques of Inference accommodate uncertainty. It is possible to delve more deeper into the relationships with help of proposition logic while traversing the sematic network and retrieve the documents pertaining to the particular node [11].

Seok et.al [3] uses a CRF model for NER task as CRF's represent a probabilistic framework for labelling and segmenting sequential data.

$$P(y|x) = \frac{1}{Z(x)} \prod_{i=1}^{n} exp \sum_{j} \lambda_j f_j (y_{i-1}, y_i, x, i)$$

Input: is a sequence of words x_1 to $x_{n;}$ tags y_1 to y_n are allocated. Equation $f_j(y_{i-1}, y_i, x, i)$ is a feature function that takes input: sentence x, position i, label y of current word and label of previous word.

$$Z(x) = \sum_{y} \exp\left(\sum_{i=1}^{n} \sum_{j} \lambda_j \, f_j \, (y_{i-1}, y_i, x, i)\right)$$

is a normalization vector that 'sums' the scores.

3. RESULT COMPARISION

Prasath et.al [1] shows there is a drop in precision score when 10 topics are considered and also proves in its experiment that entity recognition and resolution perform better than pattern matching in feature extraction and is shown in Chart1.

Agrawal et.al [4] provides a CGSimilarity algorithm to construct the concept subgraphs against the use of random walks over the concept graph to end with a semantically similar concept. While both techniques formed queries of same concept phrases, CGSimilarity is much preferred.

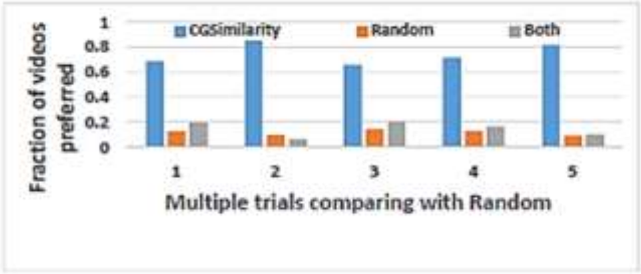


Chart -2: Comparative performance between CGSimilarity and Random.

Comparisons between the vector space models with and without embeddings shown in table2 provided by Seok et.al [3] proves that embedding is a feature that has impact.

Table -2: Comparison between types of embeddings in Seok's and et.al [3]

	Test A	Test B
Baseline	84.12%	77.06%
Baseline + GloVe	85.18%	79.48%
Baseline +Word2Vec	85.89%	**80.72%**
Baseline+CCa	**85.96%**	80.68%

Table -3: Comparison of techniques and their impact on retrieval

Paper	Chapter	Technique	Baseline Measure/Score	Technique Measure/Score	Remarks
[1]	Information Retrieval	Ontology based Topic classification of the recognized entities	Keywords: 0.46 Pattern Matching: 0.53 Na�e Bayes: 0.65 Baseline- VSM: 0.47	Topic indexed based retrieval: 0.55	14.145% improvement
[4]	Similarity Measure	Concept Graph Based query generation	Random: 0.1	CGSimilarity: 0.8	Using CGSimilarity is much efficient in doc retrieval
[9]	Feature Extraction – Concept Clusters	Semantic Network	IJCAI11 – Bayesian Analysis: 0.84 LDA – Co - occurrence and ProBase: 0.83	Random Walk: 0.87	Seems that Random walk covers much more of User's Intent Under this Technique
[2]	Query Understanding	GOV2 Model of Word Embeddings	RI score of RM3: 0.392	RI score after integration in RM3: 0.432	Including Synonyms is exhaustive
[6]	Feature Extraction- Word Embedding	Generalized Language Model	Recall score of LDA: 0.58	Recall score of GLM: 0.62	Vector word embedding is exhaustive

4. CONCLUSIONS

This paper concludes that knowledge representation of a document and query has a high impact on user's query intent understanding. Building index is efficient when one deals with a collection of documents where they are classified based on ontology i.e. the classes are pre-defined. If one wants to deal with information of open domain, representing data in terms of entities and concepts is recommended. Query Understanding can be tackled if one considers lexical and semantic relationships between the terms [9]. However, it is rather exhaustive to maintain a universal concept graph or semantic network for it does not give space for document to be edited or deleted. This is an issue that remains unexplored due to assuming that all documents are static. Vector space models are not recommended for they do not give enough consideration towards lexical and semantic relationships between the vectors in a text which puts the model at a disadvantage in understanding intent behind the query.

REFERENCES

[1] Prasath R, Kumar V & Sarkar S. (2015), "Assisting web document retrieval with topic identification in tourism domain", Web Intelligence, 13(1), 31–41, IOS Press, doi: 10.3233/web-150308.

[2] Kuzi S, Shtok A, & Kurland O (2016)," Query Expansion Using Word Embeddings", Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16, doi:10.1145/2983323.2983876.

[3] Seok, Miran & Song, Hye-Jeong & Park, Chan & Kim, Jong-Dae & Kim, Yu-Seop. (2016), "Named Entity Recognition using Word Embedding as a Feature", International Journal of Software Engineering and Its Applications. 10. Pg:93-104. doi 10.14257/ijseia.2016.10.2.08.

[4] Agrawal R, Gollapudi S, Kannan A, & Kenthapadi K. (2014), "Similarity Search using Concept Graphs", Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14, doi:10.1145/2661829.2661995.

[5] Andrzejewski D, & Buttler D (2011), "Latent topic feedback for information retrieval", Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '11, doi:10.1145/2020408.2020503.

[6] Ganguly D, Roy D, Mitra M, & Jones G. J. F. (2015)," Word Embedding based Generalized Language Model for Information Retrieval", Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15, doi:10.1145/2766462.2767780.

[7] Potthast M, Hagen M, Stein B, Grae�gger J, Michel M, Tippmann M, & Welsch C (2012), "ChatNoir: A Search Engine For the ClueWeb09 Corpus", Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '12, doi:10.1145/2348283.2348429

[8] Zuccon G, Koopman, B, Bruza P, & Azzopardi L (2015), "Integrating and Evaluating Neural Word Embeddings in Information Retrieval", Proceedings of the 20th Australasian Document Computing Symposium on ZZZ - ADCS '15. doi:10.1145/2838931.2838936.

[9] Zhongyuan Wang, Kejun Zhao Renmin, Xiaofeng Meng, Ji-Rong Wen, "Query Understanding through Knowledge-Based Conceptualization", Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015).

[10] Ordonez-Salinas S. Gelbukh A, "Information Retrieval with a Simplified Conceptual Graph-Like Representation", In: Sidorov G, Hernandez Aguirre A, Reyes Garcia C.A. (eds) Advances in Artificial Intelligence. MICAI 2010, Lecture Notes in Computer Science, vol 6437 Springer, Berlin, Heidelberg, doi: 10.1007/978-3-642-16761-4_9.

[11] Koopman B, Zuccon G, Bruza P, Sitbon L, & Lawley M (2015), "Information retrieval as semantic inference: A Graph Inference model applied to medical search. Information Retrieval Journal", 19(1-2), 6–37, doi:10.1007/s10791-015-9268-9.