Pseudo Documents for Resource Scarce Languages

Maulik Vachhani, Devang Kulshreshtha*, Arjun Atreya, Pushpak Bhattacharyya and Ganesh Ramakrishnan
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay/BHU*
{maulik, arjun, pb, ganesh}@cse.iitb.ac.in
devang.kulshreshtha.cse14@iitbhu.ac.in*

***Abstract*—A web search for queries in resource scarce languages often produces poor or no results. This is more frequent when the intent of the query is transactional. Traditional methods retrieve pages having information or links about the words in the query, and not satisfying the ”need” of the intent. To improve search results in such cases, we provide an approach for generating Pseudo-Documents for resource scarce languages in case of transactional and other types of queries. We construct a rule-based Pseudo-Document generation framework to find pages that are guaranteed to be useful to the user. We combine our results with a general search engine. By testing the combined system on a set of queries in a resource scarce language(Hindi), we establish its robustness. Our experimental results show that the combined system outperforms the original system in terms of precision in search results.**

***Keywords*—Information Retrieval, Resource scarce languages, Pseudo Documents**

I. Introduction

A web search query, based on the user intention can be classified into three categories: *informational query*, *transactional query* and *navigational query* [3]. Most of the classical search engines are built on the assumption that the need behind the query is *informational.* However, according to [1, 7] in the majority of the queries the intent of the user is transactional. The classical search engines do not work well for these types of queries[2, 5]. However, when the query is in a resource rich language, various techniques such as query expansion and knowledge graphs are useful to find results for these type of queries. [4] has proposed a method for identifying transactional queries and [6] has proposed a method to handle transaction queries. But these methods are useful for resource rich languages only.

The problem is more specific to resource scarce languages(e.g. Hindi, Marathi, Bengali, Tamil etc.). The number of available documents present in the index for such languages are very less. And mostly statistical techniques are used for finding and ranking documents, even in this case. But they are based on the concept of tf-idf. Due to which if a query word is not present in a document, it will not be shown as a search result. For e.g., if a user wants to book a ticket, he/she searches the query like ”दिल्ली से पुणे(dilli se pune;delhi to pune)”. It would be convenient if we provide a link to the page where the travel information from ”दिल्ली(dilli;delhi)” to ”पुणे(pune;pune)” can directly be found. However classical search results will contain those documents which contain information about the words ”दिल्ली(dilli;delhi)”, ”पुणे(pune;pune)” and not the actual travel information. For resource scarce languages, the results would not be addressing the actual ”need” of the query. Techniques such as knowledge graphs and Query expansion can help[8] greatly for resource rich languages. However, many users(like native Indian language speakers where languages are resource scarce) want to get results for such queries in their corresponding language. Query expansion is very difficult and not much accurate for resource scarce languages[cite]. Users then have to translate the query to a resource rich language(e.g. English) and then search for information. Query expansion also performs poorly if semantically related words are also not found in the relevant document(as in the previous example). Even, if after query expansion we find the relevant document, it may not be present at the top of the search results. The user has to scan through the complete list of results.

In this paper, we try to identify and present documents which we are sure of are highly beneficial to the user but not present in the index. We call these documents as Pseudo-Documents, since they are not present in the index yet are highly relevant to the user. We are certain about relevance of the Pseudo-Documents to show for a particular query, and we can show them at the top, along with the normal search results. The methods are specifically targeted for queries in resource scarce languages. Showing additional results on the top of existing search results is high-risk, high-gain. Hence, we have to present only those documents which are sure to be relevant to the user. Hence precision of such a system is important. We have developed our approach to minimize the risk involved and simultaneously achieve gain in the precision of the search. We compare the precision of our proposed approach with Sandhan[1], a multilingual search engine for 9 different Indian languages.

II. Architecture of the system

In this section, we describe the working of our system. The block diagram is shown in figure 1. As described in the figure 1, first we translate or transliterate the query to Resource-rich(English) language. For translation/transliteration to English, we have used the Microsoft Bing Translate API. Parallely, we also pass this query to General search engine(Sandhan) and retrieved the resultant documents. Another hand after translating the query we parse it and classify into one of the following classes:

[1] The Sandhan search engine is hosted at http://www.sandhansearch.in

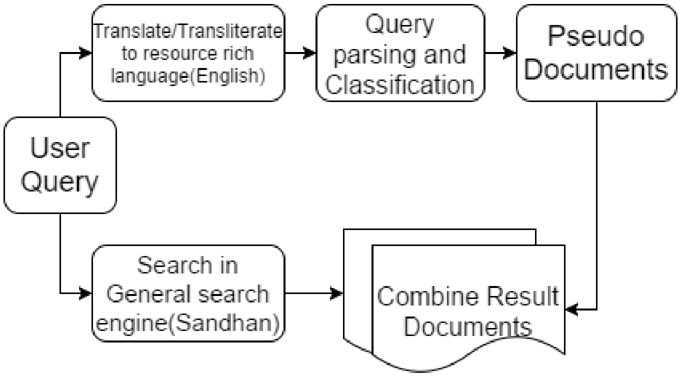
Translate/Transliterate to resource rich language(English)
Query parsing and Classification
Pseudo Documents
User Query
Search in General search engine(Sandhan)
Combine Result Documents

Fig. 1. Block Diagram of the System

- Transactional Query: A query in which a user intents to initiate the transaction is called transactional query.
- Computational Query: A query in which a user intents to perform formula based calculation is called computational query.
- Informational Query: A query that covers broad topic or intention of a user is to get information about some specific topic is called informational query.
- Multimedia Query: A query in which user expects the result which contains links to videos, audios or images is called multimedia query.
- Lookup Query: A query in which user wants to find meaning of a word or a phrase is called lookup query.
- General query: A query which is not classified as one of the other class is considered as a general query.

After classifying the query we handle the query differently based on its class. Except for general query class, we return link(s) to the document. We are certain that those documents are relevant to the user query. We call those document(s)/link(s) as Pseudo Documents because those link(s) to the documents are not present in the index table.

A. Handling Transactional Queries

A transactional query is one in which a user intends to perform transaction like buying, selling or booking some product. Websites which initiate the transactions are in English only. But the major content on these websites is NEs only. So, user with limited knowledge of English also understands these contents. It is very difficult for a user to go to the online store or portal and search for particular product or service in English. For example, when Hindi native speaker wants to buy ”गेहूँ (gehoon;wheat)”, it is very difficult for the user to go to online store and search for wheat. These websites are called medium of information access which is in the English language. But the page where metadata(image, price, description etc.) of wheat is available, it is the user required content. If we provide the direct link to this page in addition to general web search result then it would be convenient for the user to buy wheat. This will improve the precision of the result. Another example is, suppose a user wants to buy a ticket from Delhi to Mumbai. It is difficult for the user who doesn’t know English to go to online portal and fill the form in English. Instead user can search the query like ”दिल्ली से मुंबई(Dilli se Mumbai; Delhi to Mumbai)” in our search engine. In addition to general result we will provide a link to the page where the details of the form has been filled using the information provided in the query. Now it is convenient for the user to pick appropriate flight/bus/train to book the ticket. So there are many such applications.

In this paper, we explain how to handle transactional query using travel ticket booking example. The same can be applied for the rest of queries like buying, selling etc. First we translate non-English query to English. Then, in a query when we find two named entities which are the names of locations and/or time then we consider query as potential transactional query. Then we validate the query that symbol or word which connects the two locations is intent to ask about booking of the travel ticket. For example, if connecting word is like ”in” or ”,” then we won’t consider query as transactional query. After validation we do location resolution. In this each word of the query is checked to be a potential city/state/station etc. Then we identify source and destination and map them to transportation code. Then we extract the date or day from the query if it is available. We convert the time into absolute time if it is present in relative time like tomorrow, today, next week, next month etc. If user has provided the information like bus/train/flight in the query then we return specific link only.Once we get the values of source, destination, and date; URLs for buses, trains, flights can be generated on the fly, which upon clicking will redirect to the bookings website. Different templates are used to generate URLs for different websites.

B. Handling Computational Queries

A computational query is in which user intent to perform a mathematical/scientific calculation or unit transformation. It is difficult for a user who doesn’t know English to search on websites for specific unit transformation which provides many different unit transformations. So, it is convenient for such a user if our search result leads him/her to a page which provides user required unit transformation or calculation. To check whether a query belongs to a computational query class or not; first, we translate the query in English. In a translated query if we find numeric, mathematical symbol and unit only then we consider query as potential Computational Query. Then we validate the query and generate the URL upon clicking lead a user to a page where required task can be performed. Wolfram alpha is such a website which provide mathematical/scientific calculation or unit transformation. So in our system, we generate the URL which leads a user to a page of Wolfram alpha where user required task can be performed easily.

TABLE I. Relevance of the query.

Query	Type of query	p@5 (Sandhan)	p@10 (Sandhan)	p@5 (Sandhan + Pseudo Documents)	p@10 (Sandhan + Pseudo Documents)	Pseudo Documents are brought or not(1/0)
सोमनाथ मंदिर (Somnath temple)	Informational	0.2	0.4	0.4	0.5	1
वाराणसी घाट (Varanasi Ghats)	Informational	0	0	0	0	0
स्वर्ण मंदिर, अमृतसर (Golden Temple, Amritsar)	Informational	0.2	0.1	0.4	0.3	1
उत्तराखंड में तीर्थ स्थलों (Pilgrimage sites in Uttarakhand)	Informational	0.8	0.8	0.8	0.8	0
तिरुपति (Tirupati)	Informational	0.4	0.4	0.6	0.5	1
पुणे मुंबई बस समय (Mumbai Pune bus timing)	Transactional	0	0	0.2	0.1	1
सुंदरबन आवास (Sunderbans accommodation)	Transactional	0	0	0	0	0
बंगलौर से सप्ताहांत यात्राएं (Weekend trips from Bangalore)	Informational	0	0	0	0	0
भारत में रिजर्व वनों (Reserve forests in India)	Informational	0	0	0	0	0
जबलपुर मार्बल रॉक्स (Jabalpur Marble Rocks)	Informational	0	0	0	0	0
पुष्कर मेला (Pushkar Fair)	Informational	0.4	0.3	0.4	0.4	1
वैष्णो देवी परिवहन (Vaishno Devi transportation)	Transactional	0.2	0.1	0.2	0.1	0
हिल स्टेशनों दक्षिण भारत (Hill stations South India)	Informational	0	0.1	0	0.1	0
नालंदा विश्वविद्यालय (Nalanda University)	Informational	0.6	0.3	0.8	0.4	1
दिलवाड़ा मंदिर माउंट आबू (Dilwara Temple Mount Abu)	Informational	0	0	0	0	0
पहियों पर पैलेस (Palace on wheels)	Informational	0.2	0.1	0.2	0.1	0
गोवा मुख्य समुद्र तटों (Goa main beaches)	Informational	0	0	0	0	0
रथयात्रा पुरी (Rathyatra Puri)	Informational	0	0.1	0	0.1	0
कामाख्या मंदिर (Kamakhya temple)	Informational	0	0	0.2	0.1	1
काजीरंगा राष्ट्रीय उद्यान (Kaziranga national park)	Informational	0.2	0.2	0.4	0.3	1
मुंबई से पुणे कल (mumbai to pune tomorrow)	Transactional	0	0	0.6	0.3	1
नरेंद्र मोदी का भाषण वीडियो (Videos of narendra modi speech)	Multimedia	0	0	0.6	0.3	1
जंगली हाथियों का वीडियो (Videos of wild elephants)	Multimedia	0	0	0.6	0.3	1
क्रिकेट में बेहतरीन छक्के के वीडियो (Videos of best sixes in cricket)	Multimedia	0	0	0.6	0.3	1
५ + ७ (5 + 7)	Computational	0	0	0.2	0.1	1
८ * ६ + ७ (8 * 6 + 7)	Computational	0	0	0.2	0.1	1
तोप का अर्थ (Meaning of cannon)	Lookup	0	0	0.2	0.1	1
अपराधी का अर्थ (Meaning of guilty)	Lookup	0	0	0.2	0.1	1
मजाक का अर्थ (Meaning of fun)	Lookup	0	0	0.2	0.1	1

C. Handling Informational Queries

An Informational Query is in which user intent to get information about particular place, person or movie etc. To address this type of English(Resource-rich Language) query, general search engines use knowledge graph to retrieve the results. Another hand, for resource-scarce languages, generating a knowledge graph is very difficult. But we know that websites like Wikipedia, DBpedia, and personal blogs have very rich information content. In addition to general result links if we provide a link from one of those web pages, it will be very useful for end user.

To explain how to handle informational query we use an example of Wikipedia page. First, we translate the non-English query to English and then we identify the NERs. If the query contains only NERs then we consider query as a potential informational query. Then we search query in our Wikipedia dump. If we find a perfect match between any wiki page's title and query then we add the link to that wiki page to the result. For example, a query is ”सोमनाथ मन्दिर(Somnath mandir; Somnath temple)”. Then we return the link of Wikipedia page: ”https://hi.wikipedia.org/wiki/सोमनाथ_मन्दिर”

D. Handling Multimedia Queries

A query in which user wants results in the form of videos, audios or images is called multimedia query. The audios or videos which have content in a non-English language, also have metadata in English only. So when a user searches a multimedia query in a non-English language, it is not possible to search for those videos/audios due to the difference in the languages. But if we provide a link of such video/audio it is relevant for the user. For example, a query is ”जंगली हाथियों के विडियो(junglee hanthiyo ke video; videos of wild elephants)”. Link we provide is relevant for a user if it leads to the video which has metadata in English.

To handle multi-media query, first, we translate a non-English query to English. Then we look for a word like ”audio”, ”Video” etc. in the translated query. If we find one of those words in the translated query we consider it as a multimedia query. To get the link to video our system uses YouTube API. In return, we get 25 most relevant video based on translated query. Then at our system's end, each of the videos is assigned a score based on the title's resemblance with the query. Based on the scores, the top 3 results are presented as result link in addition to general result links.

E. Handling Lookup Queries

A query in which user wants the meaning of a particular word or a phrase from the dictionary. In resource-scarce languages like Hindi, it is rare to return link to the meaning of a word or phrase using general web search technique. Hence, we apply a rule-based model to handle this type of query. First, we look for ”meaning” word in the query. If we find ”meaning” word in the query then we consider it as Lookup query. We search for this word in query language dictionary and return link to the page which contains query word. For example, a query is ”खाने का अर्थ(khane ka arth; meaning of eating)”. Then we return the link to the page which contains the meaning of ”खाना (khana; to eat)”.

III. Experiments and Results

We set up our system on the top of the Sandhan search engine. The whole system is described in figure 1. In our experimental setup, we use Sandhan as general web search engine. So when a user enters a query, we send it to Sandhan and the module which handles classified query. Then we take at most three results from the module which handles classified query and rest of the result from the Sandhan. Then we consolidate those two result set together. We tested our system on 29 standard queries. The precision values of these queries are presented in the table I. To calculate the precision of the query we assign 0/1 score to each result link. The average precision of 29 queries can be found at table II. We are creating a large gold data-set to test this system quantitatively.

TABLE II. Precision Table

Precision	Sandhan	Sandhan with our Module
P@5	0.11	0.275
p@10	0.1	0.19

IV. Conclusion

In this paper, we introduced a methodology for presenting pseudo-documents for resource scarce languages for different types of queries. We investigated in finding pseudo-documents for transactional, computational, multimedia, lookup and informational query. Our method is beneficial to the community who wants to search in their native language, but the language itself is not very resource rich. We showed the robustness of our method by analyzing the search results for various types of queries and comparing the precision of our system with the Sandhan Search Engine. The queries are in Hindi, but the model is applicable for any resource scarce language. Our method is mostly rule-based/restricted rule-based and it based on direct URL finding in most cases. In our methodology, we used to generate URLs based on specific rules. In the future, we would like to crawl lots of URLs using a small number of seed URLs and then we would like to classify URLs and return a result based on query class and URL class. We would like to classify queries and URLs into fine classes so that we can provide more accurate result links, specifically in transactional queries.

References

[1] Andrei Broder. “A taxonomy of web search”. In: *ACM Sigir forum.* Vol. 36. 2. ACM. 2002, pp. 3–10.

[2] David Hawking, Nick Craswell, and Kathleen Griffiths. “Which search engine is best at finding online services?” In: *WWW posters.* 2001.

[3] Bernard J Jansen, Danielle L Booth, and Amanda Spink. “Determining the informational, navigational, and transactional intent of Web queries”. In: *Information Processing & Management* 44.3 (2008), pp. 1251–1266.

[4] In-Ho Kang. “Transactional query identification in web search”. In: *Asia Information Retrieval Symposium.* Springer. 2005, pp. 221–232.

[5] In-Ho Kang and GilChang Kim. “Query type classification for web document retrieval”. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.* ACM. 2003, pp. 64–71.

[6] Yunyao Li et al. “Getting work done on the web: supporting transactional queries”. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM. 2006, pp. 557–564.

[7] Daniel E Rose and Danny Levinson. “Understanding user goals in web search”. In: *Proceedings of the 13th international conference on World Wide Web.* ACM. 2004, pp. 13–19.

[8] Jinxi Xu and W Bruce Croft. “Query expansion using local and global document analysis”. In: *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM. 1996, pp. 4–11.