Anwesha: A Tool for Semantic Search in Bangla

Arup Das[1], Bibekananda Kundu[2], Lokasis Ghorai[1], Arjun Kumar Gupta[1] and Sutanu Chakraborti[1]

[1]*Department of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai - 600096, India*

[2]*Centre for Development of Advanced Computing, Kolkata - 700091, India*

Abstract

Bangla is a low-resource language that is highly agglutinative, and designing effective search and information retrieval systems over Bangla is quite challenging. This paper presents our explorations toward building অন্বেষা (Anwesha), a prototype for a search engine in Bangla. To the best of our knowledge, this search system is the first such initiative in Bangla that facilitates retrieval of semantically related documents by use of diverse knowledge sources like WordNet, statistical co-occurrences (by way of Latent Semantic Analysis (LSA)) and external knowledge sources like Wikipedia (by way of Explicit Semantic Analysis (ESA)). We also present our efforts to overcome the limitations of existing spell-check and lemmatization approaches in Bangla and integrate them into Anwesha. In addition, we also present methods to explain search results by highlighting keywords that LSA or ESA reckons to be semantically related to the query. Since there is no Gold standard dataset available to evaluate the effectiveness of Bangla information retrieval systems, we have created a dataset containing query document relevance pairs in two distinct domains. We analyze the system's performance on queries having different difficulty levels. Our technique could be adapted to facilitate effective semantic search in other low-resource, highly inflected languages.

Keywords

Bangla Information Retrieval, Query Expansion, tf-idf, Latent Semantic Analysis, Explicit Semantic Analysis, Semantic search for agglutinative language

1. Introduction

Advancements in search engine technology over English are yet to translate to search over documents in Indic languages, which are relatively low-resource. One such language, Bengali (also called Bangla) is a highly agglutinative Indo-Aryan language with more than 160 inflected forms for verbs, 36 forms for nouns, and 24 other forms for pronouns[1]. Bangla has two prominent dialect variations: Sadhu bhasa[1] and Chalit bhasa[2]. Being the fifth most-spoken

The International Conference and Workshop on Agglutinative Language Technologies as a challenge of Natural Language Processing (ALTNLP), June 7-8, Koper, Slovenia

cs20s016@smail.iitm.ac.in (A. Das); bibekananda.kundu@gmail.com (B. Kundu); cs20m033@smail.iitm.ac.in (L. Ghorai); cs20m015@smail.iitm.ac.in (A. K. Gupta); sutanuc@cse.iitm.ac.in (S. Chakraborti)

[1]https://en.banglapedia.org/index.php/Sadhu_Bhasa

[2]https://en.banglapedia.org/index.php/Chalita_Bhasa

native language with 300 million speakers globally, Bangla has witnessed the fastest growth in internet users among the other Indic languages[2]. Given the ongoing efforts to digitise Bengali literary works, there is a pressing need for tools that can facilitate semantic search over these documents. This can also inspire research in Information Retrieval (IR) over other low-resource, highly inflected languages similar to Bangla, such as Assamese, Maithili, Oriya, Hindi, and Manipuri[3]. This paper presents our efforts toward building Anwesha, a Bangla search engine. Though there exist Bangla search engines like *Anwesan*, *Sandhan* and *Pipilika*, we attempt to address their shortcomings through Anwesha.

Owing to Bangla's agglutinative nature, scarce data, shortage of benchmark corpus, absence of gold standard datasets for IR evaluation, and the lack of state of the art Bangla Language Processing tools like stemmer or lemmatizer, Parts of Speech Tagger, Named Entity Recogniser(NER) and sense disambiguation tools, there are several challenges to be resolved in search. Our primary contributions are as follows. We show how Anwesha integrates diverse knowledge sources like WordNet, statistical co-occurrences (by way of Latent Semantic Analysis (LSA)) and external knowledge sources like Wikipedia (by way of Explicit Semantic Analysis (ESA)) for facilitating effective retrieval. We present tools for Bangla spell-checking and lemmatization that overcome the limitations of past approaches; these have been integrated into Anwesha. In addition, we also present methods to explain search results by highlighting keywords that LSA or ESA reckons to be semantically related to the query. Finally, we created a Gold Standard dataset containing human relevance judgements over queries of varying complexity, for evaluation purposes.

In the remainder of the paper, we first discuss the knowledge required to understand the vector space approaches used in Anwesha and the related work done in Bangla Search in section 2. The methodology used to build Anwesha is described in section 3. In section 4, we define the creation of a Gold standard dataset to evaluate the Bangla IR. We present our analysis of Anwesha's performance using different query complexities in section 5. In section 6, we present the plans for further improvement in Anwesha.

2. Background and Literature Survey

Knowledge sources: Anwesha makes use of three vector space approaches for retrieval. The first is a naïve approach where the strength of association of a term to a document is captured using term frequency and inverse document frequency (tf-idf)[3]. The second approach uses LSA[4] to perform dimensionality reduction. Both terms and documents are represented as linear combinations of underlying concepts. This facilitates retrieval of documents that do not explicitly contain the words in the query but share higher-order co-occurrences with the query words. The third approach is ESA[5] which exploits knowledge of Wikipedia. Terms and documents are expressed in terms of underlying concepts as in LSA, but in ESA these concepts are interpretable. Each concept corresponds to a Wikipedia article name. Treating Wikipedia as a corpus, the strength of association of a term to a Wikipedia article is estimated using its tf-idf score with respect to that article.

[3] https://www.uio.no/studier/emner/matnat/ifi/IN4080/h18/lectures/vector1-%281%29.pdf

In addition to these approaches, we also integrated a WordNet-based query expansion that helps users articulate their queries better by adding words related to the query words. The Lesk algorithm, a knowledge-based approach, uses a thesaurus or a dictionary as external knowledge for Word Sense Disambiguation [6]. We have used the IndoWordNet[4] [7], a WordNet for Indic languages, to assign the context-appropriate meanings to words in the query. There are 36346 synsets and 45497 unique words covered for Bangla in the IndoWordNet as of 13 April 2022. A Python-based API called "pyiwn"[5] was used to access the IndoWordNet[6]. We have used the adapted Lesk algorithm [8] as it overcomes the original and simple Lesk algorithm limitations by further adding lemma names from hypernyms, hyponyms, holonyms and meronyms.

Spell-checker: We devised our own Bangla spell-checker that was integrated into Anwesha. There are a few spell-checkers in Bangla. The one devised by Rakib Naushad[9] uses a Unicode dictionary to detect non-word errors and lists all the words with minimum Levenshtein distance as candidate solutions. However, it was unable to perform spell corrections over similar-sounding characters like ন/ Na and ণ/ Ṇa and words having different grapheme representation and similar phonetic utterances like সহজ/ Sahaja(EN: Easy) and শহজ/ Śahaja(EN: Easy), the former being the correct spelling. Our spell-checker handles typographic and orthographic mistakes to perform context-insensitive spelling correction and uses a double Metaphone algorithm to handle phonetic errors. We present the details of our implementation in 3.

Lemmatization: Bengali is a highly inflectional language having 70% inflected words [10]. Reducing words to their roots tends to improve the effectiveness of retrieval. We have surveyed two existing Bengali lemmatizers: BNLTK[7]uses a valid suffix list for nominal inflections and a mapping table for verbal inflections, whereas BIRS[11] uses a Trie data structure for longest prefix matching ; it simultaneously chops characters sequentially from the beginning and end of the input word and attempts to match it with a valid root word from the corpus and finally considers the output with minimum Edit Distance from the input word. However, both of these implementations were inaccurate in several cases; for instance, in the case of BIRS, we got output as উত্তরপদ/ Uttarapada(EN: the answer) instead of উত্তরপ্রদেশ/ Uttarapradesa(EN: Uttar Pradesh) for the input word উত্তরপ্রদেশের/ Uttarapradesera(EN: of Uttar Pradesh) as it failed to detect proper nouns. Similarly, it mapped a noun input word বীমার/Bimara (EN: of Insurance) to a verb মার/ Māra(EN: Beat) instead of mapping it to the correct root word বীমা/ Bimar(EN: Insurance) as it failed to detect the parts of speech. So we developed a new lemmatizer for Anwesha, where we used a combination of valid suffix stripping and Edit Distance mechanism, and achieved more accurate results.

Existing Bangla search engines: *Anwesan*[8] is a digital library and a search engine for the Rabindra Rachanabali collection[12]. It uses Lucene Search Engine Library and the DSpace framework for searching and indexing purposes. Presently the website is inaccessible for exploration. *Sandhan*[9] is a monolingual search engine restricted to tourism and health-related

[4]The official website and the web interface of IndoWordNet: https://www.cfilt.iitb.ac.in/indowordnet/

[5]pyiwn GitHub repository: https://github.com/riteshpanjwani/pyiwn

[6]Python notebook demonstrating usage of pyiwn in accessing IndoWordNet: https://github.com/cfiltnlp/pyiwn/blob/master/examples/example.ipynb

[7]Bengali Lemmatizer by Anirudh Adhikary: https://github.com/banglakit/lemmatizer

[8]http://anwesan.iitkgp.ernet.in

[9]http://sandhan.tdil-dc.gov.in/Search

domains based on the Bag of Words model. It is focused more on enhancing recall compared to precision, and hence the top results do not always appear to be very relevant to the query. Also, Sandhan does not seem to understand user intent appropriately, even for tourism-related queries. For example, a query related to the Taj Mahal not having the Taj Mahal explicitly specified in the query cannot fetch any relevant result. The query "আগ্রায় অবস্থিত এক বিখ্যাত স্মৃতিসৌধ"/ āgrāẏa abasthita ēka bikhyāta smṛtisaudha(EN: A Famous Memorial in Agra) does not return anything relevant to Taj Mahal. Also, there is no spell-check mechanism in Sandhan. *Pipilika*[10] is a search engine launched in Bangladesh on April 13, 2013, and primarily crawl data from Bangla News, Bangla Blogs, and Bangla Wikipedia. It reports data of interest to the residents of Bangladesh. Piplika performs query expansion[13] using a pseudo relevance feedback mechanism. However, unlike Anwesha, it falls short in terms of explicitly incorporating knowledge of statistical co-occurrences, background and linguistic knowledge (as in Wikipedia and IndoWordNet respectively).

3. Proposed Methodology

At a high level, Anwesha has the following three components:
1. **Query and document preprocessing**: The spell-check program detects the non-words in the query and suggests the candidate words to the user. It first transliterates the valid Bangla words obtained from IndicCorp[11] and IndoWordNet into English and ranks the candidate solutions in non-increasing order of their scores calculated using a Bayesian spell-check mechanism[14]. It also uses a double Metaphone algorithm to handle phonetic errors. If the user settings allow for it, the query expansion module adds related terms to the query based on IndoWordNet similarity. The matching algorithm assigns these additional terms lower weights compared to terms in the query. The query and the documents in the test collection are preprocessed in five steps: text normalisation[15][16], elimination of punctuation symbols, word tokenization, stopword removal[12] and lemmatization. The lemmatization process identifies the parts of speech of the words. Then, it removes the nominal suffixes from noun words and finds the verbal inflections for the verbs using a dictionary. If the resulting word is present in the root word corpus, it considers the input word as a lemma; else, it finds the possible candidate keys and outputs the word with minimum edit distance.
2. **Search algorithm and relevance estimation**: Cosine similarity is used over all three vector space approaches discussed before (tf-idf, LSA and ESA) for retrieval and ranking. We observed that a document is often ranked high even if it contains only a few words from the query if those words have a high presence in the document. In order to prefer documents that have more query words over those that have only a few, we have defined the relevance score as the harmonic mean of the cosine similarity and a normalized score based on the number of words from the query that is present in the document. For LSA, 600 concept dimensions were used, as they yielded the best results.
3. **Displaying the top ten relevant results**: We rank the documents in non-increasing order

[10] https://pipilika.com/
[11] https://storage.googleapis.com/ai4bharat-public-indic-nlp-corpora/indiccorp/bn.tar.xz
[12] https://github.com/stopwords-iso/stopwords-iso

Table 1
Definition of the complexity level of a query

Query Type	Complexity Level
The query contains exact words, phrases or sentence from the document.	1
The query is not present as it is in the document. There is a slight deviation.	2
The query is a generalised phrase capturing the overall story or the document's theme.	3
It is a general query not related to any specific document.	4

of their relevance scores and display the top ten documents. We explain the search results by highlighting the keywords in the ranked documents. This is straightforward in the case of tf-idf, the words in the query are highlighted. In the case of LSA, we first represent the concepts both in the query and document as linear combinations of words. Then, we calculate the Hadamard product between these representations and highlight words in the document with the highest coefficients. In ESA, we highlight the words in a document whose representations in the concept space have the highest cosine similarity with the concept representation of the query vector.

4. Gold Standard Dataset Preparation

There are several IR test collections available in English[13]. Unfortunately, there is no Gold standard dataset available to test the effectiveness of Bangla IR. So, we have created a document collection containing 182 short stories, novels and essays written by Rabindranath Tagore[14] and 1000 newspaper articles published in 2013 crawled from the Bangla newspaper Prothom Alo[15]. The collection contains 100 newspaper articles each from one of the ten categories: বাংলাদেশ/ Bānlādēśa(EN: 'Bangladesh'), খেলা/ khēlā(EN: 'sports'), বিজ্ঞান ও প্রযুক্তি/ bijñāna ō prayukti(EN: 'technology'), বিনোদন/ binōdana(EN: 'entertainment'), আন্তর্জাতিক/ āntar-jātika(EN: 'international'), অর্থনীতি/ arthanīti(EN: 'economy'), জীবনযাপন/ jībanayāpana(EN: 'life-style'), মতামত/ matāmata(EN: 'opinion'), শিক্ষা/ śikṣā(EN: 'education') and আমরা/ āmarā(EN:'we-are'). Using a restricted number of documents helped examine the results and focus on precision-oriented measures. Rabindranath Tagore's work has two different dialect variations: Sadhu bhasa(101 documents) and Chalit bhasa(81 documents). Compared to the news articles, the literary documents are very lengthy and constitute 69.32% of the unique words in the vocabulary. We designed 94 queries, 26 queries belonging to complexity levels 1 and 2, 19 queries in complexity level 3 and 23 queries in complexity level 4. The definition of complexity levels is shown in Table 1. We obtained graded relevance judgement from the Bangla users on each of the top ten retrieved documents our search algorithms considered relevant to a query. The users rated every document as either highly relevant (by assigning a score of 3), reasonably or partially relevant (by assigning a score of 2) or irrelevant (by assigning a score of 1). We collected at least five user responses for every query document pair and determined the mean of the user ratings to calculate the final relevance of a document to a query.

[13]http://ir.dcs.gla.ac.uk/resources/test_collections/
[14]https://rabindra-rachanabali.nltr.org/
[15]https://www.prothomalo.com/

Table 2
Performance of tf-idf, tf-idf+Query Expansion and LSA across different query complexity levels @k=10

Complexity Level	Evaluation Measure	tf-idf	tf-idf + Query Expansion	LSA
1	MAP	**0.4**	0.38	0.38
	nDCG	**0.81**	0.76	0.78
	Mean Precision	0.76	0.74	**0.77**
2	MAP	0.4	0.35	**0.43**
	nDCG	**0.83**	0.77	0.82
	Mean Precision	0.73	0.67	**0.77**
3	MAP	0.38	0.31	**0.39**
	nDCG	0.79	0.7	**0.8**
	Mean Precision	0.7	0.62	**0.72**
4	MAP	0.49	0.33	**0.54**
	nDCG	0.75	0.67	**0.73**
	Mean Precision	0.62	0.56	**0.64**

Table 3
Performance of ESA on queries requiring external knowledge @k=10

Approaches	MAP	Mean nDCG	Mean Precision
tf-idf	0.2	0.52	0.46
tf-idf+Query Expansion	0.14	0.46	0.4
LSA	0.24	0.55	0.52
ESA	0.45	0.83	0.78

5. Results and Analyses

There are ten queries from Rabindranath Tagore's work in complexity levels 1 and 4 and seven queries in complexity levels 2 and 3. Further, three queries in complexity level 1 belong to Sadhu bhasa. We take eighteen queries from every complexity level to evaluate Anwesha's performance with respect to mean average precision(MAP), normalized discounted cumulative gain(nDCG) and mean precision. We present our results in Table 2. Lemmatization boosted the MAP scores by 5.56% and 8.31%, and nDCG scores by 3.33% and 4.82% on an average when using tf-idf and LSA respectively. We observe that LSA outperforms tf-idf over MAP and mean precision with the increase in the complexity level of the queries.

In Figure 1(a), we present results showing the effectiveness of query expansion on a representative set of fourteen queries, seven from each of the two categories. The queries from both categories had the same query intent. However, in the second category, some words in the query were substituted with synonymous terms. IndoWordNet augments queries with synonymous terms after lemmatization. This case study helped confirm that IndoWordNet enhances the user query when it does not contain terms that precisely match the content of the relevant documents.

In Figure 1(b), we compare the effectiveness of tf-idf and LSA on seven direct queries and seven indirect queries on related themes. We observe that tf-idf outperforms LSA when the queries are precise. In contrast, LSA consistently outperforms tf-idf when the queries and the retrieved relevant documents do not share many words; instead, they share a common theme.

Figure 2 shows a snapshot of the user interface of Anwesha. The user can choose one of the four options (tf-idf, IndoWordNet based query expansion, LSA and ESA) for retrieval. The figure shows how a query "ফুটবল সংক্রান্ত খবের" / Phuṭabala saṅkrānta khabēra(EN: Football related news) undergoes spelling correction (খবের/khabēra →খবর/ khabara) and the search results are explained. This is achieved by highlighting the keywords in a top retrieved document that LSA or ESA reckon to be semantically related to the query. In Figure 2, the keywords highlighted by LSA are দল/ Dala(EN: team), কোচ/ kōca(EN: coach) and সাফ/ sāpha(EN: SAFF-a famous football tournament). Similarly, beneficiary keywords like গোলে/ gōlē(EN: goal), লিগে/ ligē(EN:

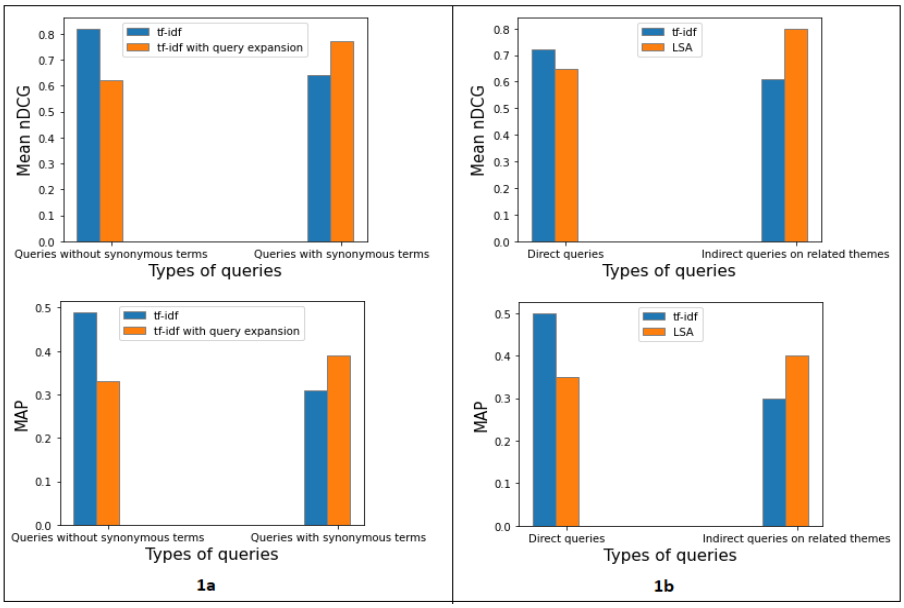

Figure 1: Bar graphs illustrating the (1a) effectiveness of query expansion measured using Mean nDCG and MAP @k=10, and (1b) performance of tf-idf and LSA over direct queries and indirect queries on related themes measured using Mean nDCG and MAP @k=10.

league), ক্যাম্পে/ kyāmpē(EN: camp), মিডফিল্ডার/ miḍaphilḍāra(EN: midfielder), পেলেগ্রিনি/ pēlēgrini(EN: Pellegrini - a Chilean professional football manager) and ক্যাথলিক/ kyāthalika(EN: catholic) were identified in remaining top retrieved documents. Interestingly, many of these words were not explicitly present in the query but are easily seen to be relevant to the query.

Techniques like ESA help in integrating background knowledge from sources like Wikipedia to facilitate more effective retrieval. Table 3 illustrates the effectiveness of ESA on five queries in complexity level 4 that benefit from such background knowledge. We observe that ESA outperforms tf-idf, query expansion using IndoWordNet and LSA by a wide margin.

There is no silver bullet that works the best across all types of queries and user requirements. An approach like tf-idf works the best on a precise query like "পন্টিংয়ের ফার্স্টক্লাস ক্রিকেটের শেষ দিন"/ Panṭinẏēra phārsṭaklāsa krikēṭēra śēṣa dina(EN: The last day of Ponting's first-class cricket). This is a query in complexity level 1 where only one relevant document(2213) has to be retrieved from the corpus. For the same query articulated as a complexity level 2 query such as "রিকি পন্টিংয়ের ক্রিকেট মাঠে শেষ দিন"/ Riki panṭinẏēra krikēṭa māṭhē śēṣa dina(EN: Ricky Ponting's last day on the cricket field), we find that both tf-idf and LSA are successful in retrieving the relevant document. When the query is reformulated as a complexity level 3 query as "রিকি পন্টিংয়ের খেলার মাঠে শেষ দিন"/ Riki panṭinẏēra khēlāra māṭhē śēṣa dina(EN: Last day of Ricky Ponting on the playground) we find tf-idf, LSA and ESA retrieving the relevant document. However, when the query is represented as a complexity level 4 query, for instance, "অস্ট্রেলিয়ার বিশ্বকাপজয়ী ব্যাটসম্যান অধিনায়কের অবসর"/ Aṣṭrēliẏāra biśbakāpajaẏī byāṭasamyāna adhināẏakēra abasara(EN: Retirement of Australia's World Cup-winning batsman captain), it requires background knowledge about Ricky Ponting in that he was an Australian World Cup winning captain and a renowned batsman. Hence only ESA was able to

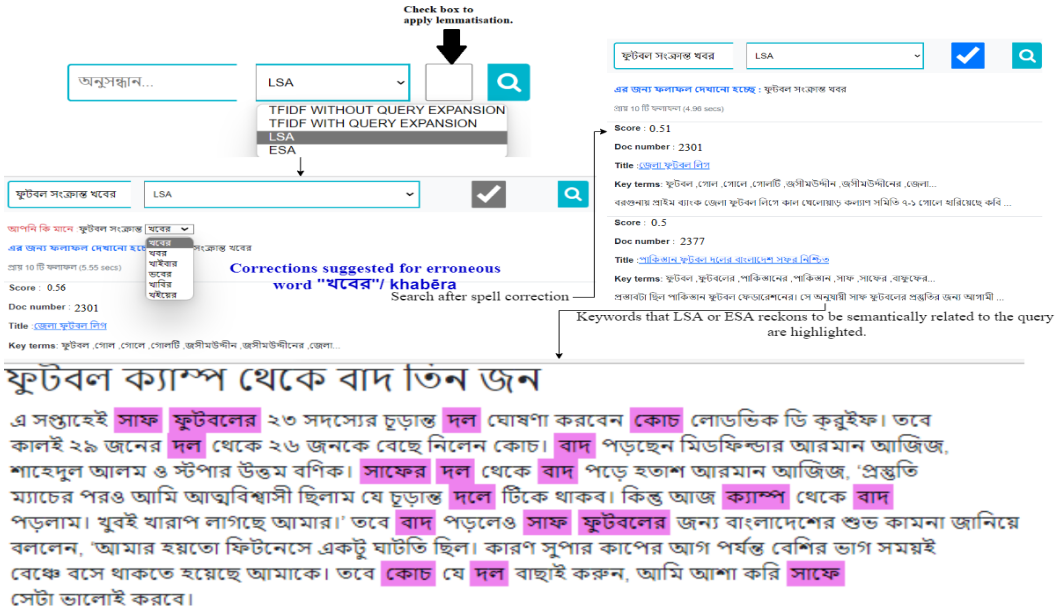

Figure 2: Anwesha's user interface and explanation of search results by highlighting keywords

retrieve the relevant document. All the other three approaches failed here. It may be noted that due to the absence of relevant Wikipedia articles related to the literary works of Rabindranath Tagore, we have used ESA only on the 1000 news articles.

We independently evaluated the success of spell-check and lemmatization. Compared to Rakib Naushad's approach[9] our spell-check algorithm was much faster and it boosted the mean reciprocal rank scores by 17.31% on a list of 2019 misspelt words. Our lemmatizer produced an accuracy of 88% as opposed to 73% by BIRS[11].

6. Conclusion and Future Work

To the best of our knowledge, ours is the first effort to incorporate knowledge of IndoWordNet, Wikipedia and statistical co-occurrences to facilitate semantic search in Bangla and allow for an explanation of retrieved results by highlighting terms reckoned to be relevant to the query by various approaches. We have also compiled relevance judgements over queries at diverse complexity levels to create a Gold Standard dataset for evaluation and used this for systematically analysing our results. Our technique could be adapted to facilitate effective semantic search in other low-resource, highly inflected languages. As part of future work, we intend to use NER which should help in containing indiscriminate query expansion using IndoWordNet which has adversely affected query expansion effectiveness in select cases in our current implementation. We also intend to handle idiomatic or multiword expressions and integrate relevance feedback to further improve the effectiveness of our IR system. We also need to conduct a more thorough evaluation of ESA over a wider class of queries. We have addressed the limitations of existing lemmatization and spellcheck algorithms in our current work. Context-sensitive spellcheck

may be explored in future. The dataset[16], code and a detailed analysis of our work are present here: https://github.com/ArupDas15/Anwesha.

References

[1] S. Bhattacharya, M. Choudhury, S. Sarkar, A. Basu, Inflectional morphology synthesis for bengali noun, pronoun and verb systems, in: In Proceedings of the national conference on computer processing of Bangla, 2005, pp. 34–43.

[2] KPMG, Indian languages- defining india's internet, https://assets.kpmg/content/dam/kpmg/in/pdf/2017/04/Indian-languages-Defining-Indias-Internet.pdf, Accessed: 2021-10-01, 2017.

[3] P. S. Ray, M. A. Hai, L. Ray, Bengali Language Handbook, Center for Applied Linguistics, 1966. URL: https://files.eric.ed.gov/fulltext/ED012914.pdf.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (1990) 391–407.

[5] E. Gabrilovich, S. Markovitch, Computing semantic relatedness using wikipedia-based explicit semantic analysis, in: IJCAI, 2007.

[6] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone, in: Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC '86, Association for Computing Machinery, New York, NY, USA, 1986, p. 24–26. URL: https://doi.org/10.1145/318723.318728. doi:10.1145/318723.318728.

[7] P. Bhattacharyya, Indowordnet, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), 2010, pp. 3785–3792.

[8] S. Banerjee, T. Pedersen, An adapted lesk algorithm for word sense disambiguation using wordnet, in: Procs. of CICLing 2002, 2002, pp. 136–145.

[9] R. Noushad, Bangla spell checker, https://github.com/RakibNoushad/Bangla-Spell-Checker, 2020.

[10] A. Chakrabarty, U. Garain, Benlem (a bengali lemmatizer) and its role in wsd, ACM Trans. Asian Low-Resour. Lang. Inf. Process. 15 (2016). URL: https://doi.org/10.1145/2835494.

[11] M. Kowsher, I. Hossen, S. Ahmed, Bengali information retrieval system (birs) (2019). doi:10.5121/ijnlc.2019.8501.

[12] S. Das, S. Banerjee, P. Mitra, Anwesan: A search engine for bengali literary works, World Digital Libraries 5 (2012) 11–18. doi:10.3233/WDL-120003.

[13] M. R. Islam, J. Rahman, M. R. Talha, F. Chowdhury, Query expansion for bangla search engine pipilika, in: 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 1367–1370. doi:10.1109/TENSYMP50017.2020.9231043.

[14] M. D. Kemighan, K. W. Church, W. A. Gale, A spelling correction program based on a noisy channel model, in: COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics, 1990.

[16] Dataset available in Zenodo: https://doi.org/10.5281/zenodo.6583149

[15] A. Kunchukuttan, The IndicNLP Library, https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf, 2020.
[16] S. Alam, T. Reasat, A. S. Sushmit, S. M. Siddique, F. Rahman, M. Hasan, A. I. Humayun, A large multi-target dataset of common bengali handwritten graphemes, in: International Conference on Document Analysis and Recognition, Springer, 2021, pp. 383–398.