Exposing Knowledge: Providing a Real-Time View of the Domain Under Study for Students

Omar Zammit[1], Serengul Smith[2], Clifford De Raffaele[1], and Miltos Petridis[2]

[1] Middlesex University Malta, Pembroke, Malta
{ozammit, cderaffaele}@ieee.org
[2] Middlesex University, London, UK
{s.smith, m.petridis}@mdx.ac.uk

Abstract. With the amount of information that exists online, it is impossible for a student to find relevant information or stay focused on the domain under study. Research showed that search engines have deficiencies that might prevent students from finding relevant information. To this end, this research proposes a technical solution that takes the personal search history of a student into consideration and provides a holistic view of the domain under study. Based on algorithmic approaches to assert semantic similarity, the proposed framework makes use of a user interface to dynamically assist students through aggregated results and wordcloud visualizations. The effectiveness of our approach is finally evaluated through the use of commonly used datasets and compared in line with existing research.

Keywords: Search Engine Keywords · Similarity Analysis · Text Enrichment

1 Introduction

Confidence in search engines has increased, some Internet users nowadays tend to give high veracity to a website because of its inclusion or high ranking in a search engine result [1]. With the amount of information that exists on-line it is more difficult for users to find accurate information [2] and therefore it is impossible for an Internet user to find information without the use of a search engine [3]. Unfortunately, search engines have deficiencies. Search engines tend to be biased and favour certain web sites over others [4]. Discrepancies between search engine results make it difficult for Internet users to decide which search engines to trust. It is already challenging for Internet users to judge the relevance of on-line content [5] and ignore fake news [6] let alone having such inconsistency and doubts about validity.

By design, search engines are targeting a generic audience and search results might not be suitable for a specific group of people [7] like students. We are focusing on providing a better search experience to students while they are doing research, and try to overcome some of the deficiencies imposed by search engines. To achieve this we are proposing a solution that aims to help students

focus more on the domain they are studying by exposing them to various resources related to the domain. The solution takes into consideration previously searched queries related to the current domain being studied using a combination of similarity analysis techniques. A bag-of-words is created that is used to query third-party APIs to find relevant papers and construct a wordcloud. The proposed solution includes a graphical user interface that will allow students to have a holistic view of the domain being studied.

This paper proposes a framework that takes into consideration the students' personal search history to provide an integrated and comprehensive view of the search domain. Following a brief review in Section 2 on related work, the paper presents in detail the proposed framework design and implementation in Section 3. Section 4 presents and discusses the results obtained from the implementation when the framework was compared with existing solutions. Lastly, a conclusion is drawn in Section 5.

2 Current Solutions

Various solutions are trying to assist students in their study. Some suggest to move away from a search engine and focus on an educational search engine targeting a particular domain [7]. Such approach is problematic since students tend to rely on search engines before libraries to search a new term or when they are unfamiliar with a new topic [8], and therefore it might be challenging to convince them to move away from search engines. Some focused their studies on users URL visited or browsing clicks to understand browsing habits [7, 9–11]. We are taking a similar approach but we are focusing mainly on the keywords visited by the student since these are the entry point of a web search session. A web search session starts with a student issuing a query, the query is processed and the result may surface a website. Students will validate the result and visit the website (see Fig.1)[12]. Entities are bidirectional, students can start with a query and can continue formulating different queries until the required result is obtained [9]. Queries are often appended to URLs [13] and some studies already

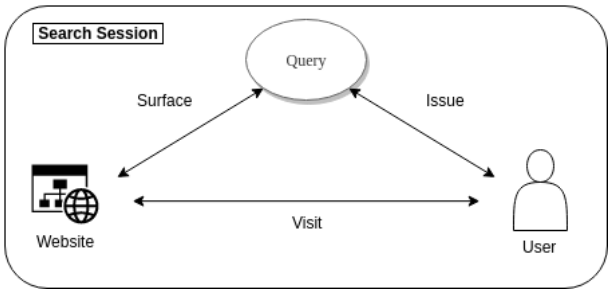

Fig. 1. A web search session as explained in [12]

showed that such data can be used to learn about users browsing behaviour [14] or to understand how students search with the purpose of learning. Usta et al.

(2014) focused on secondary school students search behaviour in a learning environment and compared such behaviour with general web search engines trends. Vidinli and Ozcan (2016) proposed a general modular framework for query suggestion algorithm development to overcome the issue that search engines are targeting a very diverse population. Their research focused mainly on secondary school students since such students have difficulty in formulating queries. The only issue with such a framework is that they are targeting secondary school students only.

Smith et al. (2017) did an exploratory study on query auto-completion usage during a search session with assigned tasks. To monitor user activity they used a modified version of CrowdLogger[3], a Google Chrome extension that collects searched keywords. Keywords where re-submitted using Google QAC API and all `Search Engine Results Pages` (SERP) were scraped and cleaned from images and other elements before displaying them to the user. Such an approach can be used to collect data and evaluate a system that its main aim is to learn more about browsing habits. We opted to take a seamless approach, instead of forcing students to install external plugins, we are reading keywords directly from the browser local history database. In Bast and Weber (2006) the authors mined query logs and used query frequency to predict query completion and rank suggestions [15].

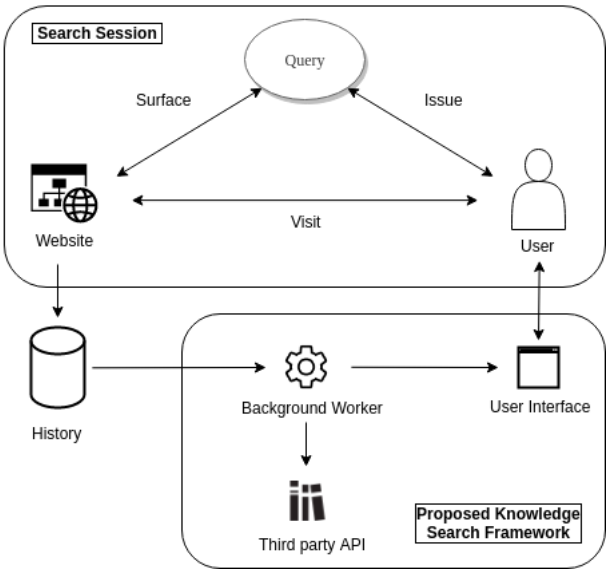

Fig. 2. Proposed framework showing additional components

[3] https://crowdlogger.cs.umass.edu/

3 Proposed Knowledge Search Framework

Keywords search terms play an important role to understand user's psychology [16] and thus we are using keywords searched by the student to understand which domains relate to the student. Past students keywords are taken into consideration since these requests determine the context of the research [17, 18]. To improve student experience we are extending the search session as described in Kim et al. (2012) by adding an additional framework (see Fig. 2). Internet browsers, like Google Chrome, are storing history data on the local computer in SQLite databases [16] for reference. Such databases contain data about URLs visited by the student and keywords searched online.

3.1 Background Worker

We implemented a background worker to extract information from the local history files and predict suitable content. To achieve this the SQLite history file is copied into an accessible location (since this cannot be read while being used by the browser) and SQL statements are executed to extract keywords currently being searched by the student (refer to Listing 1.1 for an SQL example to retrieve data).

Listing 1.1. Query to get all keywords searched by the user from Google Chrome

```
1 select * from urls
2 inner join keyword_search_terms
3 on urls.id = keyword_search_terms.url_id
4 where urls.last_visit_time > ?
5 order by urls.last_visit_time asc limit ?;
```

Text Enrichment using Third-Party API Keywords are not enough to determine what students are searching for. Search queries tend to be short ambiguous and under specified [19]. Having short text is less effective due to its brevity and less sparsity of words and when dealing with such data, enriching the semantics using external entities is essential [20]. Various studies used third party like Wikipedia to enrich text or find similarity between keywords [10, 20–23].

We implemented text enrichment in the background worker so that more insight is known about the domain related to the keywords submitted by the student. For each keyword searched by the student that is stored in the local history, a Google search URL is created (see Listing 1.2) and sent to Google.

Listing 1.2. Google request URL format

```
1 url = f"https://www.google.com.mt/search?" \
2       f"q={keyword}&oq={keyword}&client=ubuntu"
```

The HTML response contains search results in the form of HTML anchor tags consisting of a URL to an external source and a short description

(anchor text). The background worker will parse the response and identify anchor tags. Since results may contain trending and e-commerce anchor tags other than sponsored links [21], Python libraries based on Levenshtein distance[4] were used to determine if the anchor tag is relevant to the keyword being searched. This was done by comparing the anchor tag description to the keyword itself.

Levenshtein distance is the number of deletions, insertions or substitutions required to transform a source string s into a target string t [24]. Computed as:

Step 1: Let n be the length of s and m be the length of t.
Step 2: If $min(n, m) = 0$ then $lev = max(n, m)$. No more steps.
Step 3: Create a matrix d containing $0..m$ rows and $0..n$ columns.
Step 4: Set the first row to $0..n$ and first column to $0..m$.
Step 5: Process each $s[i]$ value from 1 to n
Step 6: Process each $t[i]$ value from 1 to m
Step 7: If $s[i] = t[i]$ then $cost = 0$
Step 8: If $s[i] \neq t[i]$ then $cost = 1$
Step 9: Set $d[i, j]$ as follows:

$$d[i,j] = min \begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases} \quad (1)$$

Step 10: Repeat from step 5 until $d[n, m]$ value is found.
Step 11: $lev = d[n, m]$

The smaller the Levenshtein distance between the keyword and the anchor text description, the more similar the two strings are [24]. Once anchor texts having a high degree of similarity are identified, a web request is done for each anchor tag link and a bag-of-words based on their HTML content is created. Each HTML response obtained from anchor tags link was cleaned as described by Hu et al. (2013), that is, removing HTML tags from the response, identify tokens, removing stop words and eliminating punctuation [25]. The normalization steps [26] done in this research are similar to the steps suggested by Gowtham et al. (2014), the only difference is that we used the Python Natural Language Toolkit, since this includes functions to convert text to tokens, has a list of stop words, can perform part of speech tagging and can convert a word to its lemma [27]. The toolkit contains an implementation of the WordNet lexical database [28] that we used to check the validity of the words since it models the lexical knowledge of an English native speaker and defines Nouns and Verbs in a well-defined hierarchy [29][30][28]. The majority of queries submitted by users over the internet are a structured collection of noun-phrases, in fact, 70% of the query terms are made up mainly of nouns and proper nouns [31] while other words like helping verbs and pronouns are considered as stop-words [21]. As stated by Barr et al. (2008) part-of-speech tagging on query keywords can be significant when

[4] https://github.com/seatgeek/fuzzywuzzy

extracting features in machine learning. We considered this fact and in addition to text normalization, tokens that are not nouns and verbs were removed from the bag-of-words. Once text enrichment is done, a local database is created that stores all keywords searched by the user and their respective bag-of-words. We took this approach so that text enrichment is only done once for a given keyword.

3.2 User Interface

When dealing with large amount of data, the focus point should not just be the collection of data but the analysis and the ability to find meaningful results from it [32]. In order to assist students a user interface was created that will allow the students to view the results and the predictions computed by the background worker (see Figure 3). The user interface is divided as follows:

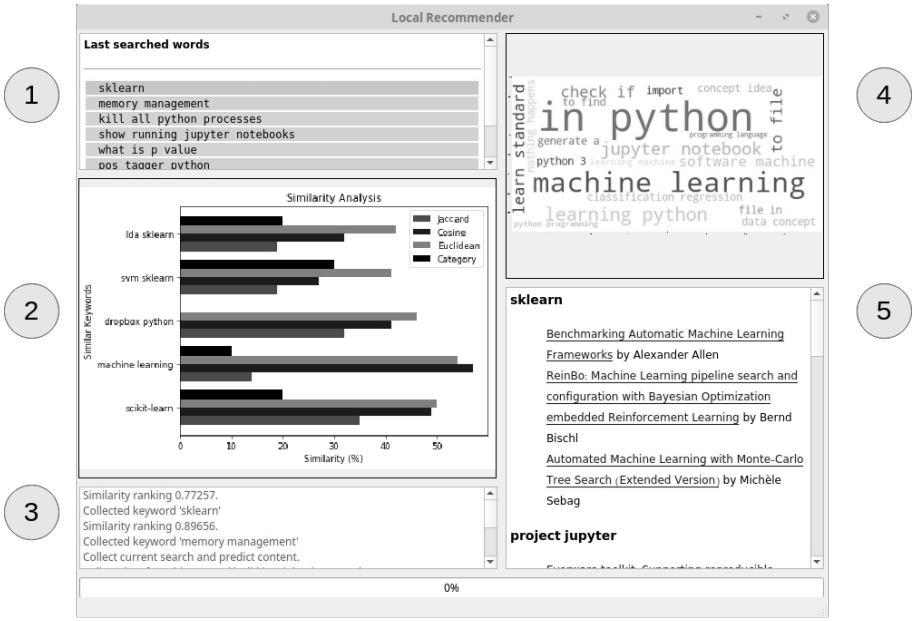

Fig. 3. Evaluation User Interface to Display Predicted Data

1. *Current Search*: Provides a list of the last keywords searched by the student.
2. *Similar Searches*: Previously searched keywords refer to Section 3.3.
3. *System Logs*: Contains system logs.
4. *Domain Wordcloud*: Representation of the most commonly used `ngrams` in the domain being searched.
5. *Academic References*: Paper recommendation.

The framework visually exposes students to new terminologies, by including a wordcloud containing bigrams extracted from the bag-of-words associated with each similar keyword identified. Bigrams were first searched in Wikipedia to ensure their validity and submitted to the Arxiv database [33] to retrieve papers related to the domain being searched by the student.

3.3 Similarity Analysis

Feature Extraction for Similarity Analysis In order to find similar keyword searches similarity analysis was used. Hansen and Jaumard (1997) stated that in order to group data, a dataset $O = \{O_1, O_2, ..., O_n\}$ of N entities is needed [34]. The dataset is made up of the keywords searched by the user and their respective bag-of-words. To classify samples O, one should identify p characteristics of each sample and end up with a matrix X of $N \times p$. Since these characteristics define and will determine the dissimilarities between entities. Fernando et al. (2014) identified various features that helped them in characterization and categorization of Weblogs and other short texts. Most of the features rely on the words (tokens) within the text [35]. For effective transformation and for representation, word frequencies must be normalized in terms of their frequency within a document and within the entire collection [29]. To achieve this Bafna et al. (2016) used TF-IDF with K-means and hierarchical algorithms to classify news, emails and research papers on different topics [36]. TF-IDF was used since this is a technique used to reduce the importance of common terms in a collection so that it ensures that the matching of documents is more influenced by discriminative words having low frequency [29]. The aim is to normalize the words, taking in consideration their frequency within a document and within the entire collection [29]. As described by Erra et al. (2015) TF-IDF measure for a term t will be:

- A higher value when t appears many times within few documents.
- A low value if t appears many times in many documents or fewer times in one document.
- A low value if t appears in all documents.

Let $D = \{d_1, d_2, ..., d_n\}$ be a collection of documents or block of texts. TF-IDF for word t can be computed as follows:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) \tag{2}$$

Where $tf(t, d)$ is the number of instances of t in a document d. And $idf(t, D)$ which is the inverse document frequency can be described as [37]:

$$idf(t, D) = \log_{10} \left(\frac{|D|}{|\{d|t \in d\}|} \right) \tag{3}$$

Where the total amount of documents D is divided by the number of documents containing term t.

In this research each bag-of-word for a given keyword was treated as a document and converted into a TF-IDF matrix.

In order to find similar keywords, we used three similarity measures. Cosine similarity, Euclidean distance and Jaccard similarity. The first two measures take as an input the TF-IDF vector to compute the similarity, while the latter takes the actual bag-of-words. Every time a student searches a new keyword, the background worker will detect the keyword and creates a bag-of-words

and a TF-IDF vector. Cosine similarity, Euclidean distance and Jaccard similarity are computed comparing the searched keyword bag-of-words with existing keywords. Similar keywords are selected as follows:

- Let $C = \{x : x$ keyword having high Cosine similarity$\}$
- Let $J = \{x : x$ keyword having high Jaccard similarity$\}$
- Let $E = \{x : x$ keyword having high Euclidean distance$\}$
- Display only entries from $C \cap J$, $C \cap E$ and $E \cap J$.

Cosine Similarity Cosine similarity can be used with TF-IDF to measure the similarity between two vectors since such measure is suitable because it focuses on the orientation of the document rather than the magnitude [38]. Cosine similarity ranges from -1 (exactly opposite), to 1 (exactly the same) [38].

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \sqrt{\sum_{i=1}^{n} {B_i}^2}} \tag{4}$$

Euclidean Distance If $X_i = (x_{i,1}, ..., x_{i,D})$ and $X_j = (x_{j,1}, ..., x_{j,D})$ are D-dimensional vectors representing two bag-of-words for two keywords that need to be compared. Since distance range can vary, normalization of the result was done using $\frac{1}{1+\eta}$. The Euclidean distance η between both vectors is computed as [39]

$$\eta = \sqrt{\sum_{d=1}^{D} (x_{i,d} - x_{j,d})^2} \tag{5}$$

Jaccard Similarity Jaccard similarity can determine the similarity between two data sets and is computed by dividing the number of features that are common between two datasets by the number of features that are not common [40]. Let A and B be two bag-of-words for two keywords that need to be compared. Jaccard similarity can be computed as.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

Weighted Score Based Aggregation A weighted score based aggregation was used as suggested by Vidinli and Ozcan (2016) to aggregate the three algorithms used and compute the final similarity score. A score is assigned for each keyword searched by the student from the local database. The top 10 keywords having the highest scores are displayed in the user interface and presented to the student as top similar keywords. Each similarity score V was assigned a coefficient k and the final score was computed as.

$$Score = k_{cos} \times V_{cos} + k_{jak} \times V_{jak} + k_{euc} \times V_{euc} \tag{7}$$

4 Evaluation

In order to evaluate the proposed solution and assess the semantic relationship validity, a quantitative methodology was adopted whereby the effectiveness of our approach was mathematically measured and statistically analysed based on datasets used by **Mturk-771**[5][41], **Rel-122**[6][42] and **WordSimilarity-353**[7][17], since these are some of the most commonly used datasets for similarity analysis [43]. Datasets are composed of two terms and their similarity score based on human judgment. A grid search [44] approach similar to the one described by Buitinck et al. (2013) was conducted on each dataset to determine the best similarity coefficient and to validate the results obtained. A nested loop was created that allowed to iterate through 11 coefficient values (from 0 to 2 with 0.2 increments) for each similarity analysis totalling a 11^3 combinations. Within each combination, a similarity score was computed by our system for each word pair in the dataset. Pearson product-moment correlation coefficient [45] and Spearman rank-order correlation coefficient [46] were used to compare the similarity score obtained by our system with respect to the datasets human judgment score. For every grid search iteration, the highest Pearson and Spearman rank was noted together with the iteration coefficients and tabulated in Table 1.

Table 1. Grid search results for all datasets

WordSimilarity-353 grid search results				
Cosine Coefficient	Jaccard Coefficient	Euclidean Coefficient	Pearson	Spearman
1.4	0	0	**0.4568**	0.5664
1.8	0	1.2	0.4370	**0.5687**
Mturk-771 grid search results				
Cosine Coefficient	Jaccard Coefficient	Euclidean Coefficient	Pearson	Spearman
0.8	0	0	**0.4419**	**0.5283**
Rel-122 grid search results				
Cosine Coefficient	Jaccard Coefficient	Euclidean Coefficient	Pearson	Spearman
2	0	0	**0.4389**	0.4699
0.6	0	0.2	0.3389	**0.4796**

Results in Table 1 show that Jaccard similarity did not contribute in improving the accuracy of the analysis, whilst the Euclidean distance contributed in improving the Spearman Correlation for some datasets. In order to evaluate our approach, we compared our similarity results to existing research mainly focusing on the work done by Li et al. (2017). In their research these authors used Wikipedia features to find the similarity between terms and they compared their results with existing benchmarks.

[5] http://www2.mta.ac.il/ gideon/mturk771.html
[6] http://www.cs.ucf.edu/ seansz/rel-122/
[7] http://www.cs.technion.ac.il/ gabr/resources/data/wordsim353/

Table 2. Comparison of Pearson and Spearman as Li et al. (2017)

	Pearson		Spearman	
Dataset	Benchmark	Our System	Benchmark	Our System
Mturk-771	**0.56**	0.44	**0.62**	0.53
Rel-122	**0.64**	0.44	**0.65**	0.47
WordSimilarity-353	**0.56**	0.46	**0.76**	0.57

As shown in Table 2, albeit the obtained results values are lower than the benchmarks identified in Li et al. (2017), this does not imply that the proposed solution is inadequate in its ability to assist students. One should note that the evaluation results are measuring the ability of our system to compute the similarity between two resultant keywords or for a given word search, and only strictly score on the ability to retrieve a preset keyword from the tuples within the dataset. Conversely however, in the context of aiding student learning and searching, our framework is able to take into consideration a range of relevant keywords for each search and to this end our similarity analysis is configured to return top 10 similar keywords. Thus, robustness of returned results is inherently provided through our framework since the top 10 similar keywords returned for each query incorporate both similarities based on human judgement (in line with the dataset) as well as contextually relevant keywords which are extracted through similar analysis which in most instances are not covered by the limited combination of tuples present within these datasets.

5 Conclusion

Retaining focus whilst undertaking research and finding relevant information is proving evermore difficult for students due to current limitations with query formulation and search engines deficiencies. Thus, this paper takes advantage of the browsing history database to extract keywords and try to understand what students are searching for. We proposed a solution that captures the keywords searched by the student in real-time and provides a holistic view of the domain under study. For a given keyword, using various similarity analysis, we are identifying similar previously searched keywords, creating a domain wordcloud and retrieve academic papers related to the domain under study. Results and predictions are aggregated and presented to the student within a user interface that can be used alongside the Internet browser.

The evaluation performed showed that Cosine similarity and Euclidean distance contributed in increasing the accuracy of the proposed solution while Jaccard similarity did not contribute. Moreover, the proposed solution provides a more comprehensive output with respect to current systems since the framework displays the top 10 relevant keywords for each search through the use of a supporting wordcloud. This methodology ensures that students can review both similar data tuples commonly found within datasets as well as contextually relevant keywords pertinent to their research domain.

References

1 Bartlett, J., Miller, C.: Truth, Lies and the Internet a Report Into Young People'S Digital Fluency (2011), http://www.demos.co.uk/files/Truth_-_web.pdf

2 Kraft, R.: A maschine learning approach to improve precision for navigational queries in a Web information retrieval system (2002)

3 Chiru, C.: Search Engines: Ethical Implications. Economics, Management, and Financial Markets **11**(1), 162–167 (2016)

4 Introna, L.D., Nissenbaum, H.: Shaping the web: Why the politics of search engines matters. Information Society **16**(3), 169–185 (2000). https://doi.org/10.1080/01972240050133634

5 Leeder, C.: Student misidentification of online genres. Library and Information Science Research **38**(2), 125–132 (2016). https://doi.org/10.1016/j.lisr.2016.04.003

6 Tredinnick, L., Laybats, C.: Evaluating digital sources: Trust, truth and lies. Business Information Review **34**(4), 172–175 (2017). https://doi.org/10.1177/0266382117743370

7 Vidinli, I.B., Ozcan, R.: New query suggestion framework and algorithms: A case study for an educational search engine. Information Processing and Management **52**(5), 733–752 (2016). https://doi.org/10.1016/j.ipm.2016.02.001

8 Cheng, Y.H., Tsai, C.C.: Online research behaviors of engineering graduate students in Taiwan. Educational Technology and Society **20**(1), 169–179 (2017)

9 Usta, A., Altingovde, I.S., Vidinli, I.B., Ozcan, R., Ulusoy, Ö.: How K-12 students search for learning? Analysis of an educational search engine log. SIGIR 2014 - Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 1151–1154 (2014). https://doi.org/10.1145/2600428.2609532

10 Smith, C.L., Gwizdka, J., Feild, H.: The use of query auto-completion over the course of search sessions with multifaceted information needs. Information Processing and Management **53**(5), 1139–1155 (2017). https://doi.org/10.1016/j.ipm.2017.05.001

11 Feild, H., Allan, J., Glatt, J.: CrowdLogging: Distributed, private, and anonymous search logging. SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 375–384 (2011). https://doi.org/10.1145/2009916.2009969

12 Kim, J.Y., Collins-Thompson, K., Bennett, P.N., Dumais, S.T.: Characterizing Web content, user interests, and search behavior by reading level and topic. In: WSDM 2012 - Proceedings of the 5th ACM International Conference on Web Search and Data Mining. pp. 213–222. ACM, New York, NY, USA (2012). https://doi.org/10.1145/2124295.2124323

13 West, A.G., Aviv, A.J.: Measuring privacy disclosures in URL query strings. IEEE Internet Computing **18**(6), 52–59 (2014). https://doi.org/10.1109/MIC.2014.104

14 Tikhonov, A., Prokhorenkova, L.O., Chelnokov, A., Bogatyy, I., Gusev, G.: What can be found on the web and how: A characterization of web browsing patterns. Proceedings of the 2015 ACM Web Science Conference pp. 1–10 (2015). https://doi.org/10.1145/2786451.2786468

15 Bast, H., Weber, I.: Type less, find more: Fast autocompletion search with a succinct index. In: Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. vol. 2006, pp. 364–371. ACM (2006)
16 Rathod, D.: Web Browser Forensics: Google Chrome Available Online at www.ijarcs.info. International Journal of Advanced Research in Computer Science **8**(December), 5–9 (2017). https://doi.org/10.26483/ijarcs.v8i7.4433
17 Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. Proceedings of the 10th International Conference on World Wide Web, WWW 2001 **20**(1), 406–414 (2001). https://doi.org/10.1145/371920.372094
18 Bharat, K.: SearchPad: explicit capture of search context to support Web search. Computer Networks **33**(1), 493–501 (2000). https://doi.org/10.1016/S1389-1286(00)00047-5
19 Pound, J., Hudek, A.K., Ilyas, I.F., Weddell, G.: Interpreting keyword queries over web knowledge bases. In: ACM International Conference Proceeding Series. pp. 305–314 (2012). https://doi.org/10.1145/2396761.2396803
20 Shirakawa, M., Nakayama, K., Hara, T., Nishio, S.: Wikipedia-Based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes. IEEE Transactions on Emerging Topics in Computing **3**(2), 205–219 (2015). https://doi.org/10.1109/TETC.2015.2418716
21 Rajeshwarkar, A., Nagori, M.: Optimizing Search Results using Wikipedia based ESS and Enhanced TF-IDF Approach. International Journal of Computer Applications **144**(12), 23–28 (2016). https://doi.org/10.5120/ijca2016910498
22 Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07. pp. 787–788 (2007). https://doi.org/10.1145/1277741.1277909
23 Ferragina, P., Scaiella, U.: TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In: International Conference on Information and Knowledge Management, Proceedings. pp. 1625–1628 (2010). https://doi.org/10.1145/1871437.1871689
24 Haldar, R., Mukhopadhyay, D.: Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. Computing Research Repository - CORR (2011), http://arxiv.org/abs/1101.1232
25 Hu, X., Tang, J., Gao, H., Liu, H.: Unsupervised sentiment analysis with emotional signals. In: WWW 2013 - Proceedings of the 22nd International Conference on World Wide Web. pp. 607–617 (2013). https://doi.org/10.1145/2488388.2488442
26 Gowtham, S., Goswami, M., Balachandran, K., Purkayastha, B.S.: An approach for document pre-processing and K Means algorithm implementation. In: Proceedings - 2014 4th International Conference on Advances in Computing and Communications, ICACC 2014. pp. 162–166. IEEE (2014). https://doi.org/10.1109/ICACC.2014.46
27 Loper, E., Bird, S.: NLTK: The Natural Language Toolkit (2002). https://doi.org/10.3115/1118108.1118117, http://arxiv.org/abs/cs/0205028
28 Kilgarriff, A., Fellbaum, C.: WordNet: An Electronic Lexical Database. Language **76**(3), 706 (2000). https://doi.org/10.2307/417141
29 Patil, L.H., Atique, M.: A novel approach for feature selection method TF-IDF in document clustering. In: Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013. pp. 858–862. IEEE (2013). https://doi.org/10.1109/IAdCC.2013.6514339

30 Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11), 39–41 (1995). https://doi.org/10.1145/219717.219748
31 Barr, C., Jones, R., Regelson, M.: The linguistic structure of English web-search queries. In: EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1021–1030. EMNLP '08, Association for Computational Linguistics, Stroudsburg, PA, USA (2008). https://doi.org/10.3115/1613715.1613848
32 Aljrees, T., Shi, D., Windridge, D., Wong, W.: Criminal pattern identification based on modified K-means clustering. In: Proceedings - International Conference on Machine Learning and Cybernetics. vol. 2, pp. 799–806. South Korea (2017). https://doi.org/10.1109/ICMLC.2016.7872990
33 McKiernan, G.: arXiv.org: the Los Alamos National Laboratory eprint server. International Journal on Grey Literature **1**(3), 127–138 (2000). https://doi.org/10.1108/14666180010345564
34 Hansen, P., Jaumard, B.: Cluster analysis and. Mathematical Programming **79**(1-3), 191–215 (oct 1997). https://doi.org/10.1007/BF02614317
35 Perez-Tellez, F., Cardiff, J., Rosso, P., Pinto, D.: Weblog and short text feature extraction and impact on categorisation. Journal of Intelligent and Fuzzy Systems **27**(5), 2529–2544 (2014). https://doi.org/10.3233/IFS-141227
36 Bafna, P., Pramod, D., Vaidya, A.: Document clustering: TF-IDF approach. International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016 pp. 61–66 (2016). https://doi.org/10.1109/ICEEOT.2016.7754750
37 Erra, U., Senatore, S., Minnella, F., Caggianese, G.: Approximate TF-IDF based on topic extraction from massive message stream using the GPU. Information Sciences **292**, 143–161 (2015). https://doi.org/10.1016/j.ins.2014.08.062
38 Chaithanya, K., Reddy, P.V.: A Novel approach for Document Clustering using Concept Extraction. International Journal of Innovative Research in Advanced Engineering (IJIRAE) **3** (2016), www.ijirae.com
39 Mesquita, D.P., Gomes, J.P., Souza Junior, A.H., Nobre, J.S.: Euclidean distance estimation in incomplete datasets. Neurocomputing **248**, 11–18 (2017). https://doi.org/10.1016/j.neucom.2016.12.081
40 Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of jaccard coefficient for keywords similarity. In: Lecture Notes in Engineering and Computer Science. vol. 1, pp. 380–384 (2013)
41 Halawi, G., Dror, G., Gabrilovich, E., Koren, Y.: Large-scale learning of word relatedness with constraints. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1406–1414. ACM (2012). https://doi.org/10.1145/2339530.2339751
42 Szumlanski, S., Gomez, F., Sims, V.K.: A new set of norms for semantic relatedness measures. In: ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. vol. 2, pp. 890–895 (2013)
43 Li, P., Xiao, B., Ma, W., Jiang, Y., Zhang, Z.: A graph-based semantic relatedness assessment method combining wikipedia features. Engineering Applications of Artificial Intelligence **65**, 268–281 (2017). https://doi.org/10.1016/j.engappai.2017.07.027
44 Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013), http://arxiv.org/abs/1309.0238

45 Dillon, M.: Introduction to modern information retrieval (1983). https://doi.org/10.1016/0306-4573(83)90062-6
46 Spearman, C.: The Proof and Measurement of Association between Two Things. The American Journal of Psychology **100**(3/4), 441 (1987). https://doi.org/10.2307/1422689