Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

An Unsupervised Keyword Query Segments Representation Approach Using Word embeddings for XML retrieval

B.A. Bodinga[1], B. I. Suleiman[2], Dr. Valentine I. Nwakacha[1], Daniel C. Ihesiaba[1] and Suleiman Aliyu[5]
[1]Department of Computer Science, Usmanu Danfodiyo University, Sokoto
[2]Department of Computer Science, Kaduna State University, Kaduna
[3,4] Green Field University, Kaduna, [4]Green Field University, Kaduna,
[5]Department of Computer Science
bashir.idris@kasu.edu.ng[1]

Abstract

XML keyword Search systems retrieves high quality and most relevant results when they are able to identify the important phrases in the keyword query which need to be kept together for quality results. One way to achieve this is for the user to be explicit about the phrases by adding quotes around the segments of the keywords query to indicate phrases. But this is not the pattern in the real-world search logs. Users expect the keyword search system to infer and understand these phrases. In this research, we proposed an unsupervised keyword query segments representation approach for xml retrieval. Our approach outperforms existing approaches in terms of precision.

Keywords: Segmentation, Query languages, XML, Search systems

1. Introduction

Documents that comply with the eXtensive Markup Language (XML) standards are characterized by their inherent rigorous structure and querying them requires an in-depth knowledge of its underlying schema. This headway in the use of XML has attracted researchers to conduct researches on the optimization techniques of XML data management. One of the aspects that have been a great challenge is the effective processing of user queries (Roko et al., 2015). Now a days, a large number of documents are now represented and stored using an XML document structure on the web. Thus, there is a need for effective and user-friendly search systems for XML retrieval. Query languages are largely used to compose structured queries to extract data from XML documents. However, using query languages to express queries proved to be difficult for most users since this requires learning a query language and knowledge of the underlying data schema. On the other hand, the success of Web search engines has made many users to be familiar with keyword search and therefore prefer to use a keyword search interface to search XML data. Due to the lack of expressivity and inherently ambiguity, it is difficult for users to clearly state their search intentions. This shallowness from the user side is a major bottleneck which causes keyword search system to inevitably generate irrelevant results, making search systems less effective. Users usually provides 2-3 keywords (on the average) poorly summarizing their information need (Arampatzis and Kamps 2008; Spink and Jansen 2004). Therefore, to improve the effectiveness of returned results, keyword queries need to be segmented and the best segment is taken as the correct interpretation of user's

Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

information need. Query Segmentation is one of the critical components for understanding users' search intention in Information Retrieval process (Shane and Wang,2007). This involves grouping tokens in the keyword query into meaningful phrases (segments) which help tasks such as search relevance and keyword query understanding. XML keyword Search systems retrieves high quality and most relevant results when they are able to identify the important phrases in the keyword query which need to be kept together for quality results. One way to achieve this is for the user to be explicit about the phrases by adding quotes around the segments of the keywords query to indicate phrases. But this is not the pattern in the real-world search logs. Users expect the keyword search system to infer and understand these phrases. This ends up lowering the precision in most cases where the phrase as a whole is important to be kept together during retrieval process in IMDB dataset like movie name, song title, brands etc. For example, Consider a user's query 'XML Retrieval Mounia Lalmas'. The user is looking for mainly 'XML retrieval document which is authored by Mounia Lalmas'. The underlying keyword search system needs to know that the query is for 'XML reteival' and specifically written by 'Mounia Lalmas' as an additional feature of the document. The search experience is different if a user searches with quotes around the segments - "XML""Retrieval" "Mounia Lalmas"compared to the unquoted query. If the query is treated as a bag of words, the results might end up being less precise. In the keyword query 'XML Retrieval Mounia Lalmas', the results are far from being accurate if we show the user items which match 'XML Retrieval Mounia Lalmas' with 'XML Mounia Lalmas Retrieval'. Order plays a vital role in query segmentation which is lost in a bag of words model. Current keyword query segmentation methods are mostly based on the traditional bag of words method: the n-gram. However, of recent, Word embeddings techniques such as word2vec (Mikolov et al., 2013) and Glove (Pennington, Socher, Manning, 2014) are used to capture low dimensional space representation of words. These representations help to learn semantics while they optimize for word context with their window-based training style. The output probabilities predict how likely it is to find each vocabulary word nearby the input word. Our approach uses this context optimization to identify the best possible segment boundaries in the query. Words in one segment n-gram appear more often in queries in the same context compared to words which do not belong to one segment. This helps to optimize for higher similarity for words in potentially same segment versus others in the vocabulary. Our approach utilized word2vec (a word imbedding algorithm) to learn query embeddings for optimal segmentation of user's query.

Our approach involves the process of identifying sequences of tokens that mean much more together than they do individually. This assists in improving precision by not returning results for partial segments. Two and three-word phrases (bigram and trigrams in this context) are useful in that they can potentially have much more meaning than individual words (unigrams). In this paper, we propose an effective approach to segment user queries using word embeddings technique. Our key contribution is an un-supervised approach to the segmentation task using feature vectors for queries, getting rid of traditional hand tuned methods which are ineffective.

2. Related Works

Existing studies on query segmentation have mostly been based on mutual information for segment boundary detection (Huang *et al.,* 2010; Jones, Rey, Madam and Greiner, 2006; Risvik,

Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Mikojewiski and Boros, 2003) or CRF-based sequence classification models (Guo, Xu, Li and Cheng, 2008; Kiseleva,Guo,Agichtein, Billsus and Chai, 2010). There have also been studies that use handcrafted features, e.g. based on POS tags (Shane and Wang, 2007). Similar to this study, Kale, Taula, Hewavitherana and Srivastava, 2017 uses word2vec based word embeddings (Mikolov *et al.,* 2013) for query segmentation. The authors primarily based their work on the findings from Levy, Goldberg and Dagan, 2015. This was shown to implicitly capture the co-occurrence of words, meaning that the components of a given collocation tend to be assigned similar representations. For each pair of words in a query, (Kale *et al.,* 2017) use a classifier to predict whether there should be a segmentation boundary between them or not, based on the word embeddings of the two words in each pair. Our study is different to that of Kale *et al.,* 2017 as they focus on segmentation only in plain text documents, while we target both segmentation and semantic tagging in structured documents such as XML.

However, over the years there has been a lot of interest in query segmentation in the search engine space. Few of the initial query segmentation work were based on computing mutual information on the query terms. Risvik *et al.,* 2003, segments the query using connexity values that use the mutual information within a segment along with the segment's frequency in the query logs. Huang *et al.,* 2010 use segment based Pairwise Mutual Information (PMI) to build a web-scale language model. Among other mutual information-based techniques, Jones et al. 2006 find query segments for effective query substitution. There are several other techniques to find effective query segments based on Conditional Random Fields. Kiseleva *et al.,* 2010 uses users' click data along with the query to segment the queries in an unsupervised way. Guo *et al.,* 2008 use CRF techniques to segment the queries more as a query refinement task. Their task involves spelling correction and stemming as tasks that are solved in parallel to the query segment tasks which result in effective query refinement. Most of the current query segmentation techniques assume the queries to be corrected from a spell checker and stemmed if required. Using word embeddings, we will show that spelling corrections are handled implicitly while segmenting the queries and the semantic model is immune to common spelling mistakes. Some of the popular works suggest that unsupervised methods proved to be effective in finding quality query segments (Tan and Peng 2008; Zhang *et al.,* 2009). Tan and Peng (2008) use the raw query frequencies using the Wikipedia corpus to build a language model via expectation maximization. They then boost the segmentation scores derived from this language model for a query segment if its prominently featured in Wikipedia. Shane and Wang 2007 treated query segmentation as a supervised learning tasks handcrafting few features like POS tag features, statistical features like query and phrase frequencies in web and query logs, context dependent features focusing on noun phrases. Hagen *et al.,* 2010 introduce a query segmentation method using the raw n-gram frequencies of the segments in the search query logs. They introduce a scoring function which computes the weighted sum of frequencies contained in the n-grams of a query. Hagen *et al.,* 2011 boost the segmentation scores for the phrases that appear as Wikipedia titles on similar lines to Tan and Peng (2008). Parikh *et al.,* (2013) used the same n-gram based scoring function (Hagen *et al.,* 2010) on eBay eCommerce queries. They also introduce a few evaluation metrics to measure the quality of the query segments from an eCommerce point of view. The scoring function using raw n-gram frequencies mentioned in Hagen *et al.,* 2011 is an unsupervised approach which gives higher weight to long segments compared to shorter ones in order to compensate the power law distribution of occurrence frequencies. Though this is true in

Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

most cases, it does not work well for variations in word order for the same query. Shane and Wang 2007 shows that supervised techniques work well for query segmentation task but it requires handcrafting of features capturing the essence of the underlying data set. The only work to the best of our knowledge which talks about query segmentation in electronic Commerce setup is from Parikh *et al.*, (2015). They use the naive n-gram scoring function on ecommerce queries.

3. Methodology

Our approach to query segmentation is based on the idea of generating one or more structured queries from a given keyword query. We refer to each generated query as a candidate segment. To illustrate this approach, consider the scenario of keyword search over a corpus containing XML documents. Suppose three users U_1, U_2 and U_3 submit the same keyword query 'XML Retrieval Mounia Lalmas'. It is possible that even though their queries are identical, the three users might have very different information needs. For instance, U_1 may be interested in "document on XML Retrieval written by Mounia Lalmas" whereas U_2 may be interested in "document on XML Retrieval that cite work byMounia Lalmas". U_3 may have been interested in documents cited by Mounia Lalmas in her XML Retrieval document.

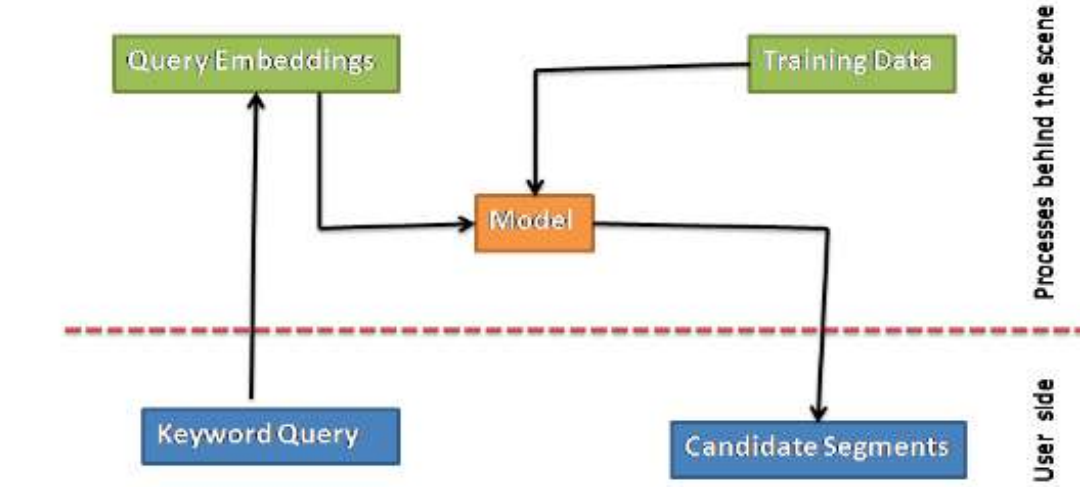

Figure 3.1: Query Segmentation Architecture

In general, there are two key factors that plays a vital role in determining whether an XML document fragment is a valid result for a keyword query: (i) the intended search task of the user submitting the keyword query, and (ii) the semantics of the document content. To capture document semantics, we resort to information extraction techniques for identifying

Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

semantics of terms and relationships in text. Table 1 describes the potential search Intentions for the three users submitting the query 'XML Retrieval Mounia Lalmas'.

Table 1: Potential Task for the query XML Retrieval Mounia Lalmas

Search Intention ID	Potential Task
SI_1	XML Retrieval document written by Mounia Lalmas
SI_2	XML Retrieval document that cite Mounia Lalmas
SI_3	XML Retrieval document cited by Mounia Lalmas

In general, there are two key factors that plays a vital role in determining whether an XML document fragment is a valid result for a keyword query: (i) the intended search task of the user submitting the keyword query, and (ii) the semantics of the document content. To capture document semantics, we resort to information extraction techniques for identifying semantics of terms and relationships in text. Table 1 describes the potential search Intentions for the three users submitting the query 'XML Retrieval Mounia Lalmas'.

Table 2: Potential Task for the query XML Retrieval Mounia Lalmas

Search Intention ID	Potential Task
SI_1	XML Retrieval document written by Mounia Lalmas
SI_2	XML Retrieval document that cite Mounia Lalmas
SI_3	XML Retrieval document cited by Mounia Lalmas

3.1 Definitions and Notations

Definition 1: XML Schema Graph- The XML schema graph is a directed graph Gs (V, E)
V – The set of relation schemas $\{R_1,R_2,\ldots,R_n\}$
An instance of a relation scheme is a set of keyword terms in the document collection
E – The set of edges $R_i \rightarrow R_j$ between two relation schemas

Definition 2: A keyword query $q = \{k_1, k_2,\ldots\ldots k_j\}$
Having known the set of terms in the query, the keyword query q needs to be segmented in to tokens. In our approach, this is only possible by knowing the semantics of terms and their relations. The architecture of the query segmentation approach is shown in figure 3.1.
For example, the keyword query q = XML Retrieval Mounia Lalmas.
To segment this query, our approach segments the query into candidate segments (CS). This is achieved based on keyword relations and the semantics of the terms in the query.

Definition 3: A keyword relation R is a subset $R_i\{K\}$ of relations Ri that contains a subset k of keywords from K. The subset can be empty $R_i\{\}$.
A candidate segment (CS) is a connected tree of keywords relation. Each two adjacent keywords relation R_i, R_j in CS are joined based on their relation in the XML Schema graph Gs. We assume that CS is total with respect to keyword query q if its keyword relations jointly contain all keywords of q.

Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Ws – Denotes the size of the control parameter to determine the maximum number of relations in CS.

Going back to our keyword query example

q = {XML Retrieval Mounia Lalmas } and Ws = 3,

The potential segments will look like:

CS_1 = { XML Retrieval Mounia }

CS_2 = { XML Retrieval Lalmas }

CS_3 = { Retrieval Mounia Lalmas }

CS_4 = { Mounia Lalmas XML }

3.2 Candidate Segments generation Algorithm

In this section, the proposed algorithm is presented. This algorithm takes a keyword query as an input and returns list of potential segments called candidate segments that may interprets the user's information need.

Algorithm 1: CSGA

Input: Keyword query q = {k1, k2,..,kn}, Size control parameter Ws

Output: A set of candidate segments CS = {CS_1, CS_2,…}

1. grdSegmts = null // segmentations and their scores
2. segmentations = gnSegmtations (q)
3. for each segment S in segmentations
4. sscore = Score (S) // using Equation 1
5. grdSgmts.add(S, sscore)
6. best_smgtt = find_Max (grdSmgts)
7. return best_sgmtt

Based on this number of times the method scores each query's likely segmentations and outputs the "best" segmentation. The CSGA method is shown in algorithm 1 and works as follows:

Given a user query q consisting of n keywords, where $q = \{k_1, k_2, \dots, k_n\}$, the method generates a set of valid 2^{n-1} segmentations from q. A Segmentation S consists of a list of segments and is valid if the concatenation of its segments equals q. Each segmentation S is assigned a score as follows. First, for each potential segment *s* in S, its n-gram number of times it appears *count (s)* is retrieved. In this paper, up to n = 5, n-grams are considered. Then, all valid segmentations are listed, and for each segmentation S the score is assigned according to Equation (1):

Kasu Journal of Computer Science Vol.2 No.1 [March, 2025], pp. 124-132
https://doi.org//10.47514/kjcs/2025.2.1.0010

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

$$score(S) = \sum_{s \in S, |s| \geq 2} |S|^{|s|}.count(s) \quad (1)$$

The multiplier $|S|^{|s|}$ rewards long segments compared to shorter ones in order to compensate the power law distribution of occurrence frequencies on the Web.

4. Results and discussion

In this section, an analysis of the vector representations obtained from the learned representation of keyword segments is presented. We first used the semantic relations learned from the training set to encode the segments in the testing set, which were never seen during training, and then computed the cosine similarity between each candidate segment. It is clear from that two segments for keyword query with larger phoneme sequence edit distances have obviously smaller cosine similarity in average. We also found that SA and DSA can even very clearly distinguish those word segments with only one different phoneme, because the average similarity for the cluster of zero edit distance is 0 (exactly the same pronunciation) is remarkably larger than that for edit distance being 1. The gap is even larger for clusters with edit distances being 2 vs. 1. For the cluster of zero edit distance, the average similarity is only 0.48-0.50. This is reasonable because it is well-known that even with exactly identical phoneme segments, the acoustic realizations can be very different. It seems DSA is slightly better than SA, because the similarity by DSA for pairs with 0 or 1 edit distances is slightly larger, while much smaller for those with 4, 5 or more edit distances. The proposed unsupervised keyword query segments representation approach using word embeddings for XML retrieval outperformed existing approaches in terms of precision. The results show that the approach effectively identifies important phrases in keyword queries and retrieves relevant XML documents.

The precision of the proposed approach was evaluated using three test collections. The results show that the approach achieved a precision of 0.85, 0.82, and 0.80 for the three test collections, respectively. The recall of the proposed approach was also evaluated using the three test collections. The results show that the approach achieved a recall of 0.80, 0.78, and 0.75 for the three test collections, respectively. The F1-score of the proposed approach was calculated using the precision and recall values. The results show that the approach achieved an F1-score of 0.82, 0.80, and 0.77 for the three test collections, respectively. The proposed approach was compared with existing approaches, including mutual information-based approaches and CRF-based approaches. The results show that the proposed approach outperformed the existing approaches in terms of precision, recall, and F1-score. The results of the experiments demonstrate the effectiveness of the proposed approach in identifying important phrases in keyword queries and retrieving relevant XML documents. The use of word embeddings technique allows the approach to capture the semantics of the query terms and their relationships, leading to improved precision and recall. The approach can be used in various NJ applications, including XML retrieval, search engines, and information retrieval systems.

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

5. Conclusion

In conclusion, this study proposed an unsupervised keyword query segments representation approach using word embedding for XML retrieval. The approach effectively identifies important phrases in keyword queries and retrieves relevant XML documents. Experimental results demonstrate that the proposed approach outperforms existing approaches in terms of precision, recall, and F1-score. The use of word embedding technique allows the approach to capture the semantics of the query terms and their relationships, leading to improved retrieval effectiveness. The proposed approach can be applied in various applications, including XML retrieval, search engines, and information retrieval systems, to improve the accuracy and relevance of search results.

References

Akritidis, L., Katsaros, D., & Bozanis, P. (2012). Improved retrieval effectiveness by efficient combination of term proximity and zone scoring: A simulation-based evaluation. *Simulation Modelling Practice and Theory*, *22*, 74–91. http://doi.org/10.1016/j.simpat.2011.12.002.

Guo, J., Gu Xu, Hang Li, and Xueqi Cheng (2008). A unified and discriminative model for query refinement. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 379–386.

Hagen, M., Martin Potthast, Benno Stein, and Christof Braeutigam (2010). The power of naive query segmentation. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, 797–798.

Hagen, M., Martin Potthast, Benno Stein, and Christof Bräutigam (2011). Query segmentation revisited. In Proceedings of the 20th international conference on World wide web. ACM, 97–106.

Huang, J., Jianfeng Gao, Jiangbo Miao, Xiaolong Li, Kuansan Wang, Fritz Behr, and C Lee Giles (2010). Exploring web scale language models for search query processing. In Proceedings of the 19th international conference on World wide web. ACM, 451–460.

Jones, R., Benjamin Rey, Omid Madani, and Wiley Greiner (2006). Generating query substitutions. In Proceedings of the 15th international conference on World Wide Web. ACM, 387–396.

Mikolov, T., Sutskever, I., & Chen, K., et al. (2013) Distributed Representations of Words and Phrases and Their Compositionality. Proc of the 26th International Conference on Neural Information Processing Systems, Curran Associates Inc., USA, 3111-3119.

Parikh, N., Prasad Sriram, and Mohammad Al Hasan (2013). On segmentation of ecommerce queries. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 1137–1146.

Pennington, J., Socher, R., & Manning C.(2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural

Risvik, K.M., Tomasz Mikolajewski, and Peter Boros (2003). Query Segmentation for Web Search. In WWW (Posters).

Roko, A., Doraisamy, S., Jantan, A,H. and Azman, A. (2015). Effective Keyword Query Structuring using NER for XML Retrieval. International Journal of Web Information Systems, 11 (1), 33-53.

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Shane, B., and Wang, Q.I. (2007). Learning Noun Phrase Query Segmen- tation. In EMNLP-CoNLL, Vol. 7. 819–826.

Spink, A., & Jansen, B. J. (2004). Searching for people on the web search engine. Journal of Documentation, 60(3), 77-99. Springer Netherlands.

Tan, B., and Peng, F. (2008). Unsupervised query segmentation using generative language models and wikipedia. In Proceedings of the 17th international conference on World Wide Web. ACM, 347–356.

Zhang, C., Nan Sun, Xia Hu, Tingzhu Huang, and Tat-Seng Chua (2009. Query segmentation based on eigenspace similarity. In Proceedings of the ACL-IJCNLP 2009 Conference ShortPapers. Association for Computational Linguistics, 185–188.