International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC) (12th&13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

NLP based Optimization Technique for Performance Improvement of knn

Tejaswini S Channe, Manoj B Chandak
Department of Computer Science, Nagpur University, India
tejuchanne@gmail.com
Department of Computer Science, Nagpur University,India
chandakmb@gmail.com

ABSTRACT
Conventional techniques such as spatial inverted index used for nearest neighbor search extends the conventional inverted index to cope with multidimensional data, and also able to answer the nearest neighbor queries with keywords in real time, which, as shown in this paper, has a few deficiencies that seriously impact its performance. Motivated by this, I develop a new access method based on natural language processing that improves the performance of k nearest neighbor search.
Keywords - NLP, Nearest Neighbour Search, Keyword Search, knn, SPI.

I. INTRODUCTION

Conventional techniques, such as spatial inverted index and nearest neighbour retrieval, find objects satisfying both a spatial predicate, and a predicate on their associated texts Spatial inverted index is optimized for multidimensional points. It incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. There are easy ways to support queries that combine spatial and text features. But there are many disadvantages of these straightforward approaches. So I develop a new technique which is based on NLP to improve the performance of nearest neighbour search. In this paper I propose a new technique which is based on NLP that improves the performance of nearest neighbor search. Traditional techniques are unable to detect meaning of the words in different sentences, for example, consider two sentences, "break the glass" and "lunch break", here in first sentence break is verb and in second sentence break is noun. In proposed technique Stanford pos tagger and Chunker are used to solve the above problem.

II. LITERATURE SURVEY

The best method to date for nearest neighbor search with keywords is due to Felipe et al. [2]. They integrate two well- known concepts: R-tree, a popular spatial index, and signature file [6], an effective method for keyword-based document retrieval. By doing so they develop a structure called the IR2-tree [2], which has the strengths of both R-trees and signature files. Like R-trees, the IR2- tree preserves objects' spatial proximity, which is the key to solving spatial queries efficiently. On the other hand, like signature files, the IR2-tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined. But the IR2-tree, inherits a drawback of signature files that is false hits.

A. Spatial Inverted Index

The spatial inverted list (SI-index) is essentially a compressed version of an I-index with embedded coordinates. Query processing with an SI-index can be done either by merging or together with R-trees in a distance browsing manner.

1) The Compression Scheme: Compression is already widely used to reduce the size of entire document an inverted index in the conventional context where each inverted list contains only ids. In that case, an effective approach is to record the gaps between consecutive ids, as opposed to the precise ids. For example, given a set S of integers {2, 3, 6, 8}, the gap-keeping approach will store {2, 1, 3, 2} instead, where the i-th value ($i \geq 2$) is the difference between the i-th and $(i - 1)$-th values in the original S. As the original S can be precisely reconstructed, no information is lost. The only overhead is that decompression incurs extra computation cost, but such cost is negligible compared to the overhead of I/Os. Note that gap-keeping will be much less beneficial if the integers of S are not in a sorted order. This is because the space saving comes from the hope that gaps would be much smaller (than the original values) and hence could be represented with

fewer bits. This would not be true had S not been sorted.

Compressing an SI-index is less straightforward. The difference here is that each element of a list, a.k.a. a point p, is a triplet (idp, xp, yp), including both the id and coordinates of p. As gap-keeping requires a sorted order, it can be applied on only one attribute of the triplet. For example, if we decide to sort the list by ids, gap-keeping on ids may lead to good space saving, but its application on the x- and y-coordinates would not have much effect.

To attack this problem, first leave out the ids and focus on the coordinates. Even though each point has 2 coordinates, we can convert them into only one so that gap-keeping can be applied effectively. The tool needed is a space filling curve (SFC) such as Hilbert- or Z-curve. SFC converts a multidimensional point to a 1D value such that if two points are close in the original space, their 1D values also tend to be similar. As dimensionality has been brought to 1, gap-keeping works nicely after sorting the (converted) 1D value.

Put the ids back into consideration. Now that they have successfully dealt with the two coordinates with a 2D SFC, it would be natural to think about using a 3D SFC to cope with ids too. As far as space reduction is concerned, this 3D approach may not a bad solution. The problem is that it will destroy the locality of the points in their original space. Specifically, the converted values would no longer preserve the spatial proximity of the points, because ids in general have nothing to do with coordinates.

If one thinks about the purposes of having an id, it will be clear that it essentially provides a token for us to retrieve (typically, from a hash table) the details of an object, e.g., the text description and/or other attribute values. Furthermore, in answering a query, the ids also provide the base for merging. Therefore, nothing prevents us from using a pseudo-id internally. Specifically, let us forget about the "real" ids, and instead, assign to each point a pseudo-id that equals its sequence number in the ordering of Z-values.

The benefit from pseudo-ids is that sorting them gives the same ordering as sorting the Z-values of the points. This means that gap-keeping will work at the same time on both the pseudo-ids and Z-values. Since SFC applies to any dimensionality, it is straightforward to extend our compression scheme to any dimensional space.

III. TECHNIQUE TO IMPROVE THE PERFORMANCE OF KNN

A. KNN algorithm

K-nearest neighbor algorithm (KNN) is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition and many others.

KNN is a method for classifying objects based on closest training examples in the feature space. An object is classified by a majority vote of its neighbors. K is always a positive integer. The neighbors are taken from a set of objects for which the correct classification is known.

a) The algorithm on how to compute the K-nearest neighbors is as follows:

b) Determine the parameter K = number of nearest neighbors beforehand. This value is all up to you.

c) Calculate the distance between the query-instance and all the training samples. You can use any distance algorithm.

d) Sort the distances for all the training samples and determine the nearest neighbor based on the K-th minimum distance.

e) Since this is supervised learning, get all the Categories of your training data for the sorted value which fall under K.

f) Use the majority of nearest neighbors as the prediction value.

But this algorithm has some drawback such as follows There is need to determine value of parameter K (number of nearest neighbors)

Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results. Shall we use all attributes or certain attributes only?

Computation cost is quite high because we need to compute distance of each query instance to all training samples.

B. Overview of NLP based optimization technique for knn

Natural Language Processing enables communication between people and computers and automatic translation to enable people to interact easily with others around the world. Natural language processing

is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and natural languages.

The workings of techniques that are previously used (such as IR2 tree and Spatial Inverted List) are shortly explained below with example. Suppose a text document consist of thousand words. These words are saved in database. Important words are extracted from text document and saved in new database. If the query comes then it extract the words from database of important word and gives the output. Here the drawback is it requires lot of time in database creation.

Therefore I proposed a technique based on NLP in which keyword extraction is done automatically from the text document, there is no need to create separate database to save important words from the text document. For automatic extraction of words from text document I will use parts of speech (pos) tagging and chunking. The output of POS Tagging and Chunking is shown in fig. 3.

C. POS Tagging

POS tagging is the process of assigning a part-of-speech to each word in a sentence. Example is given in figure2.
POS tagging is useful in information retrieval, text to speech and word sense disambiguation and also useful as a preprocessing step of parsing

Here I will use Stanford pos tagger because its accuracy and performance is better than other. In this I will extract the parts of speech from text document that is whether words are nouns, verbs, adjectives, etc. Word tags for parts of speech are shown in table1.

D. Chunking

Chunking in NLP is changing a perception by moving a chunk, or a group of bits of information, in the direction of a deductive or inductive conclusion through the use of language.

Chunking has also been fairly commonly used as a preprocessing step. Chunking is also called shallow parsing and it is basically the identification of parts of speech and short phrases (like noun phrases). Part of speech tagging tells you whether words are nouns, verbs, adjectives, etc, but it doesn't give you any clue about the structure of the sentence or phrases in the sentence.

IV. FIGURES AND TABLES

The general idea of working of spatial inverted index is shown in following figure.

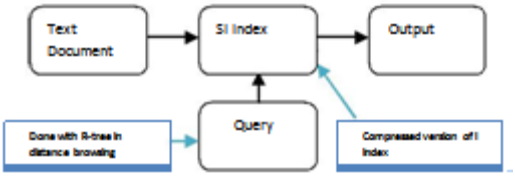

Fig.1: working of spatial inverted index

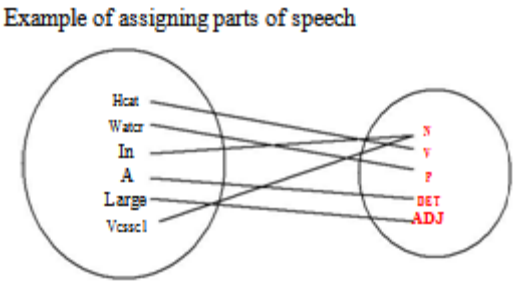

Fig.2: assigning parts-of-speech

Output of pos tagging and chunking

Fig.3: output of pos tagging and chunking

Table for word tags of parts of speech

Word tags		*Word tags*	
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	To
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive Wh-pronoun
PRP	Personal pronoun	WRB	Wh- adverb

Table1: Word tags for parts of speech

V. CONCLUSION

From the study of different techniques on nearest neighbor search and the technique proposed in this paper, I conclude that NLP based optimization technique performs well in terms of accuracy and the time required for database creation.

FUTURE WORK

In this paper two parts that is parts of speech tagging and chunking of NLP based optimization technique are explained and there outputs are shown. In future comparison between technique proposed in this paper an previous technique will be done to check the performance of proposed technique in this paper.

REFERENCES

[1] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[2] bbI. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 656– 665, 2008.

[3] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *Proc. of ACM Management of Data (SIGMOD)*, pages 277–288, 2006.

[4] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatialkeyword (SK) queries in geographic information retrieval (GIR)systems. In *Proc. of Scientific and Statistical Database Management (SSDBM)*, 2007.

[5] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for location-based web search. In *Proc. of Conference on Information and Knowledge Management (CIKM)*, pages 155–162, 2005.

[6] C. Faloutsos and S. Christodoulakis. Signature files: An access method for documents and its analytical performance evaluation. *ACM Transactions on Information Systems (TOIS)*, 2(4):267–288,1984.