A Distributed N-Gram Indexing System to Optimizing Persian Information Retrieval

Mohadese Danesh, Behrouz Minaei, and Omid Kashefi

***Abstract*—As the amount of information and the number of queries has been increasing today, indexing is a good solution to fight with the inherent complexity of text retrieval and accelerating information retrieval in different languages. Also N-Gram Indexing is a solution of the issues such as stemming, misspellings, multilingual and partial matching and has the advantages of language independent and error endurance. Persian is a name of a language which is common in the Middle East. It is spoken in some countries like Iran, Afghanistan and Tajikistan. Therefore, Persian is the language of many documents is published on the net. But, not more researches have been done about the Persian documents retrieval. In this paper, we present a method for Persian documents retrieving using N-gram indexing and distribution technique. The proposed index is a method of more effective answering queries that increases the quality of information retrieval substantially and we gain more optimizing retrieval in Persian documents. But the speed of N-gram indexing is low; to solve this problem we design a distributed N-gram indexing mechanism for large systems of Persian language. Compare with the other methods in this field, we improve the quality of retrieved documents and also the speed of information retrieval.**

***Index Terms*—Information retrieval, indexing, n-gram, distributed, Persian.**

I. Introduction

The process of information retrieval (IR) has become especially important as a result of growing increase in the amount of information stored in the available source [1]. IR aims to help users to find the desired information among the large amount of unstructured ones. There are different models of IR such as Boolean retrieval, linguistic and knowledge-based approaches, inferential networks and statistical model, which includes vector space model (VSM), probabilistic model, latent semantic indexing (LSI) and document clustering [2]. Controlling and locating the best and most relevant documents related to specific information needs are the biggest challenge of the present time [1]. The reasons of using IR systems can be summarized into three cases[3]: (1) faster processing of large volume documents, (2) comparing between documents, (3) having an organized retrieval.

Indexing is one of the best acceptable methods of retrieving information from the contents of documents and is known as the most efficient solution to the rapid and careful retrieval of information [4], [5]. In this paper, considering the growing trend of Persian documents in web pages, we propose a distributed N-gram based indexing to improve the IR quality. The proposed distributed architecture is to bring scalability for multitude and ever growing amount of information, and to cope with the large amount of retrieval request.

The amount of information in Persian language on the internet has increased in different forms. As the style of writing in Persian language is not firmly defined, there are too many web pages in Persian with completely different writing styles for same words and phrases [6]. The most common writing challenges in Persian language that affects the IR are as follows: [7]-[9]

1) Writing in informal or colloquial. As an example words such as «خانه» and «به آن‌ها» are used as their colloquial forms as «خونه» and «بهشون». Colloquial writing also majorly affects the syntax of the sentences by ignoring prepositions and substantial changes in ordering of words.
2) Use of foreign words. In Persian web pages, especially in scientific web pages, a lot of word from other languages have being selected, which some of them are presented in Persian language.
3) Complex Inflection. Persian language includes more than 2800 declensional suffixes. Close fitting, non-close fitting or fitting with pseudo-space of affixes with words is another challenging issue in Persian. For the word "vase" can officially wrong but commonly spell as «گل‌دان», «گلدان», and «گلدان» in Persian. In some cases affixes and especially suffixes may change regarding lemma. As an example the word "bird" spelled as «پرنده» but the plural form is «پرندگان», where the lemma have deformed. This morphophonemic rules also affect the affixes intra-combination. These rules need phonetic attribute of words and so are hard to computationally implement. For instance, single first person pronoun adjective «م» is inflects the word «شرکت» as «شرکتم», «خانه» as «خانه‌ام», «دانا» as «دانایم», and combine with plural suffix «ها» as «هایم» but with «ان» as «انم».
4) Multiple types of writing for a word. In Persian some words have different correct spelling. As an example, words «اتاق» and «اطاق» are spelled correct, pronounce the same, and equally mean "room".
5) Homographs. The same word may have different meanings. This is because of the lack of vowel in Persian writing. Examples are «مَرد» which means «man», «مُرد» which means «died», but both of these words are written as «مرد».
6) Word spzacing. In Persian in addition to white space as inter-words space, an intra-word space called

Manuscript received September 5, 2012; revised November 18, 2012. This work was founded by Computer Research Center of Islamic Science (CRCIS).

The authors are with the School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran (e-mail: mddanesh@comp.iust.ac.ir, b_minaei@iust.ac.ir, kashefi@{iust.ac.ir, ieee.org}).

pseudo-space delimits word's parts. Using white space or do not using any space instead of pseudo-space (Zero Width Non-Joiner in Unicode character encoding), is a great challenge in Persian writing. The lack of a set of comprehensive writing rules, results in writing the same words and phrases in multiple forms by native Persian writers. For example «آبسردکن» is a complete, correct and meaningful sentence means "Cool the water" but «آب‌سردکن» is a word and it means "water cooler".

In this paper, we studied the behavior of lemmatized N-grams in retrieving Persian information, as N-gram indexing method reacts well against spelling errors and changes and overlooks lingual morphemic change. We used document weighting to index and determine the rank of found documents. Since retrieval speed is very important in indexing, we used distributed processing because of the large amount of retrieved information and decrease in the speed of N-gram indexing. Results show that using a combination of unigrams and 3-grams improves the retrieval of Persian document.

We used the Hamshahri corpus as our test bed that consist of the news articles of the Hamshahri magazine from 1996 to 2007. This collection contains over 160,000 articles with different topics. The sizes of articles arevary from short news articles (less than 1KB) to very long articles (140KB) with a mean of 1.8KB. Emphatic signs, punctuations, stop words, separators, and numbers have all been omitted during the preprocessing of the corpus. The preprocessing basically consists of a process to optimize the list of terms that identify the corpus, which is previous to the query process. This optimization can be focused to reduce the number of terms eliminating those with poor information. In the more basic case of preprocessing, a list of stop words is used to reduce the number of terms that identify the collection. The included terms in the stop words list do not provide information about documents and have high frequency in them such as «و»,«که»,«را»,«از»,«به», «در» [10]. Another step in preprocessing phase is lemmatization. The term lemmatization stands for the reduction of an inflected word forms to its stem or root form. Lemmatization also reduces the number of terms that identify the document collection. This issue is very important because reduces the storage space and computational time [4].

In order to avoid linear scanning of the text corpus for each query, which leads to considerably low response time and performance, we index the documents well in advance [11]. For doing this operation, the major steps required are as follows:

1) Collect the documents to be indexed
2) Breaking documents into word-based N-grams derived from documents
3) Produce a list of indexing terms
4) Calculate n-grams of each query and compared it with each of n-grams of documents
5) Weighting and organizing documents based on the similarity of the documents with the queries

Search engines return a fixed number of matches ranked by their similarity with input query. Ranking is a process of matching a query to the documents and allocating scores to the documents according to their degree of similarity [12]. The search should also be cost effective and time effective. It is very important that the result obtained by the search query should be equivalent to what we are actually searching for [13]. We obtained best results for Persian texts retrieval with word-based N-gram model using a new weighting method. Since the string of large information consist of N-grams of documents and queries, gets entire of memory space and for dealing efficiently with high traffic of user queries, we used alphabetical distribution as a stress test to share multitude information into multiple processors and respond to several user queries in a little time.

The remainder of this paper is organized as follow. Section 2 described the most related works, Section 3 studies some attributes of Persian language and look over its challenges in IR. Section 4 is dedicated to the concept of N-grams. Section 5 discusses the IR in a distributed way. The proposed N-gram based IR method is described in Section 6. Evaluation results came into Section 7 and Section 8 concludes the paper.

II. Related Works

Indexing a text document refers to automatically extracting words that are suitable for building an index for the document. Mansour et al. [14] use a number of grammatical rules to extract stem words that become candidate index words and computes weights for these words relative to the container document. The weight is based on how spread is the word in a document and not only on its rate of occurrence. The candidate index words are then sorted in descending order by weight so that information retrievers can select the more important index words.

Another work on indexing is [15] that proposed the new indexing method uses character based N-grams and word-based systems so as to overcome the high false drop to the mismatches between queries and documents, new word and phrase problems that exist in the character-based and word-based systems. Another work available in [16], handle the high dimensionality of text documents, by map each document (instance) into R (the set of real numbers) representing the trigram frequency statistics profiles for a document. Li et al. [17] proposed a new keyword extraction method based on tf/idf with multi-strategies. The approach selected candidate keywords of unigrams, bigrams and trigrams, and show that their proposed method can significantly outperform the baseline method. Ethan et al. [18] discussed the implementation techniques that allowed to use N-gram based retrieval methods on a gigabyte corpus on commodity personal computer hardware. They showed that using appropriately tuned gamma compression, extensible hash tables and significant amounts of pre-calculation on the inverted index allows the indexing of a one gigabyte multilingual corpus with 256 MB of memory.

Fuzzy logic [19], VSM [20] and language modeling methods[21]were used in the studies done about the retrieval of Persian documents. Persian test collection is introduced in [7] where authors used a number of documents in law domain. The size of this collection is small (only 15 judged queries) and the relevant documents are discovered by just one IR system using pooling method. A main drawback of this collection is being domain specific, resulting in a very limited application. Nayeri and Oroumchian [19] proposed the design and testing of a fuzzy retrieval system for Persian

(FuFaIR) with support of fuzzy quantifiers in its query language. Their comparison shows that performance of FuFaIR is considerably better than VSM. Also experiments in [21] suggest the usefulness of language modeling techniques for Persian. Oroumchian et al. [20] used N-gram method in search for Persian documents and they give better results by applying LCA (Local Context Analyses) and N-Grams. Considering trigrams and 4-grams extracted from documents and not using stemming process they showed that compared with VSM, 4-gram method increases retrieval efficiency for searching.

Our work presented in this paper differ from other experiments in four major ways, (1) we used a larger test collection with more queries that needs more calculation to speed up IR, (2) we have tested a novel methods that not employed yet, (3) we utilize word-based N-gram method to get meaningful phrases that are most similar to queries, and (4) we took advantages of distribution to gets our indexing method less than 1 milliseconds.

III. Persian Languages

Persian language is one of the used languages in the Middle East, so there are significant amount of Persian documents available on the web[16, 20]. There are 32 letters in Persian language. Words are written from right to left and numbers from left to right, both horizontally. There are no differences between the capital and small letters. Most letters join the following letters. Depending on the position of each letter in a word, it can come after joining or non-joining letters and this has caused complexities in this language [16].

A. Spelling Normalization

Normalization includes cleaning the context up from unnecessary marks such as !, .,? And etc so that the text would be prepared for the next lemmatizing and stopping processing. Stop words are ones having no role in the word`s meaning; lemmatizing includes writing the root of a word and deleting the prefixes and suffixes.

Entrance of Arabic letters in farsi such as Tanwin (‘‘) and Hamze (ء), two writing forms of some letters such as (ی) and (ک) and some words such as «طهران» and «تهران» for "tehran", different spaces among a word, between words, or even no space, and create new words in farsi that is n't in the dictionary are some problems causing changes in the results of internet searches and should be resolved in the normalization.

In our experiments we detected such errors using a normalization method to convert every occurrence of these to a single form of them. N-Grams method was used to identify such errors and turn each word into a fix and standard form; using this method, we could resolve the problems of the words' structure, too.

B. Persian Lemmatization

One of the main challenges in IR systems is variations in word forms. In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications [22]. Persian has a complex morphology. Referring this challenge, stemming process is used to reduce inflected (or sometimes derived) words to their stem, base or root form. Most Persian words (except some proper nouns and words borrowed from other languages) are derived from a root. A root usually consists of three letters. We can view a word as derived by first applying a pattern to a root to generate a stem and then attaching prefixes and suffixes to the stem to generate the word [23]. Thus, the key terms of a query or document are represented by stems rather than by the original words. For this reason, we uses the lemmatizer proposed in [8], [22].

IV. N-Gram Indexing

The operations of text normalization and indexing are essential components of any IR system. We may regard the document as having been processed into a stream of indexing terms [11]. In information retrieval, each needed –by –user word is compared with the saved documents. When the saved documents are little, text is serially searched and each one of the words in the document is compared by the user query, but if there are a lot of documents, text search is divided into two sections. 1) Indexing, 2) Searching. In indexing, an index of a list of searched word and whole evaluated document texts are prepared, but, in searching, only the index, instead of whole text, is searched [24].

Efficient algorithms and data structures may be used to index a document collection, to enable rapid retrieval of documents containing query words [11]. In N-Grams indexing, the retrieval location is also used for texts including dictation errors or changes and, successfully, does the alphabetic string compare among the words. This kind of indexing does not need any previous information about the concepts or language of the aimed text. N-Grams include N-Characters or word burst.

In N-gram approach, a string of N sequential characters are extracted from a text without considering word length or boundaries and Word based N-gram consisting of N consecutive words [25]. N-grams are generalized words consisting of N consecutive symbols. The N-gram model evaluates the statistical characteristics of texts in a given collection without considering the dictionary, the verbal and syntactic features of a natural language [16].

The N-gram length (N) and the method of extracting N-grams from documents vary from one author and application to another [16]. N-gram indexing method as one of the types of indexing which reacts well against errors and spelling changes in words [26] and also by changing indexing and query limits from words to concepts [12] can produce a rational result for Persian texts. Therefore, we can summarize advantages of N-gram representation as follows: (1) they are very resistant countering the grammatical and print errors of the document. (2) They do not need language preprocessing such word stemming or stop word removal.

There are disadvantages, too. One is the usage of an index through N-Grams method leading to the creation of incorrect results created by the production of N-Grams, which may be semantically meaningless. N-Grams indexing can be as massive as it can not be controlled or managed. In such cases, finding the user's query among the indexes is time consuming, makes problems in the speed of retrieving, and, as the users' requests increase, this time would become longer [27]. N-Grams indexed words include overlapped substrings of

conclusive words which cannot be correctly reflect the conceptual information of the text and this problem happens in the character based indexing. So we used the text based indexing in our proposal [15]. A few efforts have been done to decrease the size of N-gram index [28], [29].

Index compacting is one way to reduce the index size and save more indexes in a fewer space. Total time cost of data transfer from memory to CPU cache and the omission of compactness to the non-compacted data would be less. Also, some method use diffuse extraction [4], [30]. This method needs previous knowledge about the N-Grams frequency and N-Grams extraction with higher frequency. This structure reduces the precision rate because N-Grams might be extracted having no relation with the user request. [31] was proposed which has the advantages of N-Gram indexing and doesn't have a long run time building and loading indexes. The main factor of the increase of disk space and running time is that various N-Grams are generated from each word. In N-Grams etymology method, a single N-Grams is replaced with each word and the other N-Grams derived from it.

Most information retrieval systems use the word-based N-Grams method because of its advantages to the character-based one .some of this method`s advantages are:

Number of the unique words is fewer than the character-based N-Grams generated in the same text (n>3).

Etymological methods can be only used in the based on word systems.

Character-based N-Grams cannot remove the stop words, but it is possible by using the standard statistical techniques such as central subtracts or TF-IPF method.

Consequently, we use the based on word indexing method to use the advantages of indexing method [18].

V. Distributed Information Retrieval

Recent days have seen an explosive growth in the availability of various kinds of data. Vast amounts of unstructured documents are available in many fields, and when we consider the web, documents can be retrieved from all over the world. A large size of documents has to be shifted to retrieve the information profitable to the user [24]. Ranking, on the other hand, is a process of matching an informal query to the documents and allocating scores to documents according to their degree of similarity to the query [3]. To organize and save a massive amount of alphabetic and information, improving the search quality and reducing the time cost, and an effective search method are necessary [13].

In the distributed information retrieval system, a medium is needed to send the user's needs and query to the processors to process, collect, and organize responses. Arranging and providing the best responses are necessary in the response organization stage [32].

VI. Distributed N-Gram Indexing

A. Text Preprocessing

Farsi documents, because of their complexity and problem when being processed, need a preprocessing before indexing. The existence of various written prescriptions, spaces between or in the words, characters' different dictation, morphological changes, and pos tagger are the Farsi language problems which should be resolved in the preprocessing stage. This was necessary due to the variations in the way text can be represented in Persian. The preprocessing section receives the document (s) and the possible query from the user and performs the following tasks:

Convert text files to UTF-8 encoding.

Detect and remove some characters such as punctuation marks, diacritics, non-letters and stop words (Stop words are the words that don't play any roles in determining an special range of information [10] in texts and queries and simplify the searchable text in order to decreases the volume of the documents, changes document weight and increases speed of the search. We make a list of Persian stop words.

Word stemming; Sometimes a user type a special keyword to search the web which that word does not exist in the document exactly but its meanings are. Suppose that a certain noun is singular in some cases and plural in other cases. If the indexing algorithm does not perform word stemming, then it would treat each form of this noun as a different and independent term, but we know this is incorrect. In information retrieval, we can consider words of morphemic differences to be equal because most of the times they have similar concepts. Keywords of a query or document are presented in the form of its root. Because Stemming algorithms are developed to decrease a word in the form of its root. The number of the separate words available in them decreases and counting frequencies is done only on these roots [33]. A little researches are performed about Persian stemming [23], [34], [35]. Affixes are not independent letters which sometimes are used in combination with the other words, while making new words, new meanings to join them. These letters are divided into three categories: (1) the letters which are added to the beginning of the words that we call them prefixes. (2) The letters which are located between two words that we call them infixes and (3) the letters which are added to the end of words that we call them suffixes or rhymes. As the affixes and effectiveness on words and meanings are important in Indo-European languages, it is true in Persian language which is subset of Indo-European languages. Each word contains a stem that contains the main idea In Persian. This means that total number of words can be decreased into a few words that considered as stem, and the rest of the words are derivation this stems. Deriving a word of the principal stem causes that the main concept of that word takes a better shape, or expresses a syntax role in a sentence [22]. With this rule we stemmed both noun and verb to its root completely, Checking nouns and verbs against of their special Affixes and extracting stem words from nouns and verbs separately.

B. N-Gram Extraction

Recognizing the text structure`s limitation which are sentence, clauses, and words is the first step of text processing called tokenization .Tokenization's Algorithms identify the dots and spaces between words for determining the text limitations.

We use N-grams model and move word by word from the beginning of the document toward the end of the sentence in order to produce N-grams. If m was the number of studied words, the length of the existing N-grams in the sentence will

be between 1 and *m*-1. Continuing the process for other sentences, if an extracted N-gram was repeated, its frequency has just increased; otherwise it will be added to dictionary with frequency of 1.

C. Weight Assignment and Index Selection

In this step, we assign weights to the stemmed words regarding their document. Several methods are used for weighting terms, which the most widely used is TF-IDF [17]. The first factor is the frequency of occurrence of that word in its container document and second factor is its frequency in whole documents. This way of weighting is a statistical criterion which is used for the evaluation of the importance of a special word in a collection of documents. Since we follow the importance of a word in the whole collection of documents and the weighting method of TF-IDF is efficient just for a special document, we changed TF-IDF as (1). In our experiment, it seems the weight of an N-gram depends on three factors, which represent the significance of an N-gram in its container document and in the corpus.

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{1}$$

where

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$n_{i,j}$: the number of occurrences of the considered term in document d_j

$$idf_i = \log \frac{|D|}{|\{d_j : t_j \in d_j\}|}$$

$|D|$: total number of documents in the corpus

$|\{d_j : t_j \in d_j\}|$:number of documents where the term t_i appears

First factor, $tf_{i,j}$, and second factor, ifd_i, are the same as TF-IDF, but third factor, $B_{i,j}$, is depend on inverse aggregation of N-gram in the document. With respect to our observation, the rate of aggregation 1-grams is twice 2-gram and triple 3-gram in any document. So we develop a formula for the spread factor such that it increases as the word spreads over the documents, and decreases as the term concentrates in a specific document.

$$w_{i,j} = tf_{i,j} \times ifd_i \times B_{i,j} \tag{2}$$

Based on this result, we consider probability of 1-grams as α, 2-gram is (α/2) and 3-grams are (α/3). In this formula, $W_{i,j}$ is the weight of the term i in the document j and $B_{i,j}$is the desired n-gram coefficient which is calculated as follows ($B_{i,j}$ can be each of α, β or γ values dependent of n or length of term i based on number of words):

$$\alpha(\textit{No. of } 1\textit{Grams}) + \beta(\textit{No. of } 2\textit{Grams}) + \gamma(\textit{No. of } 3\textit{Grams}) = 1$$

$$\alpha\text{: the probability of } 1\text{-grams} \tag{3}$$

$$\beta\text{: the probability of } 2\text{-grams} = 2\alpha$$

$$\gamma\text{:the probability of } 3\text{-grams} = 3\alpha$$

Since the total probability for each existence in math is supposed 1, then the total probability of n-grams in the query is let 1 too, whereas in this paper we study up to 3-grams, so the total probability of n-grams with *n*=1, 2, 3 has been calculated. The values of α, β and γ show the probability of 1-grams, 2-grams and 3-grams respectively in the query and documents. These n-gram coefficients multiply in the number of the same n-gram in the query. If the length of query was equals to 3, then the coefficient $B_{i,j}$ for unigram, 2-gram and 3-gram terms are considered to be 16.7, 33.3 and 50 percent respectively.

If the query length is 2, then γ=0 and if the length is 1 then $B_{i,j}$=0. In order to calculate the weights of the queries with over 3 words, we divide them into unigram to 3-gram groups. Then we calculate $B_{i,j}$ for each group using the formula presented in equation (2) and add them together. For example, for the term «تعداد تلفات جاده‌ای بم» the obtained sub queries are according to Table I.

For each of the lines in Table I we add the weights obtained in each column and consider the maximum weight obtained in each line and take it as the final weight of a special document. If all the N-Grams obtained in one of the lines of the table are similar to another line, we ignore it. Finally, we choose the documents which consists of the most term weights based on the user query and arrange the in a descending way.

D. Distribution

Data mining often needs to spend many resources for saving and needs a high calculation time. Distributed data mining analyzes the data in the distributed field and compromises a concentrated set of data with the distribution analysis of it. Using the data mining as distributed, data set can be processed as parallel and distributed.

Serial search of all data sets, according to the CPU and I/O consumption time cost, is inefficient and, as data become massive, the memory problem will appear. In order to give a better response to the user's query, distribution and reduction of search cost are necessary.

Distributed system increases the subscription, efficiency, time cost, extensibility features, and the quality of responding to the users. Through this, data mining with massive data problem can be resolved by linking separated computers to the network [36].

Therefore, data is partitioned and transmitted to the computing nodes in advance; it is important to efficiently partition and distributes the data to other nodes for parallel computation. In this paper, we focus on the problem of N-gram distribution for indexing. N-grams can be partitioned on to multiple processors with many algorithms. We use alphabetical partitioning of N-grams and transfer them across the processors. Each processor provides capability to create transparent remote processes. Number of processors is depending on volume of partitions.

Alphabetical distribution method was used in our study. This method`s dependence on language and accomplishment simplicity are its features. Propositional distribution performance is based on the number of processors and, according to the number of processors, divides the

alphabetical letters based on the letter order; if *m*, *n* are considered respectively as the number of processors and the alphabetical letters, a number of (n/m) is located in the processor. As Farsi language has 32 letters, we used four processors as local servers to process the N-Grams. Therefore, based on the Fig. 1, 8 letters are located in each processor, respectively, and any number of N-Grams, of which primary letter is in the same 8 letters' limitation, would be located in the considered processor.

In order to assess the impact of this distribution, one master and four slaves were involved in this experiment. Each slave was to copy 0.5 megabyte of data from the master.

We reached the time less than a few seconds after calculating the time needed for searching queries and reached the time less than a few seconds after saving the search results in a buffer. Table III shows the time relating to performing 20 queries in different states. Fig. 2 show this procedure in detail.

Since we extracted the available 1-Gram to 3-Grams and because there are many produced N-Grams, we would have a lack of memory problem if we want to carry out the distribution based on the N-Grams length (*n*), because the number of 2-Grams is extremely more than 1-Gram and the number of 3-Grams is extremely more than the 2-Grams. If we design the 1, 2, and 3-Grams to each processor respectively, we would have the upload problem in the RAM (especially for 2 and 3- Grams) and the search speed would be highly reduced, too.

TABLE I: N-GRAMS EXTRACTED FROM SENTENCE «تعداد تلفات جاده ای بم».

3-gram coefficient (γ)	3-gram	2-gram coefficient (β)	2-gram	1-gram coefficient (α)	1-gram
0.75	تعداد تلفات جاده‌ای	0.50	جاده‌ای بم + تعداد تلفات	0.25	بم
0.75	تلفات جاده‌ای بم	0.5	تلفات جاده ای	0.25	تعداد
0.75	تلفات جاده‌ای بم	0.50	تعداد تلفات + جاده‌ای بم	0.25	تعداد + بم
0.75	تعداد تلفات جاده‌ای	0.75	جاده ای بم	0.25	تعداد
		0.75	تعداد تلفات	0.25	تلفات + تعداد
				0.25	جاده‌ای + بم
				0.25	بم

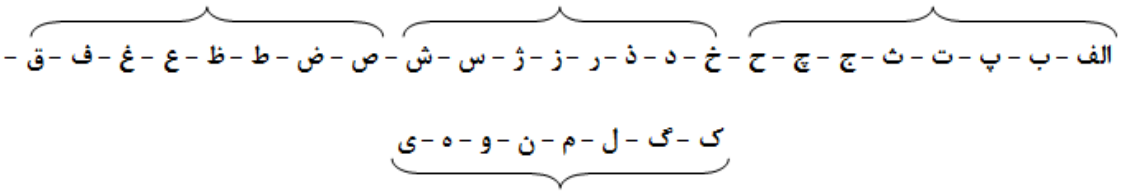

Fig. 1. Division of alphabetical distributed.

For this reason, the propositional distribution method balances the available N-Grams in each processor and there will not be any problem in uploading the information and N-Grams in the memory. In addition, the user query time would be highly reduced. If the user query time in each processor (i) is t_i and the organizing time,comparing the Grams attained from each query, and document ranking are t_r, we will have a time equation of $t_{final}=\max(t_i)+t_r$. Comparing this method with the other distribution methods and distribution disuse, we will realize better results.

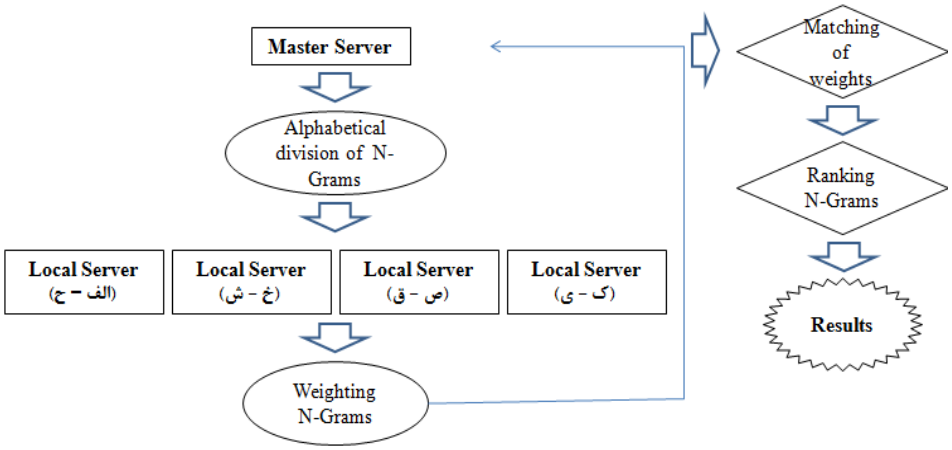

Fig. 2. Steps of distribution indexing.

VII. Evaluation

To evaluate our proposed method, we used a standard collection for Persian texts called Hamshahri. Hamshahri corpus is the largest collection for Persian texts which was prepared and distributed by Tehran University. It has 50 queries relevant to news articles. To store of produced n-grams, N-Grams are load in memory for one time, and for each search, we don't need to load any more. In this section, we turn to the comparison of our proposed method with Local Context analysis model [20] and Term-based method based on lnu.ltu weighting model [20] for retrieving Persian texts. Among of many solutions of Persian retrieval were suggested, these two models, have better results rather than others in precision and recall values. There are so many weighting models to examining Term-Based method with our proposed method [5], [7], [19], [20]. The reason we used lnu.ltu method for comparison is that Oroumchian compare the weighting method of lnu.ltu with other weighting methods and find it more efficient than other weighting methods and obtained better results in retrieving Persian documents compared with other methods [6], [7], [19], [21]. Term-based method based on lnu.ltu weighting model, refers to studying each of the words (unigram) available in documents and queries using the weighting method of lnu.ltu for every word in the documents and queries. Local Context Analysis is based on concepts, which are defined as single nouns, two adjacent nouns or three adjacent nouns instead of keywords or terms. These concepts are calculated in the local passages retrieved and then top m ranking concepts are chosen for query expansion. Local Context Analysis only marginally improved the results over the Lnu.ltu method. This could be due to the fact that the Lnu.ltu weighting method is performing very well on the Farsi language and it is difficult to improve over this good result. So, we run more experiments with different parameters to see whether we can find better configuration for LCA or not. To do this, we used the criteria of precision and recall whose formula is shown in (4) and (5) Since there is just Hamshahri judgment of queries for our evaluation, so we have one choice to match our results by other works in this way. Because of the large number of queries that was used to comparison of our results with judgments, we used the benefits of mechanical evaluation. For this reason, each line of judgment's file that include of precision value of each query, compare with our result. We show the results of our evaluation in diagrams in Fig. 3 and Table II.

$$precision = \frac{(|\, retrieved \,| \cap |\, relevant \,|)}{|\, retrieved \,|} \quad (4)$$

$$recall = \frac{(|\, retrieved \,| \cap |\, relevant \,|)}{|\, relevant \,|} \quad (5)$$

The retrieved refers to the documents retrieved by the proposed system and the term relevant is the average of the documents denoted as relevant for each query in the corpus. To evaluate, we obtain the precision relevant to each query for fixed Recall levels e.g. 11 points in increments of 0.1.

Then we obtain the precision values of different queries for a fixed recall which are shown in Table II. We compared the evaluation results in Table II with other methods. For example, for query 1: «برگزیدگان جشنواره فیلم فجر», its precision value based on the call 1 is 0.7. We obtained the precision of other queries for the recall 0.1 and then calculated the mean precision value for a fixed recall value. List of entire queries are exist in Hamshahri corpus.

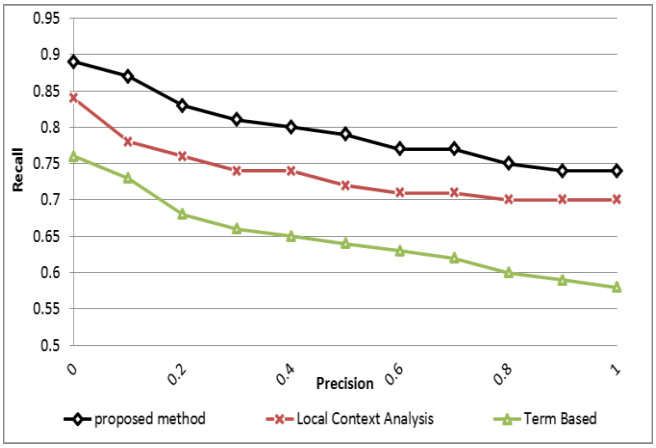

Fig. 3. Recall/precision of the proposed approach in IR

TABLE II: The Precision\Recall Values.

Recall	Precision		
	Proposed method	Term Based	Local Context Analysis
0	0.89	0.76	0.84
0.1	0.87	0.73	0.78
0.2	0.83	0.68	0.76
0.3	0.81	0.66	0.74
0.4	0.80	0.65	0.74
0.5	0.79	0.64	0.72
0.6	0.77	0.63	0.71
0.7	0.77	0.62	0.71
0.8	0.75	0.60	0.70
0.9	0.74	0.59	0.70
1	0.74	0.58	0.70

As seen in Fig. 3 our presented method is superior in performance than the other methods of retrieving Persian documents. The label local concept analysis in Fig. 3 means using a vector space model based on the weighting method of local concept analysis which was explained in details by [20]. They have proved that N-gram with N=4 are viable approaches that outperform their non-stemmed term based counter parts. They have resulted that LCA system and the vector space model with Lnu.ltu weighting scheme and *slope*=0.25 and pivoted unique normalization have better performance than the FuFaIR [19].

The results from distributed processing are shown in Table III and we could increase the run speed of queries to a large extent. We use of client-server method and web-Based to implement our distribution system. In this table the total time of each quarry processing shows in each row with its ID. For our evaluation, we use 20 queries of Hamshahri that were prepare with Tehran University and be almost 4-grams and a few number of 3-grams. These N-Grams are processed simultaneity and execute in a few milliseconds. 4 Computer are used as client and 1 computer as a server. Computer properties to run and evaluate are, RAM: 2.47 GB, CPU: INTEL 2- CORE, 2.40 GHz with O.S: windows 7 ultimate, 32 bit. In our test, local servers get N-Grams from master server, calculate N-Gram weights and send documents that are consist of their N-Grams. Master server send alphabetical N-Grams to local servers with a network way, receive documents Id from them, compare weight of each document and rank documents based on their weights.

TABLE III: TOTAL TIME OF QUARRY PROCESSING.

Query ID	With Distribution	Without Distribution
1	0.12	2.923
2	0.053	1.888
3	0.193	3.681
4	0.16	3.064
5	0.11	2.179
6	0.056	1.093
7	0.067	1.584
8	0.074	2.496
9	0.036	1.763
10	0.04	1.49
11	0.086	2.874
12	0.229	4.351
13	0.076	2.878
14	0.179	2.103
15	0.049	1.744
16	0.037	1.582
17	0.011	1.15
18	0.22	5.524
19	0.08	3.134
20	0.07	3.03

VIII. CONCLUSIONS AND FUTURE STUDIES

In this paper, we extracted one-grams to 3-grams available in documents and queries using an N-gram indexing method and we reached a high performance in extracting Persian documents by presenting a formula for weighting these N-grams and finding documents with the highest weight. We also used distribution method to increase the performance speed of information retrieval and made it possible to query in less than a few milliseconds. In future we plan to extract 4-grams from documents and compare the results with the method proposed in this paper using lnu.ltu weighting method.

REFERENCES

[1] F. CAN, *Turkish Information Retrieval: Past Changes Future*, Berlin, Allemagne: Springer, vol. 4243, 2006.

[2] N. Fuhr, "Models in information retrieval," in *Lectures on information retrieval*, 2001, pp. 21-50.

[3] A. Maristella, *et al.*, *Eds., ed: Springer-Verlag New York*, Inc., 2001, pp. 21-50.

[4] C. Manning, *et al.*, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[5] J. Mayfield and P. McNamee, "Single n-gram stemming," *Presented at the Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, Toronto, Canada, 2003.

[6] A. AleAhmad, *et al.*, "Experiments with English-Persian text retrieval," *Presented at the Proceeding of the 2nd ACM Workshop on Improving Non English Web Searching*, Napa Valley, California, USA, 2008.

[7] M. Shamsfard, *et al.*, "STeP-1: A set of fundamental tools for persian text processing," *Presented at the Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010.

[8] F. Oroumchian and F. Garamaleki, "An evaluation of retrieval performance using farsi text," *Presented at the First Eurasia Conference on Advances in Information and Communication Technology*, Tehran, Iran, 2002.

[9] O. Kashefi, *et al.*, *Towards Automatic Persian Spell Checking*. Tehran: SCICT, 2011.

[10] A. Zamanifar, *et al.*, "A new hybrid farsi text summarization technique based on term co-occurrence and conceptual property of the text," *Presented at the Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Thiland, 2008.

[11] I. A. El-Khair, "Effects of stop words elimination for arabic information retrieval: A comparative study," *International Journal of Computing and Information Sciences*, vol. 4, pp. 119-133, 2006.

[12] S. Renals, *et al.*, "Indexing and retrieval of broadcast news," *Speech Commun.*, vol. 32, pp. 5-20, 2000.

[13] A. Moffat and J. Zobel, "Self-indexing inverted files for fast text retrieval," *ACM Trans. Inf. Syst.*, vol. 14, pp. 349-379, 1996.

[14] P. A. Vijaya, *et al.*, "Leaders-Subleaders: An efficient hierarchical clustering algorithm for large data sets," *Pattern Recognition Letters*, vol. 25, pp. 505-513, 2004.

[15] N. Mansour, *et al.*, "An auto-indexing method for Arabic text," *Inf. Process. Manage.*, vol. 44, pp. 1538-1545, 2008.

[16] L. Du and Y. Sun, "A new indexing method based on word proximity for chinese text retrieval," *Journal of Computer Science and Technology*, vol. 15, pp. 280-286, 2000.

[17] L. Khreisat, "A machine learning approach for Arabic text classification using N-gram frequency statistics," *Journal of Informetrics*, vol. 3, pp. 72-77, 2009.

[18] J. Li, *et al.*, "Keyword extraction based on tf/idf for Chinese news document," *Wuhan University Journal of Natural Sciences*, vol. 12, pp. 917-921, 2007.

[19] D. E. Miller, *et al.*, *Techniques for Gigabyte-Scale N-gram Based Information Retrieval on Personal Computers*, ed, 2007.

[20] A. Nayyeri and F. Oroumchian, "FuFaIR: A fuzzy farsi information retrieval system," *Presented at the Proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications*, Dubai/Sharjah, UAE, 2006.

[21] F. Oroumchian, *et al.*, "N-gram and local context analysis for persian text retrieval," *Presented at the 9th International Symposium on Signal Processing and Its Applications*, 2007.

[22] K. Taghva, *et al.*, "Language model-based retieval for farsi documents," *Presented at the Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, vol. 2, 2004.

[23] O. Kashefi, *et al.*, "Optimizing document similarity detection in persian information retrieval," *Journal of convergence Information Technology (JCIT)*, vol. 5, pp. 101-106, 2010.
[24] M. I. Mobarakeh and B. M. Bidgoli, "Verb detection in persian corpus," *Journal of Digital Content Technology and its Applications (JDCTA)*, vol. 3, pp. 58-65, 2009.
[25] M. Basavaraju and D. R. Prabhakar, "Clustered distributed index for efficient text retrieval using threads," *International Journal of Grid Computing & Applications (IJGCA)*, vol. 1, 2010.
[26] W. Cavnar, "Using an n-gram-based document representation with a vector processing retrieval model," in *TREC*, 1994.
[27] A. AleAhmad, *et al.*, "Cross language experiments at persian@CLEF 2008," in *Evaluating Systems for Multilingual and Multimodal Information Access*. vol. 5706.
[28] C. Peters, *et al.*, *Eds., ed: Springer Berlin/Heidelberg*, 2009, pp. 105-112.
[29] J. P. Callan, *et al.*, "Searching distributed collections with inference networks," *Presented at the Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, United States, 1995.
[30] F. Scholer, *et al.*, "Compression of inverted indexes for fast query evaluation," *Presented at the Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 2002.
[31] W. Hersh, "Managing gigabytes\— Compressing and indexing documents and images (second edition)," *Inf. Retr.*, vol. 4, pp. 79-80, 2001.
[32] E. Sutinen and J. Tarhio, "Filtration with q-samples in approximate string matching," *Presented at the Proceedings of the 7th Annual Symposium on Combinatorial Pattern Matching*, 1996.
[33] P. McNamee and J. Mayfield, "N-gram morphemes for retrieval. Morpho challenge 2007," *Presented at the Available online in the Working Notes of the CLEF 2007 Workshop*, Budapest, Hungary, 2007.
[34] D. Hiemstra, "Distributed information retrieval using keyword auctions," *Ed. Enschede: Centre for Telematics and Information Technology*, University of Twente, 2008.
[35] D. Jimenez, *et al.*, "The influence of semantics in IR using LSI and K-means clustering techniques," *Presented at the Proceedings of the 1st International Symposium on Information and Communication Technologies*, Dublin, Ireland, 2003.
[36] A. Mokhtaripour and S. Jahanpour, "Introduction to a new farsi stemmer," *Presented at the Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, Arlington, Virginia, USA, 2006.
[37] K. Taghva, *et al.*, "A stemming algorithm for the farsi language," *Presented at the Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, vol. I – vol. 01, 2005.
[38] W.-C. Shih, *et al.*, "Performance-based data distribution for data mining applications on grid computing environments," *The Journal of Supercomputing*, vol. 52, pp. 171-198, 2010.

Mohadese Danesh received her M. Sc. in Computer Software Engineering from Iran University of Science and Technology, Tehran, Iran in 2011. She is now a researcher in the Computer Department of *National* Iranian *Oil Company* and a member of Young Researchers Club, Tehran, Iran.

Behrouz Minaei obtained his Ph. D. from Computer Science and Engineering Department of Michigan State University, USA, in the field of Data Mining. Now, he is assistant professor in the Department of Computer Engineering, Iran University of Science & Technology. He is also leader of Data and Text Mining Research Group in Noor Computer Research Center, Qom, Iran, developing large scale NLP and Text Mining projects for Persian/Arabic language.

Omid Kashefi received his M. Sc. in Computer Software Engineering from Iran University of Science and Technology, Tehran, Iran in 2008. He is a Lecturer and Research Associate in the School of Computer Engineering at the Iran University of Science and Technology. His major research interests include distributed systems, virtualization, system software and operating systems, and natural language processing.