P-Trie Tree: A Novel Tree Structure for Storing Polysemantic Data

Xin Zhou(✉)

Department of Economic Engineering, Kyushu University, Fukuoka 812-8581, Japan
thebestzx@163.com

Abstract. Trie tree, is an ordered tree data structure that is used to store a dynamic set or associative array where the keys are usually strings. It makes the search and update of words more efficient and is widely used in the construction of English dictionary for the storage of English vocabulary. Within the application of big data, efficiency determines the availability and usability of a system. In this paper, I introduce p-trie tree-a novel trie tree structure which can be used for polysemantic data which are not limited to English strings. I apply p-trie to the storage of Japanese vocabulary and evaluate the performance through experiments.

Keywords: P-trie · Trie tree · Polysemy · Time stamp · Binary search tree

1 Introduction

1.1 Favorite Words and High Frequency Words

Everyone has favorite words which are often spoken out or written in the letters or papers. For some people, the favorite word is "sorry", because they are gentle and always be modest for other people. And for some other people, one of their favorite words is "amazing", which is used in "Having an amazing day" or "That is amazing". For most people, favorite words are never really given much thought to, they use these words unconsciously, only when they are asked such questions about "what are your favorite words? ", they come to recall some words. Even when they tell their favorite words, they don't realize how much they use them.

Favorite words is one manifestation of high frequency words. The statistic of high frequency words is not only used for exploring people's psychological status, but also used for natural language processing, language study and text mining, etc.In text mining, high frequency words are helpful to obtain the substance of the text.

Not only English, any kind of language has favorite words. For example, in Japanese, we call it "口癖"(the pronunciation is "kuchiguse").

1.2 Kuchiguse

"Kuchiguse" can help to know people's psychological trend and character. For example, when two people are talking, if one people always say "たしかに"("tashikani"),

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 362–371, 2015.
DOI: 10.1007/978-3-319-25207-0_31

it implies that although the people seem to be understanding another people's sayings, but actually he is not interested in the topic, he wants to finish the talking as soon as possible. If one people always say "忙しい"("isogashii"), but he is not really so busy, we can infer that the people wants other people to recognize his existence and sometimes he is a little lonely.

1.3 NLP and Trie Tree

Natural language processing (NLP) is an interdiscipline related with computer science, artificial intelligence, and computational linguistics, its task is to enable computers to derive meaning from natural language input. For Japanese, some important steps of NLP, such as semantics-based words segmentation, can be solved by the Part-of-Speech Analyzer Mecab[1].

In NLP, a vital problem is how to store words. For English, we can use trie tree to store words. A trie tree has a number of advantages over a binary search tree (BST for short). It can also be used to replace a hash table, over which it has several advantages as follows:

1. For looking up data in a trie tree, the time complexity is O(m) (where m is the length of a search string).
2. There are no collisions of different keys in a trie tree.
3. There is no need to provide a hash function.
4. A trie tree can provide an alphabetical ordering of the information of nodes.

Trie tree does have some drawbacks as well. The main drawback is the more require ment of space than a hash table.

1.4 Polysemy

"Polysemy" means one word has multiple meanings. In this paper, we give it more implications which are as follows.

1. One pronunciation has different words. For example, "選考", "先行" , "専攻" and "せんこう" have the same pronunciation of "senkou".
2. One data item has different manifestations. This situation is often used in the field of cryptology. For example, a string "10" may have some concealed information which are comprised of three strings:"10", "IT" and "do". If we don't know the concealed information, we can't get to know the actual representation of the original data.
3. One word has different manifestations of other words without regard for pronunciation. For example, "make" may have one actual meaning selected from three words: "make", "more", "more", "made".

I propose a deformed trie tree named p-trie tree(p-trie for short). "P" means "polysemy". P-trie can store any kind of data, not limited to words. It can efficiently help us to search out high frequency words and the actual representations of original data.

In this paper, we apply p-trie to the storage of Japanese vocabulary.

The existing trie tree is unavailable for Japanese because of the difference of language structures. In English, any word is composed of 26 characters, every character has the same length of one digit. In Japanese, words are composed of “kanji” , “hiragana” or “katagana” , “kanji”s with equal length may have different lengths of hiragana which represents the pronunciation, same words may have different forms, sometimes words are written by “kanji”, sometimes by “hiragana”.

The paper is organized as follows. Section 2 gives the related works. Section 3 explain the structure of p-trie. Section 4 discusses the application of p-trie. Section 5 is the conclusion.

2 Related Works

Mecab[1] is a very popular Part-of-Speech Analyzer in the field of Japanese language processing. It has a number of useful functions. We often use the functions of Lexical analysis, N-Best solutions output, word by word output, pronunciation (“hiragana” or “katakana”) output, etc. Sosuke Amano et al.[4] do frequency statistics of Japanese food names from the foodlog. Because for one food, maybe there are several different names . They can use small numbers of words to describe most of the Japanese foods. Through building various kinds of libraries for frequently appearing words, Yuki Akiyama et al.[5] improve the accuracy of the comparison and identification of Japanese words. Tomoya Noro et al.[6] use a graph-based method to rank Japanese words, this is the first step toward building a Japanese defining vocabulary. These are some latest publications in the field of Japanese Language Processing, within these papers, the storage mode of Japanese words is not discussed as a key point.

All the time, there are a lot of research papers concerning on tree structure. Based on the idea of partitioning T into a set of linked small tries, each of which can be maintained efficiently, Jesper Jansson et al.[2] propose a new technique for maintaining a dynamic trie T of size at most 2 w nodes under the unit-cost RAM model with a fixed word size w. Kevin Leckey et al.[3] consider a more realistic model where words are generated by a Markov source. By a novel use of the contraction method combined with moment transfer techniques they prove a central limit theorem for the complexity of radix sort and for the external path length in a trie tree.

3 P-Trie Tree

3.1 Definition

P-trie tree is a novel trie-tree. It is an ordered tree data structure that is used to store a dynamic set or associative array where the keys can be of any type. The position in the tree defines the key with which it is associated. Any node not only links to its child nodes, but also links to a data structure called the information of this node(“Info” for short). All the descendants of a nod have a common prefix of the string associated with that node, and the root is associated with the empty string.

Compared with trie-tree, p-trie has some differences.

1. The keys in the nodes can be of any type. According to different conditions, the number of types of keys and the values for the corresponding keys may be different.
2. Any node also links to a data structure called “Info”. According to different conditions, “Info” may be vector, binary search tree or others.
3. “Info” can contain uncorrelated data which are of different types or different meanings. The data in “Info” are not limited to have relation with the keys in the nodes.

3.2 Comparison with Map (Red-Black Tree)

Map is one kind of red-black tree, it can also store a dynamic set or associative array. Compared with Map, p-trie has some advantages.

1. Map has the limitation of capacity. P-trie has a much larger capacity.
2. Map has a mechanism of automatically sorting of the data. But for other different language’s vocabulary, it is very different for us to compare which is smaller between two words. For example, in a Map, “務める”(“つとめる”) is smaller than “収集”(“しゅうしゅう”). But according to the order of Japanese phonetic alphabet, “収集” should be in front of “務める”. In p-trie, we store words by the order of phonetic alphabet, it is easy to compare two words.

3.3 Application in Japanese

In p-trie, every node terminates some words comprised of the Japanese characters in the depth-first search route from the root’s child node to this node. Each node contains a subtree which stores the different appeared words of the same pronunciation In the subtree, nodes are arranged by the sequence of time stamps which denotes the first time of appearance of this written form, the node with time stamp 0 is the first child, node with time stamp 1 is the second, and so on. Each node’s subtree stores some information (“info”) of words terminated at this node, “Info” contains four main parts: the string of the word in the written form of hiragana, kanji or katakana; the time stamp; the frequency and the vector of numbers of sentences in which the word appears.

Based on the p-trie tree structure, we can do Japanese paragraph analysis. The fundamry ental of paragraph analysis is sentence analysis, and the fundamental of sentence analysis is word analysis.

Case 1. Given 13 words, assume that words appear chronologically and each word is in separated different sentences. We can insert them to construct a p-trie tree.

The words are: せんこう　先行　選考　専攻　学問
生協　せんこう　成果　生活　セイカ
先行　成果　せんこう

For a Japanese word, there are three kinds of written forms: hiragana, katakana and kanji. For one hiragana, maybe there are several different kanjis with different meanings. Although kanjis are different, they also have the same hiragana. In some Japanese advertisements, hiragana can also be written in the form of katakana, the meaning is same, just the written forms are changed.

To classify different words which have the same hiragana(then same pronunciation), we give every word a time stamp which is ordered by the time of appearance starting from 0. In p-trie, we use hiragana characters to express words, katakana is deemed to be a different form of the same hiragana. For one hiragana, maybe there is only one word, then the time stamp is 0, maybe there are more than one written form, then time stamp is the time order number of the first appearance of this form.

The information of all the words in case 1 are shown in Table 1.

Table 1. Information of words

Japanese word	Time stamp	frequency	Sentence No.
せんこう	0	1	1
先行	1	1	2
選考	2	1	3
専攻	3	1	4
学問	0	1	5
生協	0	1	6
せんこう	0	2	7
成果	0	1	8
生活	0	1	9
セイカ	1	1	10
先行	1	2	11
成果	0	2	12
せんこう	0	3	13

Based on the definition of p-trie, we can construct a p-trie as shown in Fig. 1.

As shown in the p-trie, root node is an empty node which is denoted as “R”, from the left to right, the Japanese characters of child nodes are arranged by the order in Japanese phonetic alphabet which is shown in Fig. 3. Every pane represents “Info”. In a pane, the first line is the word, the second, third and fourth line separately represents the time stamp, frequency and the vector of sentence numbers.

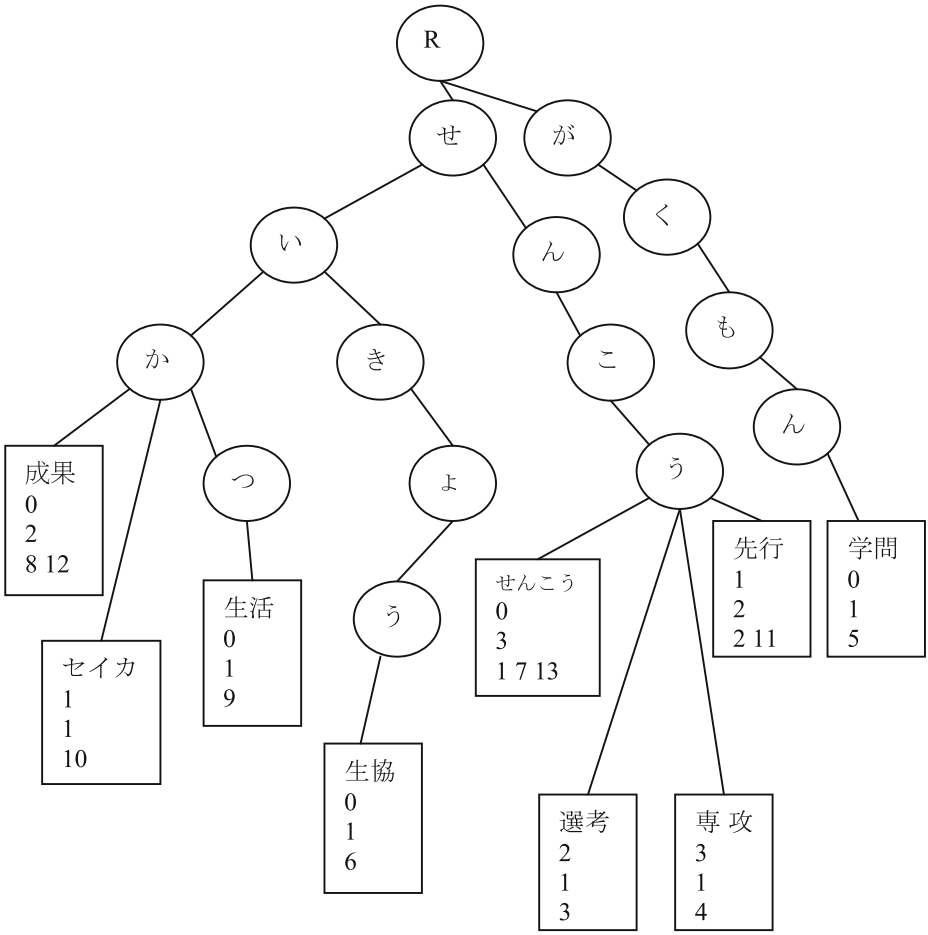

Fig. 1. The p-trie tree of case 1

The data structure of "Info" is decided by the actual applications. If we want to search by the content of characters, we shall construct a p-trie in Fig.1, if we want to search by the frequency, the "Info" shall be constructed as a binary search tree in which the time complexity of search a data item by the key (in this context is frequency) is O(logN) where N is the number of nodes in the BST. Fig.2 shows the BST of "Info" for the hiragana "せんこう".

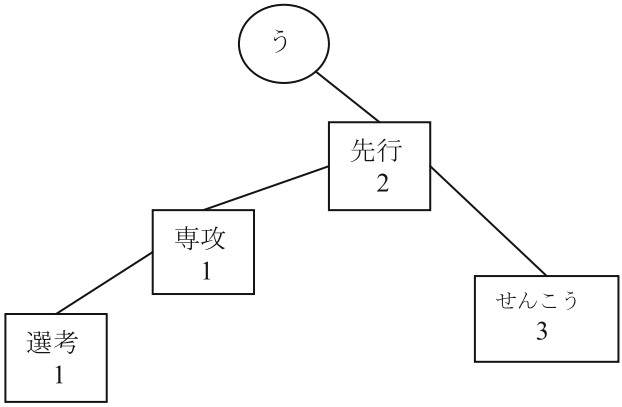

Fig. 2. The binary search tree structure of “Info”

段 行	あ段			い段			う段			え段			お段		
	平	片	罗马	平	片	罗马	平	片	罗马	平	片	罗马	平	片	罗马
あ行	あ	ア	a	い	イ	i	う	ウ	u	え	エ	e	お	オ	o
か行	か	カ	ka	き	キ	ki	く	ク	ku	け	ケ	ke	こ	コ	ko
さ行	さ	サ	sa	し	シ	si	す	ス	su	せ	セ	se	そ	ソ	so
た行	た	タ	ta	ち	チ	ti	つ	ツ	tu	て	テ	te	と	ト	to
な行	な	ナ	na	に	ニ	ni	ぬ	ヌ	nu	ね	ネ	ne	の	ノ	no
は行	は	ハ	ha	ひ	ヒ	hi	ふ	フ	hu	へ	ヘ	he	ほ	ホ	ho
ま行	ま	マ	ma	み	ミ	mi	む	ム	mu	め	メ	me	も	モ	mo
や行	や	ヤ	ya	い	イ	i	ゆ	ユ	yu	え	エ	e	よ	ヨ	yo
ら行	ら	ラ	ra	り	リ	ri	る	ル	ru	れ	レ	re	ろ	ロ	ro
わ行	わ	ワ	wa	い	イ	i	う	ウ	u	え	エ	e	を	ヲ	wo
か行	が	ガ	ga	ぎ	ギ	gi	ぐ	グ	gu	げ	ゲ	ge	ご	ゴ	go
さ行	ざ	ザ	za	じ	ジ	zi	ず	ズ	zu	ぜ	ゼ	ze	ぞ	ゾ	zo
た行	だ	ダ	da	ぢ	ヂ	di	づ	ヅ	du	で	デ	de	ど	ド	do
は行	ば	バ	ba	び	ビ	bi	ぶ	ブ	bu	べ	ベ	be	ぼ	ボ	bo
は行	ぱ	パ	pa	び	ピ	pi	ぶ	プ	pu	ぺ	ペ	pe	ぽ	ポ	po
													ん	ン	n

Fig. 3. Japanese phonetic alphabet

In Fig.3, we can see the order is numbered from left to right and from top to bottom. For hiragana, there are 71 distinct characters, the first one is "あ" and the last one is "ん". There are four additional short syllables "っ","ゃ","ゅ","ょ" and one long syllable "ー" which are not shown in phonetic alphabet. Katakana is deemed to be a special written form of the same hiragana. Therefore, there are totally 76 characters.

In fig. 1, we can see for the same root "せ","い" is in the left of "ん";"セイカ" is another form of "せいか".

3.4 Other Applications

As discussed in section "Introduction", p-trie can be used in the field of cryptology.

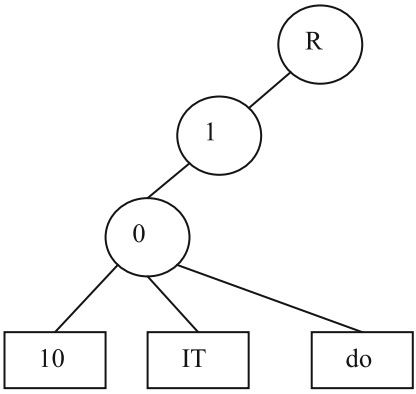

Fig. 4. A part of p-trie which stores actual representations

Fig.4 gives a part of a p-trie in which the written string "10" has three types of actual representations. For example, given a two-tuples ("10", 1) in which "1" is the index of the actual value in the "Info", if we have this p-trie, we can easily get the string "IT", otherwise, we cannot know the actual value of the original string.

4 Experiments

We build a file called "words.txt" in which each line is consisted of several Japanese words with the same pronunciation and the last one is the hiragana. In "words.txt", there are 110 words. We randomly generate a text file named "in.txt" which contains 10000000 lines, each line is a word, all of the words are from "words.txt". We construct a p-trie by the words in "in.txt" and do operations of insert and search in the p-trie.

We use C++ to do the experiments, the development environment is Microsoft Visual C++ 2010. All experiments were run on a computer with an Intel COREi5 2.7 GHz CPU, 4GB RAM, running Windows 7 OS.

4.1 Results

In the p-trie, we insert strings and search strings. If the string is existing, we output the time stamp and frequency. Sometimes, the string is not in the p-trie, but it's a prefix of a word which is in the p-trie, “かく” is an example for this case.

In table 2, we list the operations and the corresponding running time. In Init(), we build the map of Japanese phonetic alphabet and get the hiragana for every word which is in “words.txt”. In ReadIn(), we read in the data in “in.txt” whose size is 68176 KB.

Table 2. Experimental results

Operation	Running time (unit: seconds)
Init()	0.005000
ReadIn();	546.100000
insert "ありがとう"	0.000000
search "ありがとう"	0.000000
search "かく"	0.000000
search "かくｓ"	0.000000
search "かくしんかく"	0.000000
search "大学"	0.000000
search "因る"	0.000000
search "合う"	0.000000
search "昇る"	0.000000

4.2 Performance Analysis

In the p-trie, every node has at most 76 child nodes which are arranged by the order of Japanese phonetic alphabet. Each child node has a subtree in which the nodes are arranged by the order of time stamp.

Define the length of characters of the hiragana of a word as the length of the word.

The time complexity of insert, search and update is O(m) where m is the length of the word. The space complexity is $O(76^n)$ where n is the maximum length of word in the p-trie. P-trie tree uses huge space to achieve a faster running time.

5 Conclusion

In this paper, we discussed a novel data structure named p-trie tree. The most remarkable advantage of p-trie tree is it can effectively store polysemantic data. The time performance of most operations on p-trie tree is very good. The requirement of space is huge but nowadays hard drive is cheap and easy to get.

P-trie tree is not only available for the storage of vocabulary of languages, but also available for other applications with the same or similar data structure, for example, the application of information security. For different applications, the structure of

p-trie could have a little change, such as the number of child nodes, the information of child nodes, etc. New applications of p-trie tree could be future challenges.

References

1. http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
2. Jansson, J., Sadakane, K., Sung, W.-K.: Linked Dynamic Tries with Applications to LZ-Compression in Sublinear Time and Space. Algorithmica **71**(4), 969–988 (2015)
3. Leckey, K., Neininger, R., Szpankowski, W.: A Limit Theorem for Radix Sort and Tries with Markovian Input (2015). CoRR abs/1505.07321
4. Amano, S., Ogawa, M., Aizawa, K.: Frequency statistics of words used in Japanese food records of FoodLog. In: UbiComp Adjunct 2014, pp.547–552 (2014)
5. Akiyama, Y., Shibasaki, R.: A method for identifying japanese shop and company names by spatiotemporal cleaning of eccentrically located frequently appearing words. In: Adv. Artificial Intellegence (ADVAI 2012), pp. 562604:1–562604:18 (2012)
6. Noro, T., Tokuda, T.: Ranking words for building a japanese defining vocabulary. In: IJCNLP 2008, pp. 679–684 (2008)