Implementation of String Matching and Breadth First Search for Recommending Friends on Facebook

Ahmad 13512033
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512033@std.stei.itb.ac.id

***Abstract*— Nowadays, Facebook has been one of the most influential social networking services to daily human activities. Facebook connects people with their friends by exchanging message, sharing photos, publishing notes, and updating status. Facebook has a feature called *People You May Know*. This feature gives user the chance to expand their connection with new people by giving recommendation to add another user. However, Facebook recommendation is not based on the thought or interests of the user. It only depends on the expansion of mutual friends or joining the same groups in Facebook.**

This paper will cover on analysis of string matching to determine the interest of user by checking their status, notes, and other sharable features which can be examined by string matching. The analysis will include detailed example and the expansion of breadth first search to find the correct users who will be listed on *People You May Know* box.

***Index Terms*— Facebook, string matching, breadth first search.**

I. Introduction

Facebook is the most popular free social networking website that allows registered users to create profiles, upload photos and videos, update status, send messages and keep in touch with friends or family. String matching will take the advantage of updating status feature to determine user's interest. User's interest will be used to find another people on Facebook who has the same interest with him. Soon afterward, the result will be a list of recommended friends for the user.

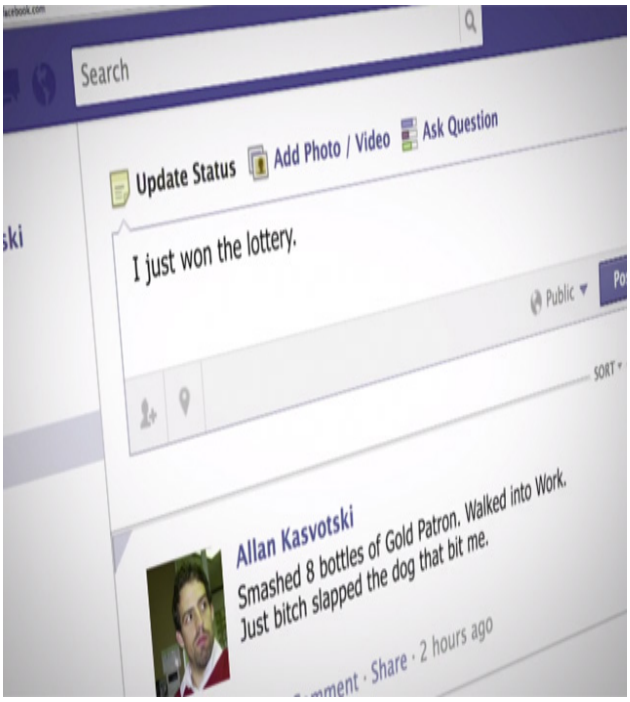
II. FACEBOOK Definition & Terms

Facebook, as explained before, is a social networking website. Members can locate other Facebook members and becoming friend with them by sending them an invitation, or they can invite people to join Facebook. It also offers instant messaging and photo sharing. Facebook also allows it users to update their status which allows users to discuss their thoughts, whereabouts, or important information with their friends. When a status is updated, it posts on the user's personal wall, as well as in the news feed of their friends.

2.1. Timeline

Timeline is a virtual space in which all content of Facebook users is organized. In a Timeline the photos, videos, and posts of any given user are categorized according to the period of time in which they were uploaded or created.

Every user can update their status to show other members what is the idea that hold on their mind. The definition of a status update means something different to everyone, but the idea is the same. A short, text-based entry telling your friends something you deem worthwhile. Some people use it to share inspirational quotes, some use it to describe their daily actions, and some use it to share links and pictures.



I just won the lottery.

Smashed 8 bottles of Gold Patron. Walked into Work. Just bitch slapped the dog that bit me.

Figure 2.1. Update status box on timeline

2.2. People You May Know Features

People You May Know looks at, among other things, your current friend list and their friends. If users are already friends on Facebook with some people from their job, they may find some more of coworkers among the suggestion.

This feature has the intention of helping newest people on Facebook find their friends quickly. The Home page will only appear with what People You May Know considers to be its very best suggestions.

People You May Know see all

Jim M
Add as Friend

Erin Elizabeth K
Add as Friend

Josh S
Add as Friend

Figure 2.2. People You May Know box

III. Basic Theories

3.1. *String Matching Algorithm*

A string matching algorithm is an algorithm to check whether a certain pattern is exists within a text. There are many kinds of string matching algorithm.

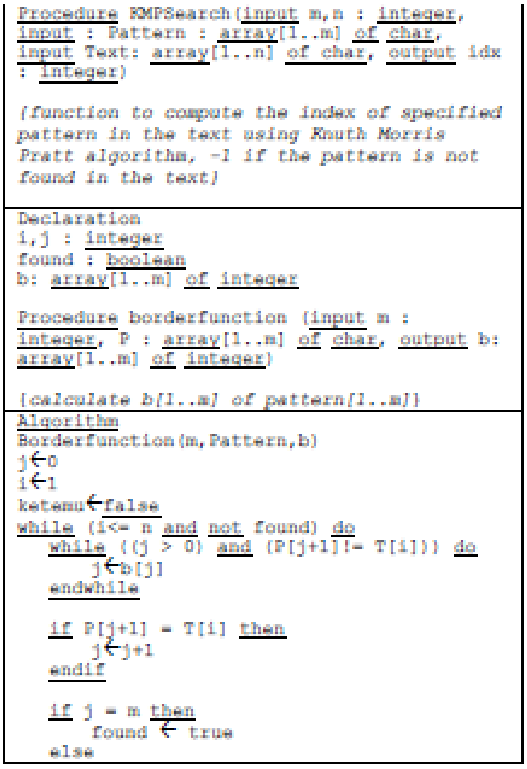
3.1.1. Knuth-Morris-Pratt Algorithm

The KMP algorithm allows shifting pattern by more than one character depend on the border function. The border function of a pattern is defined as the maximum length of the same prefix and suffix of the pattern.

Here is the algorithm of KMP:

1. Calculate border functions of the pattern
2. Align the pattern with the beginning of the text
3. For each character in the pattern, check whether the chararcter in the pattern is the same with the character in the text. If all character matches, finish.
4. If false. Shift the pattern by length minus border function(length). If the number of remaining character of text more than length of pattern, back to step 3.

Here is the full algorithm of KMP:



```
Procedure KMPSearch(input m,n : integer,
input : Pattern : array[1..m] of char,
input Text: array[1..n] of char, output idx
: integer)

{function to compute the index of specified
pattern in the text using Knuth Morris
Pratt algorithm, -1 if the pattern is not
found in the text}

Declaration
i,j : integer
found : boolean
b: array[1..m] of integer

Procedure borderfunction (input m :
integer, P : array[1..m] of char, output b:
array[1..m] of integer)

{calculate b[1..m] of pattern[1..m]}

Algorithm
Borderfunction(m,Pattern,b)
j←0
i←1
ketemu←false
while (i<= n and not found) do
   while ((j > 0) and (P[j+1]!= T[i])) do
        j←b[j]
   endwhile

   if P[j+1] = T[i] then
        j←j+1
   endif

   if j = m then
        found ← true
   else
```

Figure 3.1.1. Knuth-Morris-Pratt Algorithm

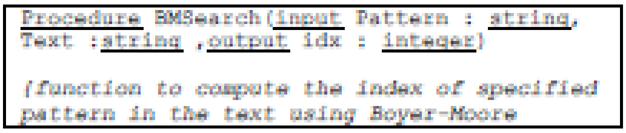
3.1.2. Boyer-Moore Algorithm

BM algorithm is pattern matching algorithm which utilizes the looking glass technique and character jump technique. Looking glass technique means the algorithm will find pattern in text by moving backwards through the pattern starting at its end.

The character jump technique means when a mismatch occurs at a character in text with a character in pattern then there are three possible cases to handle. Here are the cases:

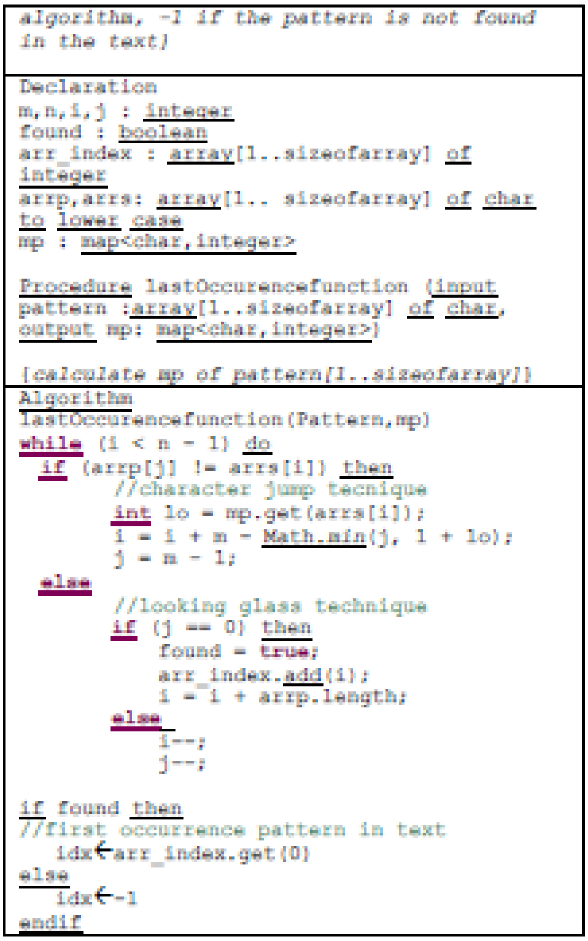
1. There is a pattern contains x somewhere, then try to shift the pattern right to align the last occurrence of x in pattern with the character in text.
2. There is a pattern contains x somewhere but a shift right to the last occurrence is not possible, hence shift pattern right by 1 character to text[1+1].
3. Occurs when cases one and two do not apply. Shift the pattern to align pattern[1] with text[i+1].

Here is the full algorithm of BM:



```
Procedure BMSearch(input Pattern : string,
Text :string ,output idx : integer)

{function to compute the index of specified
pattern in the text using Boyer-Moore
```



```
algorithm, -1 if the pattern is not found
in the text}

Declaration
m,n,i,j : integer
found : boolean
arr_index : array[1..sizeofarray] of
integer
arrp,arrs: array[1.. sizeofarray] of char
to lower case
mp : map<char,integer>

Procedure lastOccurencefunction (input
pattern :array[1..sizeofarray] of char,
output mp: map<char,integer>)

{calculate mp of pattern[1..sizeofarray]}

Algorithm
lastOccurencefunction(Pattern,mp)
while (i < n - 1) do
  if (arrp[j] != arrs[i]) then
         //character jump tecnique
         int lo = mp.get(arrs[i]);
         i = i + m - Math.min(j, 1 + lo);
         j = m - 1;
  else
         //looking glass technique
         if (j == 0) then
             found = true;
             arr_index.add(i);
             i = i + arrp.length;
         else
             i--;
             j--;

if found then
//first occurrence pattern in text
   idx←arr_index.get(0)
else
   idx←-1
endif
```

Figure 3.1.2. Boyer-Moore Algorithm

3.2. Breadth First Search

Breadth-First Search is one of searching strategy on a graph who has two main processes, visit and check each nodes of the graph and visit again the neighbor nodes from the current node. BFS started from root node, and then the validation is made for each neighbor nodes that is not visited yet, and so on until every node has already been visited or the objective of search is reached. This algorithm uses queue to store temporary nodes that will be visited or already visited.

These are the common scheme of Breadth-First-Search:

Let G is a graph and v is a root of G.

```
procedure BFS($G$,$v$) is
    create a queue Q
    create a vector set V
    enqueue v onto Q
    add v to V
    while Q is not empty loop
       t ← Q.dequeue()
       if t is what we are looking for then
          return $t$
       end if
       for all edges e in G.adjacentEdges(t) loop
          $u$ ← G.adjacentVertex($t$,$e$)
          if $u$ is not in $V$ then
              add $u$ to $V$
              enqueue $u$ onto $Q$
          end if
       end loop
    end loop
    return none
end BFS
```

IV. ANALYSIS & IMPLEMENTATION

The idea to make a new People You May Know list method is by finding the members who have the same interests. We have to make a library full of category which can identify member interests. For example, if in user's status there is a pattern *Valentino Rossi* then we can conclude that the user has put some interests to MotoGP.

Moreover, we sort interests of the user and start checking for each user using Breadth First Search. The expanding of each node in BFS will be started from mutual friends of user. The system will check one by one if there is a friend from mutual friends who share the same interest. If it is found, then system will recommend this match to the user.

The general scheme of the calculation is defined as below.

1. For each of user status, do string matching.
2. String matching will check from each pattern of interest's categorization library.
3. List and sort all of matching occurrences by each category.
4. Find using BFS started from user's mutual friend, the member who has the same interest.
5. If found, push it to the list and show the list to user.

4.1. String Matching Implementation

First, we will have a set of problem instance for the implementation. Suppose that user A's status:

User A status:
"Last night barcelona matches was fantastic"

And also, the database of interest's category
Library:
Category football: barcelona , matches
Category music: rock, pop

Category science: math

By this case, using Boyer-Moore string matching pattern we will put in the result list what are the interest of user A. The pattern will be the word of each category. If a word matches the text, then the text will be counted as the category which the word included and the word will be pushed to result list.

First try.
Text: Last night barcelona matches was fantastic
Pattern: barcelona

```
Last night barcelona matches was fantastic
barcelona   (1)
         barcelona (2)
          barcelona (3)
```

Because it matches, then the result list will push “football” and ignore another keyword from football category.

Second try.
Text: Last night barcelona matches was fantastic
Pattern: rock

```
Last night barcelona matches was fantastic
rock (1)
    rock (2)
        rock (3)
            rock(4)
                 rock(5)
                     rock(6)
                          rock(7)
                              rock(8)
                                  rock(9)
                                       rock(10)
```

Because it does not match, then the result will not push “music” yet. Try another keyword from music category.

Third try.
Text: Last night barcelona matches was fantastic
Pattern: pop

```
Last night barcelona matches was fantastic
pop(1)
  pop(2)
      pop(3)
          pop(4)
             pop(5)
                pop(6)
                   pop(7)
                     pop(8)
                         pop(9)
                            pop(10)
                               pop(11)
                                  pop(12)
                                     pop(13)
```

Because the keyword doesn’t match the text, then music category will not be pushed into the result list.

Fourth try.
Text: Last night barcelona matches was fantastic
Pattern: math

```
Last night barcelona matches was fantastic
math (1)
    math (2)
        math (3)
            math (4)
                 math (5)
                     math(6)
                          math (7)
                              math(8)
                                  math (9)
                                       math(10)
```

The keyword math doesn’t match the text, hence science category will not be pushed into the result list.

4.2. Breadth First Search Implementation

From the case before, we should now a result list which only has one interest category that is football category. The next step before showing the user who are the members that share the same interest, we should expand the tree of connection started from user A current friends.

The idea is every user has their interest list. This list will be used to check whether a member and other member have the same interest or not. If at least one of interest match each other list, then those two users will be recommended to become a friend in Facebook.

The expanding of friendship from a member, who is not yet becoming a friend with another member, is by using BFS expanding. The BFS will start the root from member current friend. The current friend will make a tree which the nodes are current friend’s friend who has the same interest as the member. The expanding will be use BFS paradigm.

Let’s say we have a situation like this.

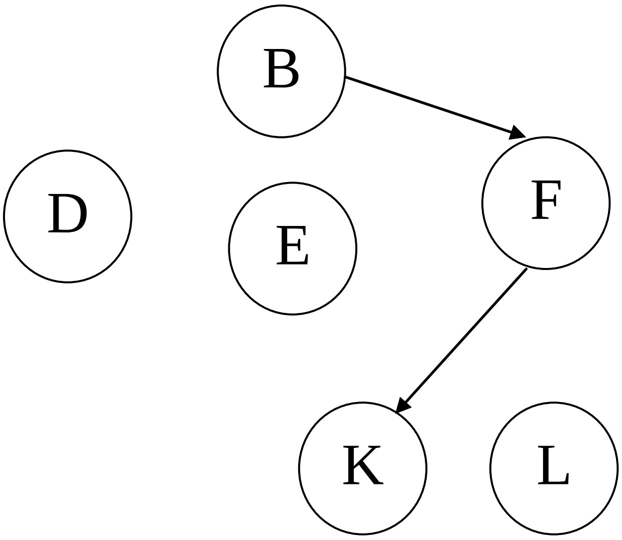
User A’s friends: B, C.
User B friends: D, E, and F.
User C friends: G, H.
User H friends: I, J.
User F friends: K, L.
User I friends: M.

User D interest: Music, Science
User E interest: Education
User F interest: Art, Football, and Basketball
User G interest: Food

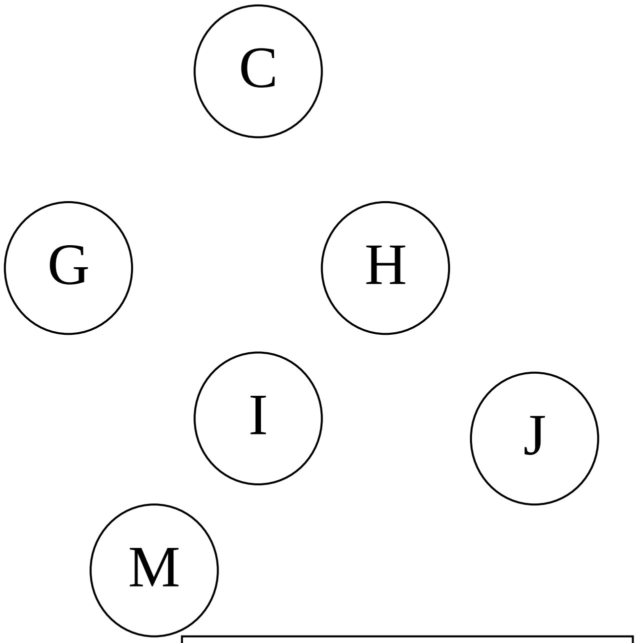
User H interest: Science, Cryptography
User I interest: Drama, Drawing
User J interest: Eating, Football
User K interest: Drinking Coffee, Football
User L interest: Gambling

User A’s interest based on his status is Football

After all information has been collected, we can make the tree of People You May Know. The tree is shown like this.



Tree formed from User B. Tree I.



Tree formed from User C. Tree II.

BFS result the formation of two trees, tree from user B and tree from user C. Let we see from user B, there is a tree that is made from the expansion of the BFS. The Tree has three nodes; they are node B, node F, and node K. The tree made from User C just only result a single node tree, which the node is the only root.

The formation of the Tree by BFS expansion only include the nodes that pass the criteria of selection, in this case is football category. Firstly, Tree I expand it edge just only to node F, because node D and node E do not surpass the selection function. Hence, there is no edge between node B to node D and E. Secondly, Tree I expand it edge again to node K from node F. It happens because node K have football category in its interest, so is not node L. Because there is no other information about friend of node K, then we stop the expanding.

The formation of Tree II is not built many edges, but none. It comes about because node G and node H which are the friends of node C do not qualify the criteria, where the interest of node G is Food and node H are Science and Cryptography. Because it can’t expand anymore edges from its nodes, the formation of the tree is stopped.

Hence, from this process we can give two recommendations of friends to user A; they are user F and user K. Yet, there is still user J who loves football too but is not recommended as a friend to user A.

V. Conclusion

From the examples, and analysis in the previous sections, we can summarize some points:

1. Recommending a new friend on Facebook could be using the method of having the same interests which could be checked by their status using string matching.
2. To make a greater number of results, we should increase the quantity of categorization library and also the width of current friends.

VII. Acknowledgment

Ahmad as the author wants to express his grateful thanks during making this paper to Mr. Rinaldi Munir and Masayu Leylia Khodra, ST., as the lecturers of our classes, and also my mom and dad who always encourage me.

References

[1] Munir, Rinaldi,*Diktat Kuliah Strategi Algoritma.* 2009. Bandung: Program Studi Teknik Informatika ITB.
[2] http://www.facebook.com
[3] http://www.techcrunch.com/2013/01/15/facebook-anounces-its-third-pillar-graph-search/
last accessed: 18 Mei 2014 21.00

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Mei 2014

ttd

Ahmad, 13512033