ALGORITHM FOR FINDING SOURCES OF MESSAGES AND DETERMINE THE MOST EFFECTIVE WAY OF SPREADING MESSAGES

A. Lomako

National Technical University of Ukraine “Kyiv Polytechnic Institute”,
37, Peremohy Ave., Kyiv 03056 Ukraine
lomakoartem@gmail.com

Abstract. The object of study is the social network and the relationships between their users. The aim of this paper is to develop a method of reproduction topology dissemination of messages and finding minimum flow of messages. Methods used in this thesis: parsing, analysis of data obtained as a result of parsing and general scientific analysis. The tree allows following in-step how information spreads in social networks and find sources of dissemination messages. To construct was used programming language C #.

Keywords: C#, API, social network, massage, node.

Introduction

Social network is an immense association of people, information in which quickly spread among all members. The first disseminators of news are users of Twitter, Facebook and only after few hours mass media. Volume of online available information increases regularly and continuously, which makes the problem of finding relevant information is very up-to-date and demanding. Speed of finding relevant information determines largely the professionalism Internet users. There are thousands of large and small Web-sites designed for searching and online. Search tools allow the user to formulate rules to requirements necessary information. In recent years, social networks have become popular which is not so in all areas of our lives. In social networks, users hereby publish information about themselves, their views and interests. Many people don’t realize that they posted information can be found and used by anyone, not necessarily with good goals. Information on social network users can find their relatives, employers, or even criminals. Determining the source of the message leads to the fact that we find people who can spread the wrong information on the network, and nose, or revolutionaries. This can help in resolving issues like personal and national security. After all, we can lead to the proliferation of information sources. The process of investigation is the dissemination of information in social networks which is a complex and time consuming task. The main problem is not the availability of information and the definition of reliable information but rather the selection of the desired user. Considering the fact that present social networks can be used to organize riots, perhaps even attempted to national security, there are many social groups that push people to suicide, and many distributed false information, and it is coordinated through social media networks, the trace their relations is an important task that can help minimize or avoid these problems. The purpose and objectives of the research is to develop a method of reproduction topology dissemination of messages. The object of investigation is the social network and the relationships between their users. The study examined the mechanism of propagation of messages in a social network.

The practical significance of the results is that the automated test units passing messages, eliminating human error.

Scientific novelty of the results is that to construct a tree spreading messages using API in C #, built a database of results and graphically displayed as a tree.

Method

Scheme of approach to distribution system messages in a social network.

Distribution system messages in a social network
- Development of distribution Posts algorithm;
- Parsing;
- Building a message spreading tree.
Approach to the implementation of distribution system messages can be divided into three interconnected parts. First it is necessary to develop a tool designation area code, which is the route we will investigate and tree distributed under News. To do this, it was decided to use the programming language C# using .NET 4.5. The application analyzes the selected post / recent posts from selected walls and building a tree last post repost / post with the output to a text file. The application does not implement authentication and build the tree for positions from closed to the public walls.

Program Vk Repost Tree Builder Description

The program is based on Vkontakte API (Vk API). Provides a convenient API (functions documented in detail, sharing data is formatted JSON) and fast (limit to three requests per second, the response from the server comes with a slight delay) access to all necessary information to build the tree. Currently in response format API has some bugs that were taken into account when writing libraries.

The application consists of an executable and libraries:

- VkRepostTreeBuilder.Console.exe to work with the application of the console;
- VkAPI.dll, which contains classes for working with Vk API for building and tree-based data;
- Newtonsoft.Json.dll, which allows parsing JSON objects. Basic functionality is built into the library. .NET 4.5, but setting features parsing libraries need to be connected separately.

To work with Vk API library is already in the language. .NET (e.g. Vk.NET, .NET Vkontakte API, Vkontakte API for .NET), but

- They are coded to work with all or nearly all of the features API, which is unnecessary because the program uses only a few methods.
- None of the libraries does not provide opportunities for asynchronous requests to the server, which can significantly speed up the processing of a larger number of simultaneous positions. With input async / await and asynchronous HttpClient, self realization methods required becomes quite easy

VkAPI.dll description and structure

Structure class library is as follows:
1) The class to work with Vk API:
- VkErrorException - an exception that is thrown in case of returning an error while processing Vk API (most likely error-access). Contains code and error description in the form in which it turned API.

- VkClient - class for the necessary additional requests API. Provides methods to obtain a specific post on the ID of the owner (owner_id) and post identifier (post_id), the latest posts from the page and get the latest repost selected post. Each method is asynchronous and contains a request to the appropriate function API. The class is designed to work with version Vk API 5.21.

- VkList - generic class that provides. NET wrapper over the list that returns Vk API instead of an array. Most methods return standard version 5.21 JSON array, but some used in the application to incorporate it right Count, which contains the number of elements in the array and the array itself is placed in the box Items.

- VkPost - class corresponding to the essence of "record" (post), which returns the API. Is not full, and only required additional fields of its prototype. Fields HasReposts and IsOriginal - properties that are based on the values of other fields of the class.

- VkReposts - special class to get the number of the corresponding repost post. Each "record" includes field Reposts - object in turn contains a single field Count - number repost this "record". The method section should be written in paragraph form with as little repetition as possible. This section will often be broken down into subsections such as participants, materials and procedure. The subsections you use will depend on what is useful to help describe and explain your experiment.

2) Classes for building and wood repost:

- SimpleTreeNode - primitive recursive tree. Each node contains fields Content - content nodes and Children - list of child nodes. Overridden ToString method recursively goes through all nodes, starting from the current to the output tree in a row.

- VkRepostsTreeBuilder - is the main application class is responsible for building the tree repost a certain post. One constructor takes as argument an instance of the class VkAPI, which is used for the necessary inquiries to the API. Default constructor uses the default VkAPI.
Example for the latest repost:

```csharp
var ownerId = 1;
using (var writer = new StreamWriter("tree.txt"))
    foreach (var tree in _TreeBuilder.BuildTreesAsync(ownerId).Result)
        writer.WriteLine(tree);
```

Results

Graphical implementation of a received message distribution tree.

1) The figure shows the node 6071 and 5990 edges

![Graphical implementation of a received message distribution tree](image1)

2) Distributors of information that are closest to the source of news (as a result, they turned 71)

![Graphical implementation of a received message distribution tree](image2)

Was developed executable file that runs:
- with the addition of the console;
- contains classes for working with Vk API;
- to build a tree-based data obtained;
- It was created library which allows parsing JSON objects;
- It created a database that contains the results of the application;
- Using the generator Gephi, built tree spreading messages that fully reflects all the nodes and links.

References
