Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет комп'ютерних систем та автоматики
(повне найменування інституту, назва факультету (відділення))
Кафедра автоматизаціїї та інтелектуальних інформаційних технологій
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до магістерської кваліфікаційної роботи
на тему «Інформаційна система інтелектуального аналізу природно-мовного контенту соціальних мереж.
Частина 1. Підсистема визначення індикаторних характеристик профілів учасників соціальних мереж"»

Виконав: студент 2 курсу,
групи ЗАКІТ-18м
спеціальності
151 – «Автоматизація та комп'ютерно-інтегровані технології»
(шифр і назва напряму підготовки, спеціальності)
Освітньо-професійна програма:
Інформаційні системи і Інтернет речей

А. Концевой
(surname and initials)
Науковий керівник Бісікало О.В.
(surname and initials)
Рецензент

ВінницяВНТУ – 2019 року

Vinnytsia National Technical University
(Full name of university)
Faculty of Computer Systems and Automation
(Full name of institution, name of Faculty (Department))
Department of Automation and intellectual information technologies
(Full name of the department (subject, framing Commission))

Thesis

to Master's qualifying paper
on the topic «Information system for intellectual analysis of natural-language content of social networks.
Part 1. Subsystem of definition for indicator characteristics of profiles of social networks participants »

Performed by: a student of 2nd course of the 3ACIT-18m group
Specialty:
151 - «Automation and Computer Integrated Technologies»
(Code name and direction of training, specialty)
Educational-professional program:
Information Systems and Internet of Things
A. Kontsevoi
(surname and initials)
Supervisor O. Bisikalo
(surname and initials)
Reviewer
(surname and initials)

Vinnytsia VNTU – 2019

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем та автоматики
Кафедра комп'ютерних систем управління
Освітньо-кваліфікаційний рівень магістр
Спеціальність 151 – «Автоматизація та комп'ютерно-інтегровані технології»
(шифр і назва)
Освітньо-професійна програма: «Інформаційні системи і Інтернет речей»

ЗАТВЕРДЖУЮ

Завідувач кафедри АІІТ
Р.Н. Квєтний
_____

"_____" 2019 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Концевому Антону Олександровичу
(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Інформаційна система інтелектуального аналізу природно-мовного контенту соціальних мереж. Частина 1. Підсистема визначення індикаторних характеристик профілів учасників соціальних мереж»
керівник магістерської кваліфікаційної роботи,
д. т. н., професор Бісікало Олег Володимирович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від "_____"
2019 року № __.
2. Строк подання студентом магістерської кваліфікаційної роботи _____2019 року
3. Вхідні дані до магістерської кваліфікаційної роботи: JSON файли, що містять наступну інформацію: текст, дата і час реакцій користувачів на події, а також повну інформація про самих користувачів , що включає ім'я, записи користувача, локація, інформація про себе тощо).
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):вступ, дослідження методів, підходів та моделей аналізу соціальних мереж,дослідження сучасного стану та проблем аналізу соціальних мереж, практична реалізація та аналіз отриманих результатів, аналіз та створення безнес моделі розповсюдження інформаційної системи, висновки.
5. Перелік графічного матеріалу: звіт про виконання робочого процесу, код програми, блок схема роботи програми, скриншоти роботи програми.
6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	д. т. н., проф. Бісікало О. В.		

Дата видачі завдання “___”_____2019 року

7. Календарний план

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Дослідження методів, підходів та моделей аналізу соціальних мереж		
2	Дослідження сучасного стану та проблем аналізу соціальних мереж		
3	Практична реалізація та аналіз отриманих результатів		
4	Аналіз та створення безнес моделі розповсюдження інформаційної системи		
5	Оформлення пояснювальної записки, графічного матеріалу і презентації		
6	Захист МКР		

Студент _____ А. О. Концевой

(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____ О. В. Бісікало

(підпис) (прізвищетаініціали)

Vinnytsia National Technical University

(Full name of university)

Faculty of Computer Systems and Automation

(Full name of institution, name of Faculty (Department))

Department of Automation and intellectual information technologies

(Full name of the department (subject, framing Commission))

Education and qualification level - Master
Specialty:
151 – «Automation and computer integrated technologies»
(Code name and direction of training, specialty)
Educational-professional program:
Information systems and Internet of things

APPROVED BY

Head of department of the AIIT
R. Kvetny
“_____” 2019

Assignment

TO STUDENT TO MASTER’S QUALIFICATION PAPER

Anton Kontsevoi
(full name)

1. The topic of the master’s qualification paper «Information system of intellectual analysis of natural-language content of social networks.
Part 1. Subsystem of definition of indicator characteristics of profiles of participants of social networks»
Scientific Supervisor
ProfessorOleh Bisikalo, Doctor of Engineering Sciences
approved by the order of the higher educational institution from “_____” 2019 № __.
2. The term of submission by the student of master's qualification work 2019
3. Initial data for master's qualification work: JSON files containing the following information: text, date and time of user reactions to events, as well as complete information about the users themselves, including name, user records, location, information about themselves, etc.).
4. Contents of the explanatory note (list of questions to be developed): introduction, introduction, research of methods, approaches and models of analysis of social networks, research of the present state and problems of analysis of social

networks, practical realization and analysis of the obtained results, analysis and creation of a support model of information system distribution, conclusions.
5. List of graphic materialworkflow report, program code, block diagram of the program, screenshots of the program.
6. Advisors on the chapters of master's qualifying paper

Chapters	Surname, Lastname	Signature, date	
		task issued	task accepted
1,2,3,4	Dr. of Sci., Prof. O. Bisikalo		

7. Date of issue of the assignment "___"_____2019

Schedule

№	The name of the stages Works	The duration of the stages of work	Note
1	Research on methods, approaches and models of social network analysis		
2	Research on the current state and problems of social network analysis		
3	Practical implementation and analysis of the results		
4	Analysis and creation of a business model for information system propagation		
5	Design of explanatory note, graphic material and presentation		
6	Master's degree defense		

Student _____ _A. Kontsevoi
(signature) (firstname, surname)

Supervisor of the master's work _____ _O. Bisikalo.
(signature) (firstname, surname)

АНОТАЦІЯ

Вданіймагістерськійкваліфікаційнійроботібулопроаналізованопоточний стан, підходитамоделідоаналізумоціальнихмереж, наведеноосновніпроблемитазадачіаналізумереж, розробленопрограмнезабезпеченнядляавтоматизованоговизначенняіндикаторниххарактеристикпрофілівучасниківсоціальнихмережіавтоматизованоговизначенняреакціїкористувачівмережінановининаплатформі Android.

Длярозробкипрограмногозабезпеченнядлявирішеннязадачімагістерської роботиобраномовупрограмування Kotlin черезїїтазручністьзастосуванняпристроренні Android додатків, атакожбібліотеку Apache OpenNLPдлясинтаскичногоаналізуданихпроновинитакористувачівмережі.

Наведено та проаналізовано результати роботи інофрмаційної системи, а також розглянуто перспективи з впровадження, розповсюдженя та маркетингу інформаційної системи.

ABSTRACT

In this master's qualification work the current state, approaches and models for the analysis of social networks were analyzed, the main problems and tasks of the network analysis were presented, the software was developed for automated determination of indicator characteristics of participants profiles in social network and automated determination of the reaction of network users to news on the Android platform.

To develop the software for solving the master's thesis, the Kotlin programming language was chosen because of its ease of use when creating Android applications, as well as Apache OpenNLP library for parsing data about news and network users.

The results of the information system operation are presented and analyzed, as well as the perspectives on information system implementation, distribution and marketing are considered.

CONTENTS

INTRODUCTION

Relevance. The relevance of this work is connected with the growth of the Internet and the continuous increase in the amount of data in it, as well as the high popularity of social networks. In this regard, it became necessary to automatically identify and process large amounts of information about participants in social networks. Because existing methods, algorithms and programs do not achieve the desired result, new methods need to be developed to solve these kinds of problems. Examples of such tasks are the task of determining the indicator characteristics of a social network participant profile. Effectively addressing these challenges can help you reach specific groups of people, their interests and hobbies, which will be useful in areas such as SEO and marketing to improve the quality of targeting, trending in the current market, and more.

Social networks are a phenomenon today. The benefits of using social networks are that you can connect with your friends quickly and easily, usually in the form of objects such as posts, pictures, videos and texts. Another feature is networking: friends, colleagues, and family.

Analysis of social networks (related to network theory) has become the main method of research in modern sociology, anthropology, geography, social psychology, informatics and research organizations, and a common topic for research and discussion. Studies in several academic fields have shown that social networks operate at many levels, ranging from families to entire nations, and play an important role in how problems are solved, organizations operate, and succeed in achieving individuals' own goals.

Aggregating information from publicly available profiles is very useful for specific purposes, such as building a marketing strategy and identifying groups of individuals associated with banned organizations.

This raises the problem of collecting, analyzing, and processing large amounts of data about social network users.

Purpose of the work. The purpose of the work is to increase the efficiency of

the system of automated determination of indicator characteristics of profiles of participants of social networks and the process of determining the reaction of users to real-time events.

Object of the study. The object of study is the process of automated determination of indicator characteristics of profiles of participants in social networks, as well as the process of determining the response of users to certain events in real time.

The subject of the study are methods and tools for analyzing the profiles of social network participants, as well as instrumental methods for determining the reaction of social network participants to real-time events on mobile devices.

Research objectives. To achieve this goal, the following tasks must be accomplished:

1. Analysis and review of methods of syntactic and linguistic analysis and their application in solving this problem.

2. Development of a methodology for determining the indicator characteristics of profiles of social network participants by analyzing their profiles and their reactions to events on the network to represent them in the form of input data.

3. Creating a program.

4. Program testing and verification of the data received.

Scientific novelty.

1. For the first time, a new approach to determining the indicator characteristics of profiles of social network participants by analyzing their profiles and their reactions to network events using syntactic methods is proposed and it allows to increase the number of determined characteristics in comparison with existing applications that only show characteristics provided by the social network.

2. The proposed approach allows, unlike existing ones, to analyze user profiles for information about groups of people on the network, as well as their reaction to an event or news on the network on a mobile device by using specialized libraries and combining natural language processing methods in order to increase the accuracy of the results compared to already existing applications.

The practical significance of the results obtained. The research performed in this paper solves the problem of analyzing and determining the indicator characteristics of social network participants' profiles (sample profiles, or specific groups of people) and their reactions to network events.

Testing of results of work. The results of the work were considered at the XLVIII Scientific and Technical Conference of Vinnytsia National Technical University in 2019 [1].

1 THEORY OF SOCIAL NETWORKS

Social network - a social structure formed by individuals or organizations. It reflects the diverse relationships between them through a variety of social relationships, starting with casual dating and ending with close family ties. The term was first proposed in 1954 by J. A. Barnes (in Class and Committees in a Norwegian Island Parish, "Human Relations").

Analysis of social networks (related to network theory) has become the main method of research in modern sociology, anthropology, geography, social psychology, informatics and research organizations, and a common topic for research and discussion. Studies in several academic fields have shown that social networks operate at many levels, ranging from families to entire nations, and play an important role in how problems are solved, organizations operate, and succeed in achieving individuals' own goals.

Social networks are also called online programs that help friends, business partners, or others communicate and connect with one another through a toolbox. These programs, known as "Online Social Networking", are becoming more and more popular [2].

1.1 Theory of social networks

Social network theory examines social relationships in terms of nodes and connections. Nodes are separate actors in networks, and connections correspond to relationships between actors. There can be many types of connections between nodes. In its simplest form, the social network is a reflection of all the links that are relevant to the study between the nodes. Networks can be used to establish the social capital of individual actors. These concepts are often reflected in a social network diagram, with nodes corresponding to nodes and links to nodes.

The form of social network helps to determine the degree of its usefulness to its

participants. Smaller, connected networks may be less useful to their members than networks with many weak connections with individuals outside the core network. Open networks with many weak links and social relationships are more likely to offer new ideas and opportunities to their participants than closed networks with many redundant connections. In other words, a group of familiar friends who only communicate with each other already have shared knowledge and capabilities. A group of people with connections to other social communities are likely to gain access to a wider range of information. In order to succeed, individuals are better off having connections to multiple networks than multiple connections within the same network. Similarly, individuals can influence or act as a broker in the middle of their social networks by connecting two networks that do not have direct links (called filling in social holes) [3].

A social Internet network is an interactive multi-user website with a number of mandatory qualities: - the content (content) of a site is created exclusively or mainly by its users; - the site is an automated environment in which users are able to create connections with other users (social connections) or social objects (thematic groups); - users have the opportunity to receive static and dynamic information about objects existing in a given social environment, about social connections between them; - users have access to communication functions with other users and social objects. Among the functions of a social network, the following should be highlighted:

– communication. As part of the communication function, people make contacts, exchange news, information (photos, videos, audio materials, links to sites, comments, messages), cooperate to achieve shared goals (rallying and maintaining social ties);

– informational. The flow of information is bi-directional, as communication participants act alternately in the role of communicator, and in the role of the recipient;.

– socializing (self-development, reflection in the system of “friends” and “groups”);

– self-actualizing (self-presentation);

– identification (when creating an individual profile, the user fills it with information about himself - name, date of birth, family status, school, university, interests, etc., which allows you to search for profiles according to specified criteria);

– the function of identity formation. According to Festinger's theory, a person is inclined to compare himself with those people with whom he has a greater number of similarities. In addition, according to the theory of cognitive dissonance, similar people positively evaluate each other. This is the main mechanism that allows a person to clearly formulate their positions in relation to other people and groups;

– entertaining. Social networks allow you to exchange not only text messages, but also multimedia files, in addition, it should be noted the importance of widgets - entertainment mini-programs created by third-party manufacturers to expand the capabilities of the user (games, media applications, etc.).

For real social networks, the following main effects and properties can be distinguished:

– the presence of their own opinions of users;

– change of opinion under the influence of other members of the social network;

– the different significance of the opinions (influence, trust) of some users for others; varying degrees of exposure of members of the social network to influence;

– the existence of indirect influence in the chain of social contacts;

– the existence of “opinion leaders”;

– the existence of a threshold of sensitivity to changes in the opinions of others;

– localization of groups ("by interests", with close opinions);

– taking into account the factors of “social correlation”;

– the existence of (usually less significant) external factors of influence (advertising, marketing campaigns) and, accordingly, external agents (media, producers of goods, etc.);

– the presence of avalanche-like effects; the impact of the structural properties of social networks on the dynamics of opinions;

– the possibility of forming coalitions;
– game user interaction;
– information management in social networks [4].

1.2 Social network classification

Some social networks are focused on finding friends and communicating with them, others are business networks, others are created for people to communicate and / or are united by interests.

There are various conditional classifications of social networks in the Internet environment. The following approach was proposed in:

– social networks of a general subject (MySpace, Facebook, Vkontakte, Odnoklassniki, etc.) in terms of non-formal education provide access to information that satisfies their own interests;

– specialized social networks, for example: Last.Fm (music social network), Geni (family social network that unites people based on family ties), Autokadabra.ru (social network of car enthusiasts). This also includes highly specialized professional networks uniting specialists and scientists, for example, Ukrainian Scientists Worldwide, Ukrainian Science of Internet Internet, e-Learning PRO, Scipeople.ru, and others. Researchers J. Leyv and E. Wenger in the work “Situational training: legitimate peripheral participation ” in order to effectively share knowledge, the necessity of creating professional communities was substantiated, the content of activities and their types were disclosed. Studying the joint activities of people in the learning process, they considered the acquisition of knowledge as a social process in which people with different levels of competencies participate (on which their authority in the group depends);

– business social networks - networks that promote professional development and building a personal career.

Among them are two categories of social networks:

– networks in which the main types of nodes are connected with participants and their profiles or pages of participants (Facebook, Vkontakte, LinkedIn, etc.);

– networks in which nodes, which are the most important, are digital objects - articles, programs, videos, bookmarks (Wikipedia, YouTube, Flickr, Delicious, etc.).

A classification of social networks is given in terms of communicativeness and dominance:

– social networks with high dominance and high communicativeness;

– social networks with high connectivity and low dominance;

– social networks with low communicativeness and low dominance;

– social networks with low communicativeness and high dominance.

Analysis of other sources allows you to classify social networks into categories such as:

– type (personal communication, business communication, video, audio, photo, geolocation, shopping, blogging, news, question-answer, bookmarks, virtual worlds, thematic entertainment / dating);

– openness of information (open, closed, mixed);

– geographical coverage (world, country, territorial unit, without region - international);

– the level of development (web 1.0 – web 3.0) [4].

1.3 Data search inside of the social networks

There are passive (direct search) and active (communication) ways of obtaining information. Direct search - Search for, for example, keywords using search services from the network itself or external search engines. Data about a person is anything that the researched person can tell about himself or people who know him or her say about him:

– human constituent data;

– full name, date and place of birth, photo. And, considering that thus protects the data, leaves in one forum a name, in another forum a date of birth, in the third forum icq and thus everywhere is registered under one nickname;

– competences, education, work experience, achievements (with details and details that a workbook is not needed). And again in one place one set, in another slightly changed, in the third with some other change. Information that collects, analyzes, and identifies the most interesting items should be collected;

– relationships: family, friends, workers. Features of personality preferences, hobbies, views, beliefs, statements on forums, blogs and microblogging with the desired problems: in fact, you can make a complete picture of the object under study;

– company data - both about the company and employees;

– the contact and founding details of the company itself, who are employees, who are the managers, who are the executors, how to contact whom, including offended employees and dissatisfied customers;

– the internal situation of the office, the relationships within the team, the size of the office, its filling with property and employees, the activity of employees and the activity of telephone conversations, the presence of clients and work with them, corporate style, venues for corporate events and style of their conduct - this is indirect evidence of the size and profitability of the company;

– data on the product (services) manufacturer, consumer, stakeholders;

– product features, properties, quality, etc.;

– product analogues and substitutes, their features;

– data on the event, the situation, the problems - what, where, when it happened, who the participants and the essence of the event, and, from different points of view and attitude to the event;

– "underwater" currents and implicit forces involved in the event can be identified both by the most popular events and by analyzing the media activity of different groups; - the consequences are obvious and not obvious can be revealed by

direct communication with the participants and witnesses and on the basis of data of interest in the event of different forces;

– from photos: people post what they find interesting, important, worthy. Thus, users are implicitly boasting about property, involvement in the event, communication with an important person. the details of the photos you can find out: location of the object; basic working projects - in the background there may be a closet with a number of to-do cases, at the roots of which are the project names; personal data on the photo of the car (the number should be painted), the address of the stay in the photo on the background of the houses, photos near his house (dachas, hacienda) with the obligatory hit of the address plate. - for metadata gps location coordinates, date-time;

– from personal messages (on "walls");

– date of birth, if at least someone greeted the user "live";

– places of rest of the object when exchanging thoughts about them with friends;

– style of behavior and communication, general trend of statements on certain issues;

– facts of biography (as well as participation in events) become a subject of discussion;

– relationships with other people or organizations are revealed through their mentions in communication;

– communication: Social media users have a wealth of information that they have not published that can be asked directly or prompted by a discussion [5].

Direct search is a search for a personal page using the built-in search engine by coincidence of several characteristics (name + date of birth; or name + place of residence). it is organized in different networks, but it is everywhere because this search is the basis for the development of this network.

This kind of search is organized differently on different networks, but it is everywhere, because this search is the basis for the development of this network.

Search Function - There are various search operators on social networks

(Facebook is "|", (vertical slash or logical "OR"), which helps to display results that contain at least one part of the search query.

In addition, there are many specialized services for searching on social networks (Poiski - shows the result of blog search and photo hosting, links and microblogs, videos, documents, domains, phone numbers, e-mail addresses and other information, Yoname - search by profiles and messages from users of certain social networks, Wink - indexes only "social" sites, where information is provided by thematic tags L. Search is carried out on already filtered and classified information, Bing-social - which correctly searches through the Internet No .

Through Groups is a search for an object in user-created groups (work, training, recreation, places of interest, interest groups and hobbies). Provoking the object itself - you can push the object to react to your actions (utterance) and thereby identify it [6].

1.4 Models of social networks

1.4.1 Graph models

This class of models can be divided into three types. Stochastic block models defined by an X matrix of size $k \cdot k$, where k - is the number of actor blocks. The element $x_{ij} \epsilon [0;1]$ shows the density of bonds between the actors belonging to block i and the actors belonging to block j. Moreover, the graph does not contain additional edges and vertices corresponding to the bonds of actors within one block.

Probabilistic graph models defined by a matrix X of size $k \cdot k$, where k is the number of actors. The element $x_{ij} \epsilon [0;1]$ shows the probability of interaction between actor i and actor j over a certain period of time.

Conventional graph models defined by a connection matrix X of size $k \cdot k$ with the properties indicated above.

For the analysis of graph models of social networks, in addition to the above

coefficients, a density coefficient is used - the ratio of the number of edges in the analyzed graph to the number of edges in the full graph with the same number of vertices (a full graph is a graph in which all vertices are interconnected).

Graph models of social networks are used to model the economic and communication relationships of individuals, to analyze the processes of disseminating information, to find various informal associations and related subgroups into which a common network of social interactions can be divided. For example, V. Baskens developed a theoretical model of relationships based on trust between sellers and buyers in the market of information products, taking into account the effects of management and training.

In this model, two types of interactions are distinguished:

– repeated interaction between the same actors

– social networks that act as information channels and communications between the seller and the buyer, giving information about the seller's relationship with other buyers.

For computer analysis, social network graphs are represented by an adjacency matrix, called a social matrix. Figure 1.2 shows the socio-matrix for the graph.

$X =$ \	n_1	n_2	n_3	n_4	n_5	n_6
n_1	0	1	1	1	0	0
n_2	1	0	1	0	0	0
n_3	1	1	0	1	0	0
n_4	1	0	1	0	0	0
n_5	0	0	0	0	0	1
n_6	0	0	0	0	1	0

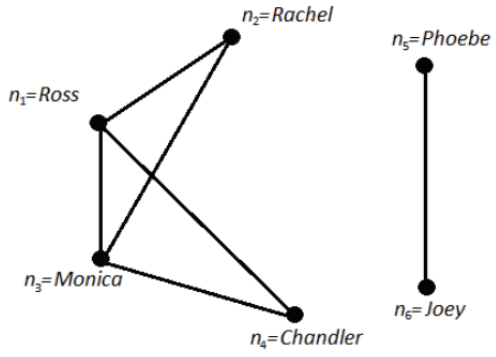


Figure 1.2 - Graph of the social network

The following characteristics of graphs are used in the analysis of social networks:

– the degree of the vertex of a non-directional graph equal to the number of edges incident to it. For a directed graph, the degree of the vertex is calculated separately for the input and output arcs. The degree of the vertex characterizes the activity of the actor;

– the density of a graph is a characteristic of the connectedness of a graph that is calculated as the ratio of the actual number of edges to their maximum number;

– centrality - an activity characteristic of an actor, calculated as the ratio of the degree of the corresponding vertex of the graph to the number of vertices with which it is possible to associate;

– actor prestige is determined by the ratio of the number of input arcs of the corresponding vertex of the graph to the number of actors in the network. Sometimes the prestige of an actor is determined by the PageRank algorithm of the Google search engine.

Using the Hammersley-Clifford theorem, we can formally establish how the structure of a social network affects the parameters of the p * model. This theorem states that the probability of the existence of a random directed graph depends only on the complete subgraphs in the graph [24].

Nonzero model parameters correspond to the set of maximum full subgraphs (clicks) of the dependency graph. Recall that a complete subgraph is a set of vertices where each pair of them is connected by an edge, i.e. the full subgraph in the dependency graph corresponds to the set of edges of the social network, each pair of which is conditionally dependent. A maximum complete subgraph is a complete subgraph that is not fully contained in any other full subgraph. Since each subgraph of a complete subgraph is also complete, therefore, if A is the maximum clique of D, then non-zero model parameters will be for A and all its subgraphs.

Using the Hammersley-Clifford theorem can significantly simplify the process of constructing p * models. For example, the loglinear model of the Markov graph depends only on the complete set of triads and stars of size k, but not on tetrads and other complete subgraphs. Moreover, the model can be further simplified by

assuming the homogeneity of its parameters, i.e. their independence from individual actors [7].

1.4.2 Algebraic models

Algebraic models are mainly used for actor clustering problems. One of these tasks is to identify actors with equivalent relationships with other members of the group and group them into positions. Key results in this direction were obtained by H. C. White, who formalized the notion of structural equivalence.

To measure structural equivalence, R. S. Burt suggested the Euclidean distance:

$$d_{ij} = \sqrt{\sum_{k=1}^{g} \left(x_{ik} - x_{jk}\right)^2 + \left(x_{ki} - x_{kj}\right)^2}\ ,\ i \neq k\ ,\ j \neq k\ , \qquad (1.1)$$

where x_{ik} — adjacency of vertices (*i, k*), ;*g* — the number of vertices in the graph.

Another popular measure of structural equivalence is the correlation coefficient:

$$r_{ij} = \frac{\sum\left(x_{ki} - \bar{x}_{row\,i}\right)\left(x_{kj} - \bar{x}_{row\,j}\right) + \sum\left(x_{ik} - \bar{x}_{col\,i}\right)\left(x_{jk} - \bar{x}_{col\,j}\right)}{\sqrt{\sum\left(x_{ki} - \bar{x}_{row\,i}\right)^2 + \sum\left(x_{ik} - \bar{x}_{col\,i}\right)^2}\sqrt{\sum\left(x_{kj} - \bar{x}_{row\,j}\right)^2 + \sum\left(x_{jk} - \bar{x}_{col\,j}\right)^2}}, \qquad (1.2)$$
$$i \neq k,\quad j \neq k$$

where x_{ik} — adjacency of vertices (*i, k*), ($\bar{x}_{row\,i}$ та $\bar{x}_{col\,i}$) — respectively, the mean of the i-th row and column in the sociomatrix, g is the number of vertices in the graph.

Both measures are the same for completely structurally equivalent actors. In other cases, their values may vary greatly. To determine social positions, several

clustering methods were proposed, the first of which was CONCOR and hierarchical clustering.

The CONCOR (CONvergance of iterated CORrelation) method is an iterative recalculation of the correlation matrix from the initial C1 matrix.

The procedure is repeated until the elements are not +1 or -1.

The hierarchical clustering method is a greedy algorithm for splitting data by some criterion. 2 subgroups are formed at each iteration of the algorithm. Unlike CONCOR, the number of social positions obtained depends on the conditions of stopping the iterative algorithm and the chosen criterion of similarity of the actors. The similarity of the actors can be determined by the correlation matrix C1 and the Euclidean distance matrix. Consider graph G (1.3) with three equivalent groups of actors.

Calculating the distance between the vertices of the graph:

$$d = \begin{array}{c|cccccc} \backslash & n_1 & n_2 & n_3 & n_4 & n_5 & n_6 \\ n_1 & - & \sqrt{2} & 0 & \sqrt{2} & \sqrt{8} & \sqrt{8} \\ n_2 & \sqrt{2} & - & \sqrt{2} & 0 & \sqrt{6} & \sqrt{6} \\ n_3 & 0 & \sqrt{2} & - & \sqrt{2} & \sqrt{8} & \sqrt{8} \\ n_4 & \sqrt{2} & 0 & \sqrt{2} & - & \sqrt{6} & \sqrt{6} \\ n_5 & \sqrt{8} & \sqrt{6} & \sqrt{8} & \sqrt{6} & - & 0 \\ n_6 & \sqrt{8} & \sqrt{6} & \sqrt{8} & \sqrt{6} & 0 & - \end{array} \qquad (1.3)$$

From the distance matrix it can be seen that the distance between structurally equivalent actors is zero.

Group connectivity is described by the density that characterizes the degree of external connection of group members.

The density between the groups βi and β j is calculated as follows:

$$\Delta_{ij} = \frac{\sum x_{kl}}{2 g_i g_j}, \qquad k, l \in \beta_i, \beta_j . \qquad (1.4)$$

The density between the groups βi and β j is calculated as follows:

$$D = \begin{array}{c|ccc} \backslash & \beta_1 & \beta_2 & \beta_3 \\ \beta_1 & 1 & 1 & 0 \\ \beta_2 & 1 & 0 & 0 \\ \beta_3 & 0 & 0 & 1 \end{array}. \quad (1.5)$$

By block model, the density matrix D is transformed into an imaginary matrix M of binary elements that serve as indicators of linkage between groups. The matrix D is binarized at some threshold of the whole group bond or the average bond density between the groups. Block models allow us to formalize social role as a generalized pattern of relationships between positions.

The main object of social role research is the composition of relationships. For example, the social network contains the relationship A - help and F - friendship. Then through the composition of relationships, you can explore new types of relationships, such as F ° F - "my friend's friend" or A A ° - "help someone who is already helping someone." Thus, the social role can be defined by some composition of relations that has a meaningful interpretation. Accordingly, social network analysis relies on the studied compositions of relationships, finding patterns - typical communication structures, and formalizing them [8].

1.4.3 P1 and P* models

Dyads are a set of two actors (vertices of a graph) and all interactions (edges) between them. The dyad for each type of interaction can be in one of four states: there is no connection between the actors, the connection is directed from the first actor to the second, the connection is directed from the second actor to the first, mutual connections of the actors. The analysis of dyads is focused on:

– establishing the influence of the input and output degrees of the vertices of the graph on the probability of the presence of edges between them;

– the identification of the reversibility of the interaction between the actors, the degree of dependence on the properties of the actors;

– determination of conditions and directions of information transfer. For triads (three interacting actors), the transitivity of interactions is additionally investigated [9].

An important indicator is the strength of connections between actors, which is defined as a linear combination of duration, emotional richness, intimacy or confidentiality and the importance of mutual services that characterize this interaction and the corresponding graph edge. Weak connections are important sources of information for the actor, as they serve as "bridges" to other social networks where you can get additional information about a certain organization or individual [10].

Consider the dyad *Dij = (xij; xji)*. It can be in one of four states: *(0; 0)* - zero dyad (no interaction), *(0; 1)* and *(1; 0)* - asymmetric dyads, *(1; 1)* - symmetric (mutual) dyad. Create a new matrix *Y* of dimension $g \times g \times 2 \times 2$ according to the following rule

$$Y_{ijkl} = \{\ 1,\ if\ D_{ij} = (\ x_{ij} = k; x_{ji} = l),\ 0. \qquad (1.6)$$

Next, consider the following model:

$$log\ logP(Y_{ij\,00} = 1) = \lambda_{\,ij} \qquad (1.7)$$

$$log\ logP(Y_{ij\,10} = 1) = \lambda_{\,ij} + \theta \quad + \alpha_i + \beta_j \qquad (1.8)$$

$$log\ logP(Y_{ij\,01} = 1) = \lambda_{\,ij} + \theta \quad + \alpha_j + \beta_i \qquad (1.9)$$

$$log\ logP(Y_{ij\,11} = 1) = \lambda_{\,ij} + 2\theta \quad + \alpha_i + \alpha_j + \beta_j + \beta_i + \rho \qquad (1.10)$$

The parameter α describes the subject's tendency to establish interaction (its rating is the output degree D_{out}), the parameter β describes attractiveness or popularity (estimate is the input degree D_{in}), θ is the graph density (estimate is the number of edges *L*), $\rho$is the characteristic of the model's tendencies to the symmetry of dyads (estimate is number of symmetric dyads *M*). In the case of undirected interactions, there are only two types of dyads and relations will include only two parameters: an estimate of the intensity of interactions and θ.

Suppose that interactions have weights in the form of integers from *0* to *C-1* (*C* is a constant). In this case, the matrix *Y* will have the dimension $g \times g \times C \times C$ and the relation (1g) takes the form:

$$log\ logP(Y_{ijkl} = 1) = \lambda_{ij} + \theta_k \quad + \theta_l \quad + \alpha_{i(k)} + \alpha_{j(l)} + \beta_{j(k)} + \beta_{(l)i} + \rho_{kl}\ . \quad (1.11)$$

Regardless of the nature of the interactions, dyads are assumed to be statistically independent and having the same probability distribution; therefore, one can introduce some likelihood function, for example, the product of the probabilities of dyad states. Therefore, models of type p1 are also called models with independent dyads [11].

P* models.Define three new matrices for a dichotomized directed social relation:

– $X_{ij}^{+} = \{x_{kl},\ \text{при}\ x_{ij} = 1\}$- forcibly added edge *i. j*;

– $X_{ij}^{-} = \{x_{kl},\ \text{при}\ x_{ij} = 0\}$- forcibly removed rib *i. j*;

– X_{ij}^{C} - forcibly removed all ribs tied to *i.j*, moreover, this very edge in the matrix X_{ij}^{C} does not exist.

Suppose that the probability that the matrix *X* accepts the value *x* is equal to:

$$P(X = x) = \frac{exp\ exp\ (\theta' z(x))}{k(\theta)}, \quad (1.12)$$

where ’ - transposed vector of model parameters; *z(x)* - social network statistics vector; *k* - is a linear function that summarizes the products of the elements of the vector (and some coefficients, and provides the desired type of probability distribution

The main problem with this formulation of the problem is to determine k, which is a very difficult task for most social networks. However, it is possible to transform the considered log-linear model into a logit model, using the fact that the random variable xij is dichotomous.

Hence the likelihood of an existence of a rib i.j:

$$P\left(X_{ij}^{C}\right) = \frac{P\left(X=x_{ij}^{+}\right)}{P\left(X=x_{ij}^{+}\right)+P\left(X=x_{ij}^{-}\right)} = \frac{exp\ exp\ (\theta' z(x_{ij}^{+}))}{exp\ exp\ (\theta' z(x_{ij}^{+}))+exp\ exp\ (\theta' z(x_{ij}^{-}))}, \quad (1.13)$$

$$\frac{P(x_{ij}=1|X_{ij}^{C})}{P(x_{ij}=0|X_{ij}^{C})} = \frac{exp\ exp\ (\theta' z(x_{ij}^{+}))}{exp\ exp\ (\theta' z(x_{ij}^{-}))} = exp\ exp\{\theta'[z(x_{ij}^{+}) - z(x_{ij}^{-})]\}. \quad (1.14)$$

If we denote the difference in square brackets as, then the logarithm of relation (2) (logit model) will have the form:

$$\omega_{ij} = log\ log\left(\frac{P(x_{ij}=1|X_{ij}^{C})}{P(x_{ij}=0|X_{ij}^{C})}\right) = \{\theta'[z(x_{ij}^{+}) - z(x_{ij}^{-})]\} = \theta'\delta\ (x_{ij}) \quad (1.15)$$

Elements $\delta\,(x_{ij})$ - these are changes in the statistics of the social network when the value changes $x_{ij}$from *1* to *0*. Such a variant of the model in which the logarithm of the probability relation is equal to a linear combination of elements $\delta\,(x_{ij})$ is called the p * logit model for a single dichotomous social relationship. For the case of edges with weights from 0 to C 1, we will have a set of C-1 logit models.

Statistical interpretation of logistic regression models for ω_{ij} depends on the assumption of independence of quantities ω_{ij}. In the p * model, the logit models for ω_{ij} are not independent, due to which the likelihood ratio statistics are not amenable to strict statistical interpretation, although its significance can serve as a guideline for the quality of the constructed model [12].

1.4.4 Models of other types

In the last decade, with the rapid development of information technology, it has become possible to collect and process information on the topology of real networks with millions of vertices. Free access to information also facilitates interdisciplinary research on complex social networks through hybrid models [13].

One obvious area of hybridization is the graph-and-game approach, which combines graph theory that studies network structure with game theory, which examines behaviors when actors act on each other.

The graph-game approach is used to analyze the structure of the network of markets and the authorities in them, to simulate the interaction between individuals and the population as a whole, to simulate the dynamics of epidemics and the process of searching information systems.

In the context of management theory, social networks are analyzed in the light of concepts such as agent, thought, influence, trust, reputation, diffusion, and the like. Methods of analysis based on threshold models, independent cascade models, seepage and infestation models, Ising models, cellular automata and Markov circuits are applied.

The use of statistical physics models for the analysis of social networks has been rapidly expanding lately.

Percolation theory describes the behavior of related clusters in a random graph. It follows from this theory that if in the model the probability p falls below some critical value of pc, then the network will consist of isolated clusters, and in the case of p> cp, the giant component will cover the entire network. Another interesting model of statistical physics is the Albert-Barabashi fitness model. The original Albert-Barabashi model reflects the enormity of the real networks, but does not take into account the possibility of a new attraction top that many other peaks will try to connect with. To account for this, a parameter has been added to the fitness model whose value for the new vertex is chosen by some probability distribution. With a high level of fitness, new peaks can have a strong bond growth.

Because people communicate and think using predominantly verbal constructs, R. R. Yager proposed to combine rigorous mathematical models of social networks with linguistic evaluations of the characteristics of their elements. For example, when describing social networking concepts as a "leader" or "fashion legislator," the analyst uses linguistic terms from the appropriate vocabulary, and this verbal description should be consistent with rigorous mathematical models of networks [14].

1.4.5 Analysis of centrality and other local properties

There are several approaches for determining the centrality of actors. First, the central actors correspond to those vertices of the network that have the largest number of edges, i.e. have the greatest degree. Secondly, the central vertex is the vertex for which the distance to the other vertices of the network is minimal. Because of this, the path from central actors to other actors is the easiest, the former is more likely to receive information circulating in the network and to control the distribution of this information. The third measure of centrality is the intermediateness or degree of involvement of a given actor in the routes between other actors. In this case, the actor is considered central, which can control the largest number of paths in the social network both between individual actors and between their groups.

For example, the Medici family was the actor with the greatest degree in the marriage network of the Florentine elite of the early 15th century. It could control the largest number of paths in a given graph. This allows us to identify the reasons why the representatives of this family subsequently occupied a dominant position in Florence. At the same time, if we consider such characteristics of the families of the Florentine elite as wealth, antiquity of the clan and condition, political status, immediate environment, we will not find significant differences between the Medici family and other oligarchs. This example shows the promise of using social network analysis for an in-depth understanding of social processes and phenomena.

When analyzing real social interactions, a situation is often observed where one actor dominates over another, for example, actor A can give orders to actor B, and B cannot give orders to A, and he is only authorized to transmit some information to him. A model of this situation will be a directed graph with edges of two different types corresponding to relations of hierarchy and information transfer. A directed graph also arises when analyzing the fame of actors on the network. In this case, to determine the central vertices, it is necessary to consider the input and output degrees

of each vertex. Actors with a higher status will have a significantly higher input degree than the output, i.e. far more actors will know them than they themselves know [15].

To analyze the degree of centrality of various actors, we use the centrality index of the vertex xi in a connected graph G having one axis of symmetry:

$$C(x_i) = \frac{S(x_i)}{S(G)}, \tag{1.16}$$

where:

– $S(x_i) = \sum_{j=1}^{n} d(x_i, x_j)$ - the sum of the distances from the vertex xi to the remaining vertices;

– $S(G) = \sum_{i=1}^{n} S(x_i)$ - total distance in column G.

To compare graphs by their degree of centrality, we introduce the centrality index of the graph G:

$$C(G) = \frac{\min_i S(x_i)}{S(G)}. \tag{1.17}$$

C (G) can be normalized and written as:

$$\delta\ (G) = \frac{2(n-1)(1-nC\,(G))}{n-2}. \tag{1.18}$$

The coefficient ™*(G) = 0* in the case of a completely non-central graph (for example, a closed cycle) and ™*(G) = 1* for the central graph (star).

Important characteristics of a network of social interactions are balance and transitivity. Balance is the absence of situations such as “positive interaction (friendship, partnership) between A and B, as well as between A and B, but negative interaction (hostility, rivalry) between B and C”. It is argued that balanced networks are psychologically more comfortable for actors and more stable than unbalanced

ones.

Transitivity is the fulfillment of conditions of the form “if there is an interaction between A and B, and also between B and C, then there is an interaction between A and B”. These characteristics describe local connections of actors and are often used in the analysis of dyads and triads.

The main indicator that determines the differences in the resources of actors is the strength of the structural position of the actor. In the theory of network exchange, to measure this characteristic, the actor strength index i is introduced:

$$GPI_i = \sum_{k=1}^{g-1}(-1)^{k-1} P\ [i]_k, \qquad (1.19)$$

where $P\,[i]_k$ - the number of disjoint paths of length k passing through the vertex i.

The analysis of the strength of structural positions is a very effective method of constructing models that explain differences in the results of actors, for example, employees' wages or company profits from the sale of a certain product, especially in the case of a small number of observations.

Analysis of related subgroups. Connected subgroups are characterized by the presence of a large number of connections between their constituent actors and a significantly smaller number of connections with third-party actors. An analysis of such subgroups allows us to study the sustainability of social structures. The simplest case of a related group is a clique or group of actors, where each is associated with each, and other actors cannot be included in this group, since they do not have links with all members of the clique. Thus, a clique is the maximum complete subgraph of a given graph. If we analyze the processes of disseminating information in graphs, we can give another definition of a connected group as a set of actors, where the path between any two actors does not contain more than one intermediate vertex. As a result, information from one actor to another in a related group is transmitted with minimal distortion. Related groups can also be distinguished using multidimensional scaling or factor analysis of the graph matrix.

The following technique is used to analyze the stability of a group structure over time. First, a three-dimensional matrix is constructed in which the rows represent the estimates of the interactions of a given actor with all other actors given by them; columns are the actor's own estimates of interactions; along the Z axis are time periods. Further, a graph can be constructed showing changes in the structure of subgroups over time. After that, correspondent analysis is used to describe the dependencies between the rows and columns of this matrix. As a result, one can visualize changes in the status of an individual against the background of changes in the status of subgroups.

Structural equivalence of actors. This approach is the exact opposite of exploring related groups. Actors are equivalent when they occupy the same positions in the social structure, i.e. when the structure and type of interactions of these actors with others are equivalent, while the equivalent actors should not interact with each other. For example, the seller's clients of a certain product will have very little or no connections at all (as a result of which they will not be a related group), but they will all be associated with the seller, i.e. the structure of their interactions will be equivalent. As a measure of equivalence, the density of bonds with structural subgroups of actors can be used.

Along with structural equivalence, regular actor equivalence is used. In this case, the actors are equivalent when they interact in the same way with actors of the same type. For example, two children may be regularly equivalent, as they are leaders in games in the school class, but they may belong to different subgroups or cliques.

Structural equivalence determination methods allow one to analyze graphs with several types of edges. For example, in the case of a network of social interactions of countries of the world, these interactions can be trade, diplomatic contacts, wars, membership in international organizations and blocs. The graphs corresponding to each type of interaction may vary significantly, but together they represent an integrated structure of interactions between countries of the world, from which indicators of strength, influence, level of development and economic power of a given state can be derived.

To determine the structural equivalence of two actors, it is necessary to compare the structure of their interactions with other actors, i.e. you need to compare the corresponding columns in the graph matrix. This can be done by calculating the distance between these vectors (for example, according to the Euclidean or Chebyshev metric) or the coupling coefficients (for example, Pearson correlation). For directed graphs, it is necessary to consider incoming and outgoing edges; for this purpose, two corresponding matrices are considered simultaneously. If there are various types of interactions between the actors, then the matrix of graph relationships is constructed for each type of interaction, and then these matrices are compiled vertically, i.e. recorded one below the other. After that, it is possible to compare the columns of the resulting matrix using various distance measures and coupling coefficients, obtaining an estimate of the structural equivalence of the actors over the entire set of their interactions.

At the next stage, the columns are rearranged in the matrices for each type of connection in such a way as to group those of them that correspond to structurally equivalent actors. As a result, the matrix is divided into structural blocks, in each of which the density is calculated. Next, a new matrix of bonds between the found structural blocks is constructed, for example, according to the following rule: if the density of bonds between two blocks is higher than the average density of bonds in the original matrix, then the corresponding element of the new matrix is 1, otherwise it is 0. Such matrices are called block models and are a means of constructing role algebras [16].

1.4.6 Analysis end estimation of the parameters of the p * model

The likelihood function for the p * model can be written as:

$$L(\theta) = \frac{exp\, exp\, (\theta' z(x))}{k(\theta)} \tag{1.20}$$

In this case, the problem of determining the function *k* again arises. This can be avoided by using the so-called pseudo likelihood function obtained by assuming conditional independence of the edges and passing to the logit model, and having the form:

$$PL(\theta) = \prod_{i \neq j} \quad \prod_{m=1}^{r} \quad P((X_m)_{ij} = 1|(X_m^C)_{ij})^{(X_m)_{ij}} P((X_m)_{ij} = 0|(X_m^C)_{ij})^{1-(X_m)_{ij}} \quad (1.24)$$

An estimate of the maximum pseudo likelihood （corresponds to the maximum value of expression (1.21). Note that the maximum likelihood estimate does not differ from the maximum pseudo-likelihood estimate only on the class of simple graph models in which the conditional probabilities of the existence of edges do not depend on the structure of edges of a given interaction, i.e. in case of conditional independence of the ribs

We denote the estimates of the model parameters obtained using logistic regression as $(\tilde{X}_m)_{ij}$. We will evaluate the quality of approximation using statistics

$$G_{PL}^2 = 2 \sum \ (X_m)_{ij} \ log \ log \frac{(X_m)_{ij}}{(\tilde{X}_m)_{ij}}. \quad (1.22)$$

To check the statistical significance of each characteristic of the network, for example, reciprocity, two models are considered, containing this characteristic and not containing it. Significance of differences in statistics G_{PL}^2 for these models can be roughly estimated using the distribution χ^2 with the number of degrees of freedom equal to the number of model parameters associated with this characteristic. Also, the significance of the regression coefficients can be checked using Wald statistics using their approximate standard error [17].

2 SOCIAL NETWORKS DATA ANALYSIS TASKS AND PROBLEMS

2.1 Natural language processing and syntax analysis

Natural Language Processing, usually shortened as NLP, is an artificial intelligence technology branch that oprates with the interaction between various information systems and humans using the language that is common to people – natural language.

The main goal of NLP is to browse, decode, process, and make sense of the text written in human languages in such a way that it would have a value.

Majorityof the NLP techniques are built upon machine learning in order to extract valuable information from the languages.

NLP is the core of these technologies, applications and services:

– language translatorslike as Google Translate, Yandex translate and so on;

– word processors such as MS Word, Grammarly that takes a heavy use ofnatural language processingto process texts, understand them and mark errors;

– voice assistants services that are commonly used in call centers to that process user requests and provide help;

– personal smart voice assistant applications such as Google Assistant, Amazon Alexa and so on.

Natural Language processing is known to be a hard to solve problem in data science. The way human languages are constructed is what makes NLP such a difficult task to accomplish by computers.

General rules that control the flow of information using human languages are too complicated for computers to comprehend. Some rules can be over the top and too vague; for example, when a personsays a sarcastic joke to spread information. But also, some amount of these rules can be processed and described pretty easily like using the character “s” to detect the plurals.

Complete understanding of the human language consists ofcomprehension

ofthe words people use and the underlying concepts that are tied to deliveringa person thoughts and ideas.

Human beings can learn a language with little to no effort depending on the language they speak and the one they want to learn; the duality and abstract characteristics of the human languages are the thingsthat make NLP acomplicated task for computers to handle.

Syntax analysis or parsing - in computer science is the process of analyzing the input sequence of characters, in order to parse the grammatical structure according to a given formal grammar.

A parser is a program or part of a program that performs parsing. During parsing, text is framed into a data structure, usually a tree that matches the syntax of the input sequence, and is well suited for further processing. Typically, parsers work in two stages: the first identifies meaningful tokens (lexical analysis is performed), the second creates a parse tree [18].

Figure 2.1 shows a diagram of a typical parser.

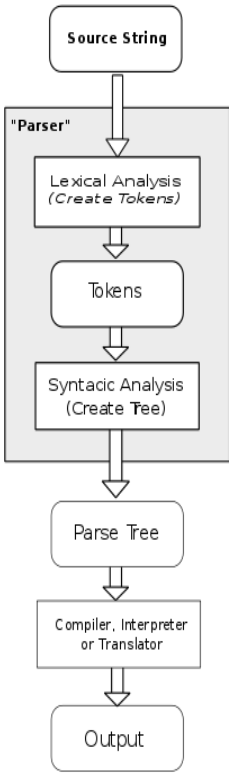


Figure 2.1 - Schematic of a typical parser

Anything that has "syntax" is automatically analyzed. Notable examples are:

– programming languages - parsing the source code of programming languages, in the process of translation (compilation or interpretation);

– structured data - data, languages for their description, layout and more. For example, XML, HTML, CSS, JSON, ini files, custom configuration files, etc.;

– building an index in the search engine;

– SQL queries (DSL language);

– mathematical expressions;

– regular expressions (which in turn can be used to automate lexical analysis);

– formal grammars;

– linguistics - natural languages. For example, machine translation and other text generators [19].

2.2 Features and applications of social network analysis

Data analysis gains popularity at a rapid paceacross the world with the rise of social networks in the early 1990s (SixDegrees, LiveJournal, Facebook, Twitter, YouTube and others). The phenomenon of socialization of personal data is related to this: facts of biography, correspondence, diaries, photos, videos, audio, travel notes, etc. became publicly available.

Thus, social networks are a unique source of information about the privacy and interests of real people. This opens up unprecedented opportunities for research and business challenges (many of which were previously impossible to solve effectively due to a lack of data), as well as the creation of support services and applications for social network users. In addition, this leads to increased interest in the collection and analysis of social data by companies and research centers.

In 2012, the Gartner think tank published a report called "The Riot Cycle for Emerging Technologies." According to the report, "Social Analytics" and "Big Data"

technologies are currently on the so-called "Peak of Excessive Expectations". In particular, Carnegie Mellon, Stanford, Oxford, INRIA, as well as Facebook, Google, Yahoo, LinkedIn and many others are actively involved in social data research. Owners of online social networking services (Facebook, Twitter) are actively investing in the development of advanced infrastructure (Cassandra, Presto, FlockDB, Thrift) and algorithmic (new search algorithms and recommendations for users, products and services) solutions for processing large volumes of custom data . Businesses are emerging and successfully developing social data repository (GNIP) services, collecting social data in specified scenarios (80legs), social analytics (DataSift), and expanding existing platforms using social data (FlipTop).

Thus, experts from research centers and companies around the world use social networking data to model social, economic, political, and other processes from the personal to the state level in order to develop mechanisms for influencing these processes, as well as to create innovative analytical and business applications and services.

However, when dealing with social data, factors such as instability in the quality of user-generated content (spam and fake accounts), problems with the privacy of users' personal data during storage and processing, as well as frequent updates to the user model and functionality must be taken into account. . All this requires constant improvement of algorithms for solving various analytical and business problems.

Social data processing also requires the development of appropriate algorithmic and infrastructural solutions that allow for their dimension. For example, the Facebook social network database now has over 1 billion user accounts and over 100 billion connections between them. Every day, users add over 200 million photos and post over 2 billion comments on various sites across the web. To date, most existing algorithms that can effectively solve current problems are not able to process data of this dimension in an acceptable time. In this regard, there is a need for new solutions that allow distributed processing and storage of data without significant loss of quality of results.

All the various online social networks can be described as a tuple of two sets: user sets and content sets. The relationship between network actors and content can be viewed through the authorship of the post and the comments. Messages can be in text, image, audio or video format. They notify network members of an event or phenomenon. Comments are a fixed format for the actors' attitude to messages [20].

The simplest analysis of online social networks is based on a self-centered graph. The vertices of this graph correspond to the central actor - "Ego" and its closest neighbors. This graph does not reflect the characteristics of the network, but it can be used to explore social roles within a group.

Some patterns can be identified by analyzing the network topology. But in most cases, you need to know the characteristics of the network, such as centrality, density, the average number of paths passing through the actor, the level of structural equivalence with a certain pattern.

Sometimes statistical characteristics are also required: variance, centralization, histograms of the degree of vertex distribution, and others. Relationship patterns in the self-centered network are meaningfully reflected in the actor's position, social or professional activity. For example, managers and administrators share a bridging template that binds individual sub-groups - the organizational units of an enterprise. The bulb pattern is well suited to describe a development team where their dense core is surrounded by layers from another structural unit with which they contact.

Each social role forms its own network structure, and can therefore be defined from it. In this way, the nature and structure of the connections can be categorized into actors, such as identifying potentially useful users and including them in a contextual search system, identifying "provocateurs", spammers, "bots", "fluders" and shielding them from other actors.

A full-scale graph should be considered in order to be guaranteed adequate ratings of the characteristics of social online networks. The methods of obtaining the corresponding data samples are quite simple and the process itself easy to parallel. But sampling may not justify the material and computational cost of receiving it. Therefore, for most researchers, obtaining, and even more analyzing, a complete

network is not possible. To adequately evaluate network characteristics, it is sufficient to obtain a representative sample that has the same characteristics as its complete network. Such a network can be assembled by randomly selecting some subset of actors.

To obtain uniform samples, there are several methods, the most famous of which are: wide crawl; accidental bypass; re-weighted casual detour; accidental bypass of Metropolis – Hasting. In general, crawl width and random crawl are sufficient to collect the data in the required amount, but the sample is far from uniform; its characteristics are offset. The Metropolis – Hasting crawl and the re-weighted random crawl yield metrics close to uniform sampling with unbiased characteristics. To find statistically unbiased network characteristics, it is enough to consider a subgraph of approximately 3 ... 10 thousand actors.

Analyzing and displaying data collected from online social networks requires the use of specialized software including UCINET, Pajek, Gephi, ORA, NetMiner, SocNet, MultiNet, GUESS, NodeXL add-ons, and the gameph, libSNA, NetworkX (Python), SNA libraries (R), SNAP (Gauss), SNAP (C ++).

To process and store data, user profiles are structured. To do this, the features and attributes are first and foremost distinguished from the account's webpage. It is also necessary to define the list of keywords and carry out their normalization using a text index, such as a bag of words and stamming approach.

In the word bag model, the text appears as an unordered set of words, without regard to grammar and word order. Next, for the information selected from the user's profile, it is necessary to conduct a stamming - to find the basics of the given word. One of the most popular and effective stamming algorithms is Porter's stammer, which is to remove prefixes and suffixes.

But this method has the following disadvantages:

– in the Ukrainian and Russian languages, where the alternation of vowels and consonants occurs, the word is usually trimmed to too short a base;

– the method is sensitive to spelling errors;

– the set of word-forming parts is different for different languages, as well as word-forming rules.

An alternative to stemming is a fuzzy word search based on finding a word set in a dictionary using a Levenstein metric. The advantages of this method are error tolerance, language independence, and step-by-step comparison of word similarity [21].

2.3 The task of collecting data about users of social networks

Social networking web interfaces are real-time data sources designed to view and interact with social network pages in a web browser or to use user data with specialized applications. Because scenarios of using social networking interfaces do not automatically allow multiple users to collect data to build a social graph, there are a number of problems:

– data privacy. Often, access to user data is only allowed for registered and authorized members of the network, which requires the support of emulating the user session through special accounts (accounts);

– poor data structure. In many cases, social networking APIs have limited functionality, which requires support to obtain static copies of HTML pages using the web-based interface, and to properly handle their dynamic portion (including performing asynchronous requests to the social server) network), extracting the necessary data using an algorithm and / or template, and constructing their structured representation, which is convenient for further automatic processing;

– restriction of access and blocking to prevent unauthorized automatic data collection and limit the load on the social network service infrastructure, service owners often impose explicit or hidden restrictions on the permissible number of requests from one user of the account and / or IP address per unit of time that takes into account the number of requests sent, as well as the support for dynamic rotation

used to collect data for user accounts and IPs;

– the dimensionality of the data necessitates the need for a parallel method of data collection, as well as methods for obtaining a representative sample of social network users (sampling) [22].

2.4 The task of defining the basic attributes of a social network user profile

When filling out their profile on a social network, users are often mistakenly or intentionally not filling in some fields or giving false information about their biography, interests and likes. In addition, on content networks (Twitter, YouTube), a user-defined profile is often limited to a set of basic attributes, which is insufficient to meet the many challenges of personalizing results.

Thus, current methods of partial identification of message authors by the values of their demographic attributes. In particular, in Internet marketing systems and recommendations, it is of particular importance to identify user demographic attributes for targeting the promotion of goods and services within user groups with the same attribute values. In addition to Internet services, such demographic characteristics are used in various disciplines: sociology, psychology, criminology, economics, personnel management, etc.

Demographic attributes can be divided into categorical categories (Gender, Nationality, Race, Marital Status, Education Level, Profession, Employment, Religious and Political Views) and Numerical (Age, Income Level).

The separation convention is that the value of a numeric attribute can be mapped to a set of categories and then be treated as a categorical attribute. In particular, age can be divided into several age categories, which is often used in practice [23].

2.5 The task of determining the response of social network users to events

Messages from social network users make up a significant portion of the textual content of today's Internet. In addition, social networks often act as informal media, where any user can publish news reports about events (Information events).

However, automatically downloading a set of messages about an unknown pre-event is a non-trivial task due to the following factors:

– a large amount of input (for example, Twitter users post several thousand messages every second);

– a large number of irrelevant / uninformative messages;

– users may describe the same event differently;

– different events may coincide in time;

– the complexity of the division of the event and its sub-events (for example, the Olympics and a specific football match within this championship).

A potential area of application is to search and compile a brief infographic of users' reactions to unknown or predefined offline and online events. Examples of such events are the next issue of a television show, sports events, natural disasters, political events, the launch of a new service for social network users and more [24].

2.6 The task of analyzing user communities on social networks

The natural feature of human society is the tendency to unite in different communities. A similar pattern is found in social networks where users are united explicitly (using networking tools to create groups and interact within them) or implicitly (linking based on shared or similar activity, role, social circle, interest, or

other attributes).

Searching for user communities is an important tool for learning and analyzing social networks, which allows you to explore the modular organization of the network and use the information obtained to solve various problems. For example, knowledge about community structure is indispensable for predicting user relationships and attributes, calculating proximity of users in the social graph, optimizing social media data flows, some analytics applications, and more.

Information about the community (modular structure) of the social network globally is being used in referral systems, spam filtering, and many other applications. Automatically certain communities of closest user contacts on a social network can be used to optimize the flow of input and output (send messages only to the "Colleagues" community, read news only from the "Close Friends" community) [25].

2.7 Analysis and comparation of existing social network analytics solutions

Social Network Analysis Software (SNA Software) is software that facilitates the quantitative or qualitative analysis of social networks by describing network features or through digital or visual representation.

Networks can consist of direct connections between nodes or indirect connections based on shared attributes, shared event attendance, or shared connections. Network functions can be at the level of individual nodes, tiaras, triads, ties and / or edges, or the entire network. For example, node-level features may include network phenomena such as intermittency and centrality, or individual attributes such as age, gender, or income. SNA software generates these features from a raw data network formatted as an edgelist, adjacency list, or adjacency matrix (also called sociomatrix), often in conjunction with (individual / layer node) attribute data. Although most network data analysis software uses ASCII text format, some software packages include the ability to use relational databases to import and / or store

network functions.

Visual representations on social networks are important for understanding network data and passing the result of analysis. Visualization often also facilitates the qualitative interpretation of network data. In terms of visualization, network analysis tools are used to change the layout, colors, size, and other properties of network representation.

Some SNA software may perform predictive analysis. This includes the use of network phenomena, such as communication for predicting individual-level outcomes (often called peer influence or infection modeling), using individual-level phenomena to predict network results, such as edge / edge formation (often called homophily models) or a particular type of triad or use network phenomena to predict other network phenomena, such as using a triad formation at time 0 to predict the formation of relationships over time [26].

The most common software packages that solve the problems of social network analysis are:NetMiner, Gephi and SocNetV.

NetMiner is an application software for conducting research analysis and visualization of big network data based on SNA (Social Network Analysis). It can be used for general research and teaching on social networks. This tool allows researchers to explore their network data visually and interactively, helping them identify the underlying patterns and structures of the network. It includes data conversion, network analysis, statistics, network data visualization, diagrams, and Python scripting programming language. It was released in 2001 as a commercial analysis software specializing in social network analysis. There is a variety of licenses not only for commercial use but also for non-commercial academic use. Current Version for Microsoft Windows (2000 or later) - 4.

NetMiner 4 features a Python script-based desktop with a script generator that allows users to generate a programmable script automatically. Users can then control the features in NetMiner 4 using a graphical interface or a scripting language. Most NetMiner features can be scripted rather than menu driven so complex batches of commands can be saved and re-executed. Various existing libraries written by Python

can be implemented in NetMiner 4 without any modifications, and you can define the usual data structures provided by Python. Users can develop their own algorithms with combinations of NetMiner features. The generated script file can be added to NetMiner 4 as one of the plugin form menus that can be used for sharing with other NetMiner users. The use of cycles, conventions, deep analysis is available. And users can create and use a batch file that runs automatically for NetMiner [27].

Gephi is an open source analysis and visualization software package written in Java on NetBeans.Gephi has been used in a number of research projects in academia, journalism, and other fields, such as visualizing the global content connectivity of the New York Times and exploring Twitter network traffic during social unrest along with more traditional network analysis topics. Gephi is widely used in the digital humanities (history, literature, political science, etc.), a community where many of its developers are involved. Gephi was inspired by LinkedIn InMaps and used for network rendering for Truthy [28].

Socializer for Social Networking (SocNetV) is a cross-platform, user-friendly program for analyzing and visualizing social networks. With SocNetV, you can draw social network graphs with just a few clicks on a virtual canvas, download field data from a file in a supported format (GraphML, GraphViz, Adjacency, EdgeList, GML, Pajek, UCINET, etc.) or scan the web to create a social network for connected web pages [28].

Edit actors and the links between them through point and click, analyze graphics and social network properties, create great HTML reports, and embed rendering layouts into the web.

Key features:

– ability to build graphs and network cohesion metrics such as density, diameter, geodetic and distance, connectivity, eccentricity, clustering ratio, reciprocity, and more;

– the presence of the following matrix procedures: adjacency plot, Laplace matrix, degree matrix, coquitation, etc;

– extended measures to analyze social networks, such as central and prestige indexes (ie own vectors and centrality of proximity, central role of interconnectivity, information centrality, central power, proximity and PageRank prestige).

– use of rapid community detection algorithms such as triad census, population census, etc;

– analysis of structural equivalence, use of hierarchical clustering, actor similarity and profile differences, Pearson coefficients;

– the layout models in this program are based either on higher popularity metrics (such as circular, center-level nodes and sizes) or on power lines (such as Kamada-Kawaii, Fruchterman-Reinhold, etc.) for meaningful social network visualizations;

– network download and network editing. Download a multi-relationship social network or create a social network yourself and add more relationships to it;

– random network creation using different models of random network generation (Barabási-Albert Scale-Free, Erdős-Rényi, Watts-Strogatz Small-World, d-regular, annular lattices, etc.);

– built-in web crawler to automatically create "social networks" from links contained in the specified source URL [29].

A comparative table of the above program complexes is given in Table 2.1.

Table 2.1 - Comparison of existing software applications in the field of social network analysis and their capabilities.

	NetMiner	Gephi	SocNetV
Distribution model	Paid subscriptions	Free	Free
Platforms	Windows	Windows, Mac OS, Linux	Windows, Mac OS, Linux
Ability to create network graphs	Exists	Exists	Exists
Ability to edit graphs	Exists	Exists	Exists
Built-in parser for social network pages	Doesn’t exist	Doesn’t exist	Exists
Support for matrix models	Exists	Exists	Exists

Support for 3rd party plugins and extensions	Exists	Exists	Doesn't exist
Detailed documentation	Highly detailed documentation	Somewhat detailed documentation	Highly detailed documentation
Support for syntax analysis	Available as a separate purchase for a paid version of the software	Doesn't exist	Doesn't exist
Support for definition of the indicator characteristics of profiles of participants of social networks	Doesn't exist	Doesn't exist	Doesn't exist

Table 2.1 - Comparison of existing software applications in the field of social network analysis and their capabilities

Existing software solves the tasks of identifying and linking actors in the network, and can also build graphs of networks and network connections. However, they do not solve the problem of analysis of actors and their indicator characteristics, which are actually members of social networks, or in other words - users.

The solution to the problem of determining the indicator characteristics of the profile of a social network participant is quite important, since it will allow to characterize and classify participants according to one or another attributes that are inherent to them. This can be useful when analyzing an audience of a particular group, or when collecting statistics from users of a particular group or resource.

An important problem that the software packages listed above do not solve is the problem of determining the response of users of a particular social network to the latest events in the world. The solution to this problem is very important because it will allow us to understand how society responds to a particular news, which will allow analysts to more easily predict the future developments in the world.

Therefore, the task of this master's work will be to create an application that will automatically determine the indicator characteristics of profiles of social network participants, as well as to syntax analyze their profiles and reactions to news and events on the network.

3 SOFTWARE DESIGN AND DEVELOPMENT

3.1 Choice of a platform

Smartphones seem to have become very popular at the same time that social media has become mainstream. People use smartphones to connect on social media. However, there are other things you can do with your smartphone as well as social media. Once you develop enough of a following, then you can use the smartphone to make money in various ways. The smartphone has resulted in more people becoming connected than ever. This has also provided people with opportunities to network and build something that is going to be great for their lives.

A mobile phone's operating system, or OS for short, is essentially the software that it runs on. The three most popular operating systems are Android and iOS. iOS runs exclusively on Apple's iPhone, while Android runs on lots of different smartphones including Samsung, Sony, Huawei, Xiaomi, OnePlus, HTC, LG, and BlackBerry [30].

Android is an operating system for smartphones, tablets, e-books, digital players, watches, fitness bracelets, game consoles, laptops, netbooks, smartbooks, Google Glasses, TVs and other devices (in 2015, support for car entertainment systems and household appliances robots).

Based on the Linux kernel and Google's own implementation of the Java virtual machine. It was originally developed by Android, Inc., which was then bought by Google. Subsequently, Google initiated the creation of the Open Handset Alliance (OHA), which is now engaged in supporting and further developing the platform. Android allows you to create Java-based applications that control the device through libraries developed by Google. Android Native Development Kit allows you to port libraries and application components written in C and other languages [31].

In 86% of smartphones sold worldwide in the second quarter of 2014, the Android operating system was installed. At a developer conference in May 2017, Google announced that more than 2 billion Android devices have been activated in

the history of Android [32].

The Android Software Development Kit (SDK) contains a comprehensive set of development tools. These include a debugger, libraries, a QEMU-based emulator, documentation, code samples, and tutorials. Supported development platforms include computers running Linux (any modern Linux desktop distribution), Mac OS x version 10.5.8 or later, and Windows 7 and later. As of March 2015, the SDK is not available for development for Android, exceptions are specialized Android applications [33].

Until the end of 2014, the officially supported integrated development environment (IDE) program was Eclipse with the Android Development Tools (ADT) plugin, although the IntelliJ IDEA IDE (all versions) fully supports Android development out of the box, just like the NetBeans IDE supports Android development through plugin. As of 2015, Android Studio, created by Google and working on IntelliJ, is the official IDE; however, developers are free to use other tools, but Google made it clear that ADT support was officially completed since the end of 2015, with the goal of focusing on Android Studio as the official Android IDE. In addition, developers can use any text editor to edit Java and XML files, and then use the command line tools (Java Development Kit and Apache Ant are optional) to create, build and debug Android applications, as well as control added Android devices (for example , start, reboot, install software packages remotely) [34].

Improving Android's SDK go hand in hand with the overall development of the Android platform. The SDK also supports older versions of Android if developers want to focus their applications on older devices. Development tools are downloadable components, so after downloading the latest version and platform, old platforms and tools can be used to test compatibility.

Android applications are packed in APK format and are stored in the / Data / App folder on the Android OS (the folder is accessible only to the root user for security reasons). The APK package contains .DEX files (a compiled executable code called Dalvik), resource files, etc.Enhancements to Android's SDK go hand-in-hand with the overall Android platform development. The SDK also supports older

versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing [35].

3.2 Choice of programming language

At Google I/O 2017, Google announced that Android will support Kotlin as a first-class programming language from now on. For this to happen, the 3.0 release of Android Studio (AS) integrated Kotlin support out of the box! The following three minor releases of AS continued to improve the Kotlin support and the tools available [36].

Kotlin is the latest statically typed open source programming language. It can run effectively on the Java Virtual Machine (JVM). Kotlin is developed by JetBrains and is officially supported by Google.

Currently, Kotlin is used to create Android apps by business leaders such as Pivotal, Atlassian, Pinterest, Evernote, and Uber. The latest App brain statistics show that in the top-end app segment of 2018, Kotlin occupies 25.3% of the market. At the same time, 40.76% of new application installations also account for applications written in Kotlin [36].

To compare Kotlin with Java, we need to understand the advantages and disadvantages of this language. Kotlin's consistent and intuitive syntax provides increased productivity for development teams. Writing and deploying a program requires less time and fewer lines of code. As a result, you get the finished application faster. It is easy for Android developers to support Kotlin code, as this language is supported in many IDEs, including Android studio, and several other SDKs. In addition, developers can work with any familiar set of tools.

Android Studio provides enhanced Kotlin support and customization tools. Developers can work simultaneously on Kotlin and Java. [37].

3.3 Software design patterns

3.3.1 Choice of an architectural pattern

There are two types of most commonly used architectural design patterns in mobile development such as Model-View-Controller and Model-View-Presenter [38].

Model-View-Presenter – a softwarearchitecturepattern derived from MVC.

The Presenter element in this template assumes the functionality of an intermediary (similar to the controller in MVC) and is responsible for managing user interface events (for example, using the mouse) in the same way as in other templates, the representation usually responds (see figure 3.2) [39].

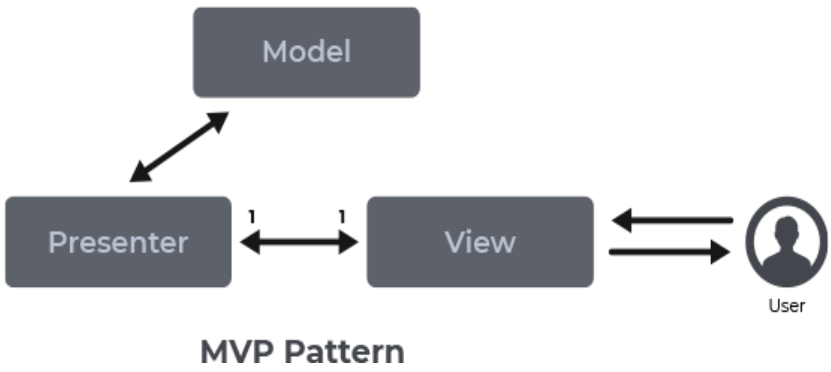


Figure 3.1 – MVP pattern

Although the Flow diagram looks same as MVC the difference is how the VIew and Presenters/Controllers interacts with each other.

It is far better than MVC as here the presenter has no Android API and it can be easily tested. The views can be tested using espresso etc. to see if the views are updated or not [40].

All this makes the MVP a perfect choice for this work.

3.3.2 Model and data layer design patterns

Often in a system entities can exist only in a single instance, for example, a system for maintaining a system message log or a display driver. In such cases, you must be able to create a single instance of a certain type, provide access to it from the outside and prohibit the creation of several instances of the same type [41].

The architecture of the Singleton pattern is based on the idea of using a global variable that has the following important properties:

1. Such a variable is always available. The lifetime of a global variable is from the start of the program to its completion.

2. Provides global access, that is, such a variable can be accessed from any part of the program.

However, it is directly impossible to use a global variable of some type, since there is a problem of ensuring the uniqueness of an instance, namely, it is possible to create several variables of the same type (for example, stack ones) [42].

To solve this problem, the Singleton pattern places control over the creation of a single object on the class itself. Access to this object is through a static member function of the class, which returns a pointer or a link to it. This object will be created only at the first call to the method, and all subsequent calls simply return its address. To ensure the uniqueness of an object, the constructors and assignment operator are declared closed [43].

The key idea in this pattern is to make the class itself responsible for controlling its instantiation (that it is instantiated only once)(see figure 3.3).

Repository mediates between the levels of the domain and data mapping layers, using an interface similar to collections for accessing objects in the domain.

A system with a complex model of the domain of definition can be simplified with the help of an additional level, for example, Data Mapper, which would isolate objects from the access code to the database. In such systems, it may be useful to add

another layer of abstraction on top of the data distribution layer (Data Mapper), in which the code for creating queries would be collected. This becomes even more important when there are many classes in the definition area or when complex, heavy queries are needed. In such cases, adding this level especially helps to reduce duplication of request code.

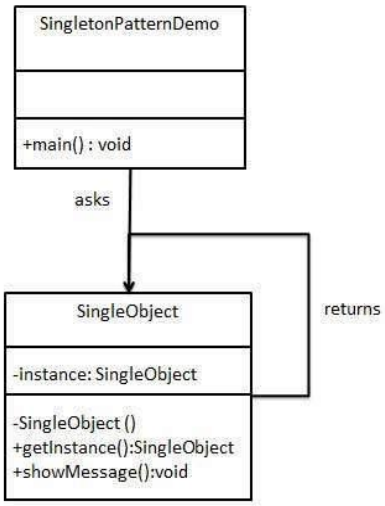


Figure 3.2 – Singleton usage class diagram

The Repository pattern mediates between the definition area layer and the data distribution layer, working like a normal collection of definition area objects. Client objects create a request description declaratively and forward them to the Repository object for processing. Objects can be added or removed from the repository, as if they form a simple collection of objects. And the data distribution code hidden in the Repository object will take care of the corresponding operations in an invisible way for the developer.

In a nutshell, the Repository pattern encapsulates the objects represented in the data warehouse and the operations performed on them, providing a more object-oriented representation of real data. The Repository also aims to achieve complete separation and one-way dependence between the levels of the domain of definition and distribution of data [44].

A commons usage scheme of repository pattern can be seen at figure 3.4.

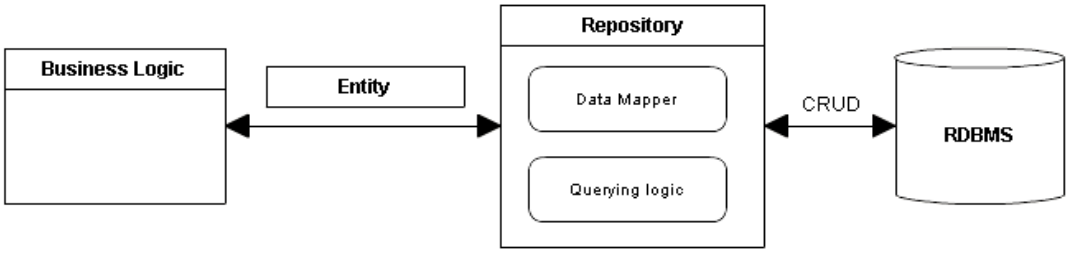


Figure 3.3 – Repository pattern usage diagram

3.4 Natural language processing

Natural Language Processing (NLP) is an actively developing scientific discipline engaged in the search for meaning and learning based on textual data.

Computer linguistics is a branch of science that studies the use of mathematical models to describe linguistic patterns. It can be divided into two large parts. One of them is studying the ways of using computer technology in linguistic research - using well-known mathematical methods (for example, statistical processing) to identify patterns. The discovered patterns are used by another part that studies the issues of comprehension of texts written in natural language - the creation of mathematical models for solving linguistic problems and the development of programs that operate on the basis of these models. This part of computer linguistics is closely related to the section of artificial intelligence involved in the development of word processing systems in natural language.

With the development of voice interfaces and chat bots, NLP has become one of the most important artificial intelligence technologies. But a complete understanding and reproduction of the meaning of the language is an extremely

difficult task, since human language has features:

- Human language is a specially designed system for conveying the meaning of words spoken or written. This is not just an exogenous signal, but a conscious transfer of information. In addition, the language is encoded so that even young children can quickly learn it.
- Human language is a discrete, symbolic, or categorical signaling system with reliability.

Categorical symbols of the language are encoded as signals for communication on several channels: sound, gestures, writing, images and so on. Moreover, the language is able to be expressed in any way.

Today, the number of useful applications in this area is growing rapidly:

– search (written or oral);
– display of suitable online advertising;
– automatic (or assisted) translation;
– mood analysis for marketing tasks;
– speech recognition and chat bots,
– voice assistants (automated customer assistance, ordering goods and services).

The Apache OpenNLP library is a machine-learning toolkit for processing natural language text. The library supports the most common functions for processing a natural language, such as language definition, tokenization, sentence segmentation, tagging of parts of speech, allocation of named objects, fragmentation, parsing and keyword splitting. Solutions to these problems are usually required for more complex text processing operations.

The Apache OpenNLP package is licensed under the Apache License. There is a fairly complete documentation for both the user and the developer of integrated solutions.The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text. OpenNLP library supprts

– tokenizationwhich splits the input text into tokens, usually words;
– sentence Segmentationthat can be used to split a whole text into sentences for

further analysis;

– part-of-speech taggingthat is used to divide the text into various grammatical elements for further analysis. Generally speaking, A part-of-speech tagger identifies the type of a word;

– named entity extractionwhich helps developers to separate names of locations, dates / events, people and other things while dynamically query processing;

– data chunking [46].

3.5Description of the program and analysis of the results

The program developed during the master's work fulfills the tasks of analyzing the reactions of users of the social network Twitter to current news, as well as determining indicator characteristics of the profiles of participants in this social network.

This development is planned to be used to collect statistical information about the response of users to an event that occurs in real time, as well as the analysis of indicator characteristics of network user profiles. The goal is to analyze statistical information and display it in the form of histograms and tables.Data collection, analysis and processing takes place on the device itself, which increases the security of the data the user is working with.

Like the vast majority of programs for Android, the program consists of a graphical interface and other application resources (graphical, text and sound resources), as well as the business logic of the application.

The graphic interface of the program is defined in xml files, which are stored in the res / layout folder. This folder can contain layouts for activities, fragments or custom views. This allows to reuse certain elements of the user interface like charts and other graphs in order to make user experience more intuitive and fluid across the application.

Figure 3.4 shows the structure of the res / layout folder.

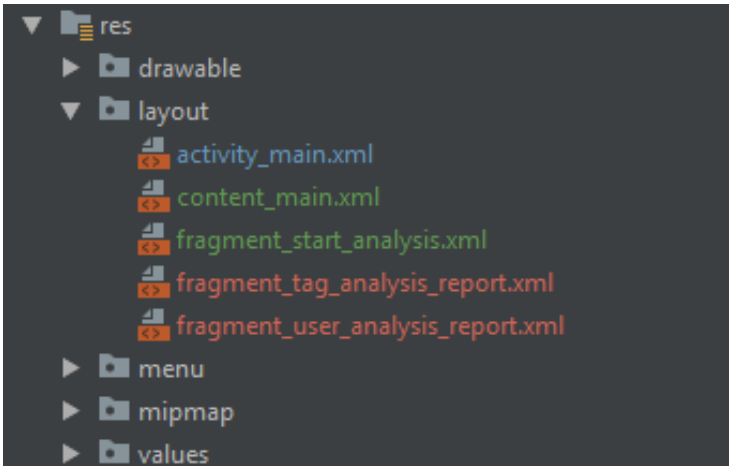


Figure 3.4–Project's res / layout folder structure

Files responsible for the business logic of the application, as well as files responsible for the logic of user interaction with the program, are located on the java / com.example.twitteranalyzer path (see figure 3.5).

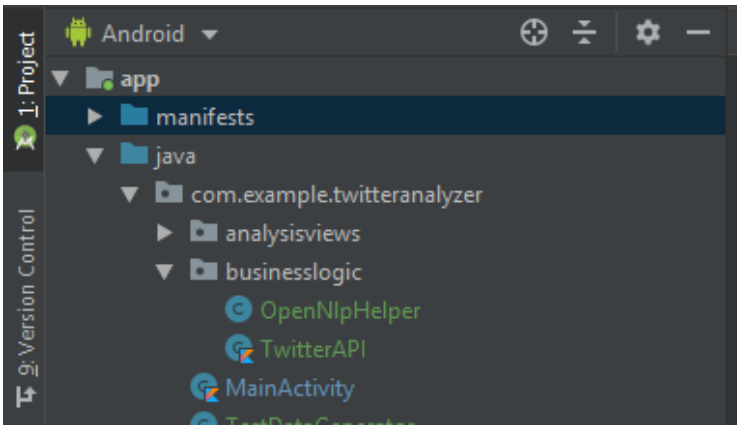


Figure 3.5–Project's java / com.example.twitteranalyzer folder structure

The structure of this folder is formed by the programmer in accordance with the selected application architecture, the design patterns used, various classes of helpers, libraries, etc.

For the program to work, you need to configure the access settings for the Twitter social network API. These parameters are key: value pairs:

- consumer_key;
- consumer_secret;
- access_token;
- access_secret.

These values are used when sending requests to the servers of the social network and necessary for the process of collecting information. To get these parameters, you must have a user account with developer privileges on Twitter. These parameters are unique for each user and allow to identify his program when accessing data on the server.

User interaction with the program starts from the main screen. In the center of the screen there is an input field where the user can enter a user nickname whose profile he wants to analyze or a hashtag to which some discussion on the network is attached. Nicknames should be entered with the use of the symbol "@" in front of the nickname itself without spaces between them. In case it is necessary to analyze the reaction to the news, the user enters the corresponding hashtag starting with the input symbol "#".

The main screen of the application is shown in Figure 3.6.

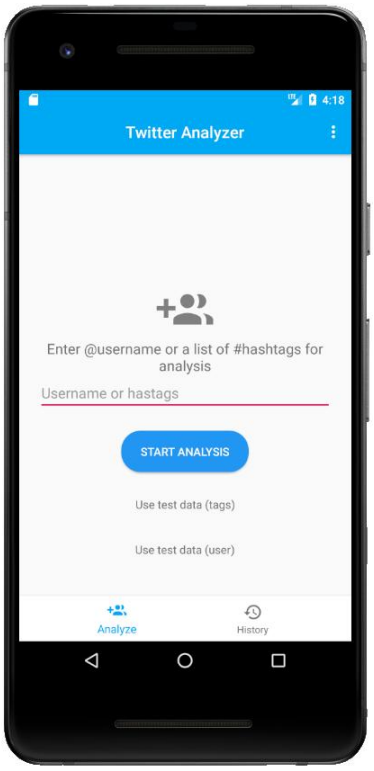


Figure 3.6 - Main screen of the application

After that, the user can click on the “Start analysis” button, which will start the process of analyzing user reactions to events on the network or analyzing the profile of the network member depending on the input.

Also, on the screen there are two additional buttons “Use test data (tags)” and “Use test data (user)”. Each of them starts the process of analyzing relevant content using pre-stored data from the Twitter network. this is necessary if the user wants to see the result of the work of the program, but does not have a Twitter developer account and therefore cannot access its API. In this case, the program will not access the Twitter API, but will use pre-prepared response files from Twitter servers.

When entering a username or hashtag, the system first calls the Twitter API. The corresponding module of the program sends the locks to the server using the previously specified access parameters, as well as the user's request. The server returns data in JSON format (Figure 3.7).

```
{
    "text": "RT @PostGradProblem: In preparation for the NFL lockout, I will be spending twi
    "truncated": true,
    "in_reply_to_user_id": null,
    "in_reply_to_status_id": null,
    "favorited": false,
    "source": "<a href=\"http://twitter.com/\" rel=\"nofollow\">Twitter for iPhone</a>",
    "in_reply_to_screen_name": null,
    "in_reply_to_status_id_str": null,
    "id_str": "54691802283900928",
    "entities": {
        "user_mentions": [
            {
                "indices": [
                    3,
                    19
                ],
                "screen_name": "PostGradProblem",
                "id_str": "271572434",
                "name": "PostGradProblems",
                "id": 271572434
            }
        ],
        "urls": [ ],
        "hashtags": [ ]
    },
    "contributors": null,
    "retweeted": false,
    "in_reply_to_user_id_str": null,
```


Figure 3.7 - Example JSON response from the Twitter API

At the stage of parsing each reaction of the user, the text of the reaction itself is extracted and divided into tokens, after which these tokens are added to the list of tokens. The list of tokens is necessary for further text processing, since many other text processing methods are needed in preliminary tokenization.

Next, the program determines the names of people, locations and dates that are in the text of tweets using the Named Entities Recognition API library Apache OpenNLP.

Then, using the Chunker API, the program selects keywords and phrases (sets of words combined in meaning and grammatically) in tweet texts. For this function to work, the program needs a list of tokens, as well as the result of the POS Tagger API, whose task is to determine the parts of speech for each word in the incoming text. This will allow the Chunker API to identify words and phrases related in meaning in the text, which in the end result will help determine the reaction of a particular user to an event on the network, as well as to the general reaction of users.

In the case of analyzing the profile of a Twitter participant, the Language Detector API is additionally applied, which, based on the model embedded in it, recognizes the language of the text and presents a list of languages that it was

possible to recognize together with the probability coefficient that this language is the language of the text.

To study and analyze the reactions of users to events, the responses of users on the Twitter network regarding events in the network united by the tag “#ukraine” were selected as input data for analysis, which means that if a given tag or word is present on a tweet, it will fall into search results.

To conduct the study, 1,500 user responses in English were collected, since the recognition model works with English speech. This figure is small for real analysis, since it covers only a small percentage of network users' reactions, but it is dictated by the limitations of the Twitter API. This limitation can be circumvented by sending requests at fixed intervals, after which the request counter is reset to zero and new requests can be sent. However, this approach will not work in the context of a mobile application, since even if it will collect data on demand in the background for a certain period of time, there is a risk that the system will stop the background process. For the test needs, this number of tweets is enough, because the goal is not to collect the most accurate statistics, but to demonstrate an approach to solving the problem.

Figure 3.8 shows the results of an analysis of user reactions to events on the network.

Figure 3.8 - user response to events on the network

The user can interact with the graphs, zooming in and scrolling through them to see all the data.

In the course of processing the results of the analysis of user reactions, the program identified the 5 most common names in the text of tweets, they turned out to be: "Donald Trump", "Joe Biden", "Rudy Giuliani", "Vladimir Putin", "Volodymyr Zelensky" (Figure 3.10).

With small amounts of data (1-5 thousand tweets), there is no need to display more results, since often the names of other people are rarely found in tweets, more often they are in the @nickname format.

Figure 3.9 shows top 5 named entities recognized in tweet texts by given query.

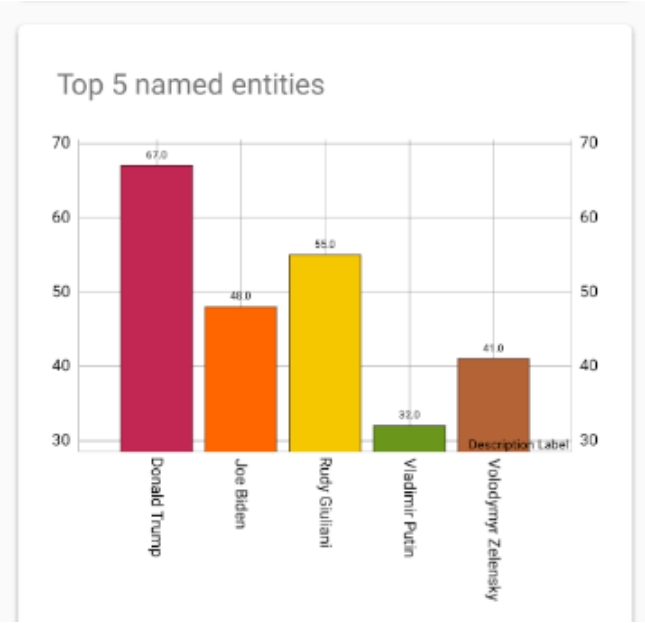


Figure 3.9 - Diagram of the most frequently encountered names in tweet texts

Figure 3.10 shows a pie chart with the mentioned locations.

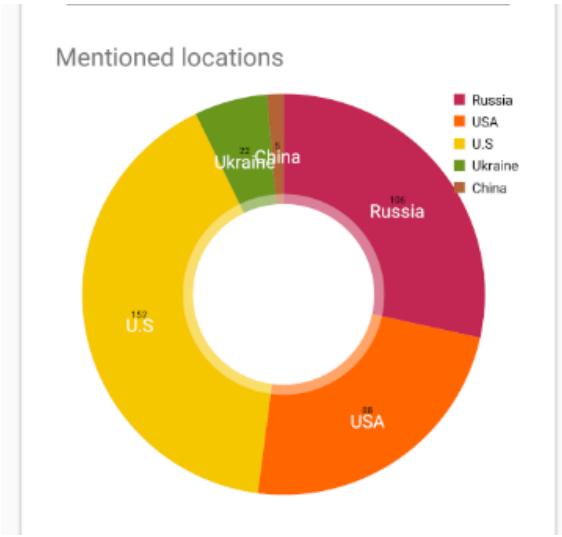


Figure 3.10 – Most mentioned locations in tweet texts

Named Entity Recognition API was able to highlight the names of locations in the text, in this case, the names of the most mentioned countries. It is worth noting that, despite the high accuracy of the results obtained, the model may not determine the names of little-known cities, which may be a problem when analyzing more “local” events in the network.

Figure 3.11 shows a diagram with the most common keywords and phrases in the texts of reactions to news on the network.

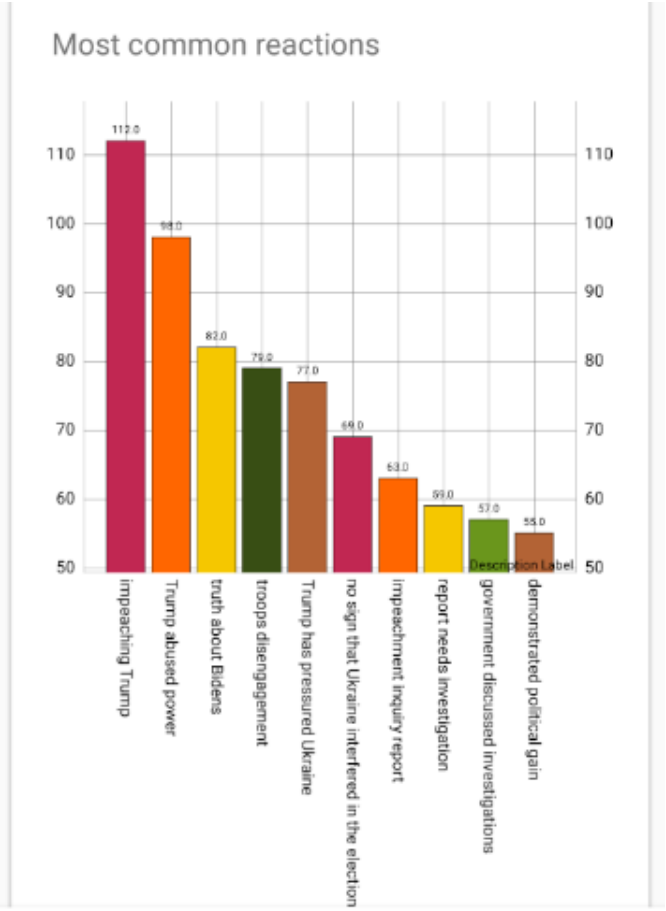


Figure 3.11 - Most common keywords and phrases

It is worth noting that the approach using Tokenization, POS Tagging and Chunking API gives much greater accuracy and greater semantic load embedded in the final results compared to using Tokenization and Tuples (for generating pairs of keywords in the text). The data approach works well in English, since it has a strict word order in a sentence, therefore there is a high probability of finding matches among the received words and phrases.

To study and analyze the profile of a member of a social network, the profile of the current US president was chosen, as well as his tweets in the amount of 2500 pieces. This figure is small for real analysis, since it covers only a small number of recent user tweets and is dictated by the limitations of the Twitter API.

Figure 3.12 shows the results of the analysis of the profile of a member of the Twitter network.

Figure 3.12 - Results of the analysis of the profile of a member of the social network Twitter

The analysis results show that the system recognized 4 languages that were supposedly used when writing tweets by a user. The result with naiballs was the accuracy of the English language, which is true. The remaining languages were recognized, since the text could contain names, names or locations that are written similar to these languages, as well as due to errors in the recognition model (Figure 3.13).

Figure 3.13 shows the pie diagram with the languages detected in the user tweets.

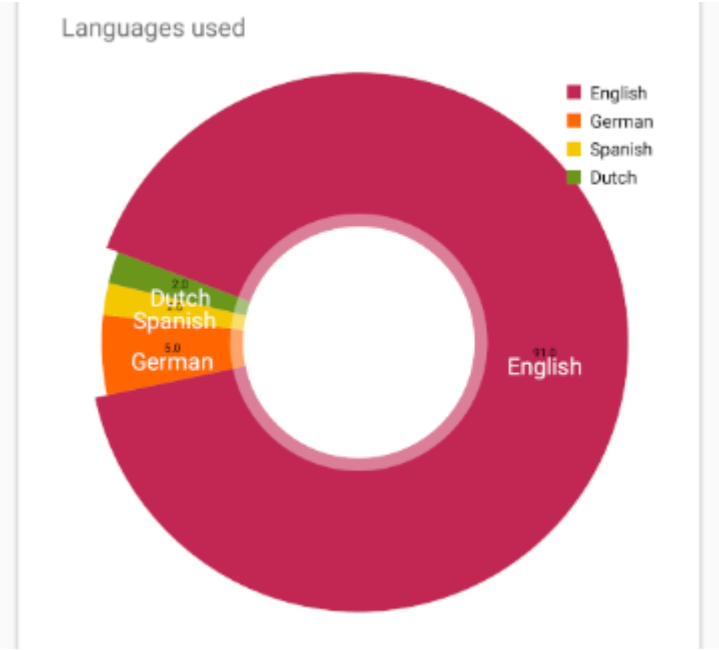


Figure 3.13 – Languages detected in the user tweets

We also analyzed the most mentioned places and personalities on the same principle as when analyzing user reactions to events on the network. These results were put on one chart in order to determine the dependence between the name and location. With this approach, between related pairs of names and locations, the difference in the frequency of references will be insignificant. However, in the case of this user and data set, this dependence was not detected.

Summing up, it can be argued that the created program successfully fulfills the tasks of analyzing the profiles of members of a social network, as well as determining the response of users to events in it.

In the course of the work, a new co-ordinated approach to solving the above problems was proposed, implemented using the Apache OpenNLP library, which made it possible to increase the accuracy of the received results, which was confirmed by the testing results of the developed program.

4 BUSINESS MODEL AND DISTRIBUTION OF INFORMATION SYSTEM

In order to become a Google developer, it is enough to register a new account or use an existing account and pay a one-time fee of $ 25. You can also invite other developers, set different roles for them and jointly put their applications for sale. At the same time, Google organizes the sale of the application itself, collecting statistics and payments, taking 30% of the sale price for itself. Applications can be free with or without advertising.

4.1 Creating a developer account on the Google Play Store

The first thing you need is a Google account. If you use Gmail, publish videos on YouTube, then, accordingly, you already have an account and you can move immediately to the second item. If you don’t have an account yet, go to the following address https://accounts.google.com/SignUp and fill out the registration form.

The developer console allows you to add your Android applications to GooglePlay, track their statistics, change the description and receive financial reports on sales of paid applications. The console itself is available at https://play.google.com/apps/publish.

Financial reports are available only for paid applications. First you need to click on the link “Set up a seller’s account”, indicate the name and parameters of the bank account or card where the earned money will be sent. Charges occur in dollars and are transferred upon reaching the threshold of $ 100. Therefore, it is better to get a dollar bank card, this is done at any bank, receive card details from the same bank and indicate them in your seller’s profile. So, you will not depend on the exchange rate and get exactly the amount that Google accrued to you [47].

4.2 Uploading the application to Google Play Store

To add a new application to Google Play, go to the developer's console at https://play.google.com/apps/publish.

Next is the "All Applications" tab and click the "Add Application" button.

A pop-up window opens where you specify the language of your application, by default it matches your settings in Google, the name of the application (up to 30 characters), and you can also start by downloading the APK file or creating a description. The order does not matter, you yourself choose what is more convenient for you in this case. In any case, the application will not be published until all required fields are completed.

The following fields are required:

– name. This is the name of your application, and is how it will be displayed to all users;

– short description. The content of your application is displayed for smartphones and tablets;

– full description. This is the long text that appears when you switch to the application on Google Play.

The following requirements apply to marketing materials:

– store requirements. These requirements are monumental and rarely subject to change; there are clear descriptions for them;

– requirements that arise from the objectives of the project: what is more relevant for the target audience of this application, what kind of marketing does the project have, etc. Sometimes it's important how the client sees it: some clients are ready to use simple screenshots and simple texts, other customers constantly change their mind about screenshots / texts, and you need to work with this.

For urgent releases, the minimum is allowed - to make marketing materials that meet the requirements of the store. In other projects, it is necessary to ensure that marketing materials are the maximum contribution to the success of the project.

Figure 4.1 shows the Google Play developer console page that allows to edit the name and description of the application.

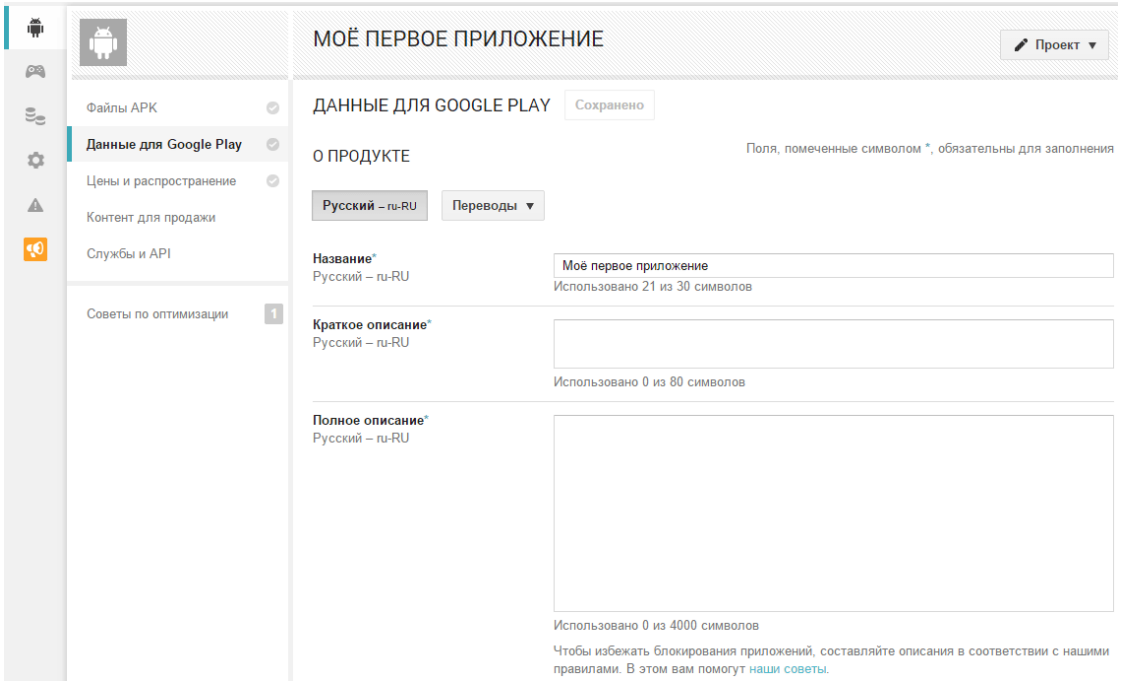


Figure 4.1 - Description of the application

It’s worth starting the preparation of marketing materials with texts. The requirements on Google Play for them are as follows:

– application name: no more than 30 characters;
– short description: no more than 80 characters;
– short description: no more than 80 characters;
– full description of no more than 4000 characters.

The maximum of screenshots available for downloading to one application is eight. At least two. How many screenshots to upload is an open question. On the one hand, we need to clearly and advantageously show the functional features of the application for the user. On the other hand, you need to form a desire for the user to download the application and see what else is there, which was not in the screenshots. Therefore, before making the maximum number of screenshots, you need to think about their need.

Store requirements for screenshots:

– JPEG format or 24-bit PNG (without alpha channel);
– at least 320 pixels;
– no more than 3840 pixels;
– The aspect ratio should not exceed 2:1.

The main purpose of screenshots is to let the user immediately understand what your application is about. Therefore, the first screenshot should be the most informative and the most attractive. The remaining screenshots show the various functional features of the application.

If your application has a version for tablets, then you need to fill in separate screenshots for seven- and ten-inch screens (thereby you prove that your application is optimized for tablets). A good solution is to make the screenshot source as large as possible (for example, ten inches) and, if necessary, make smaller copies of screenshots for small devices [48].

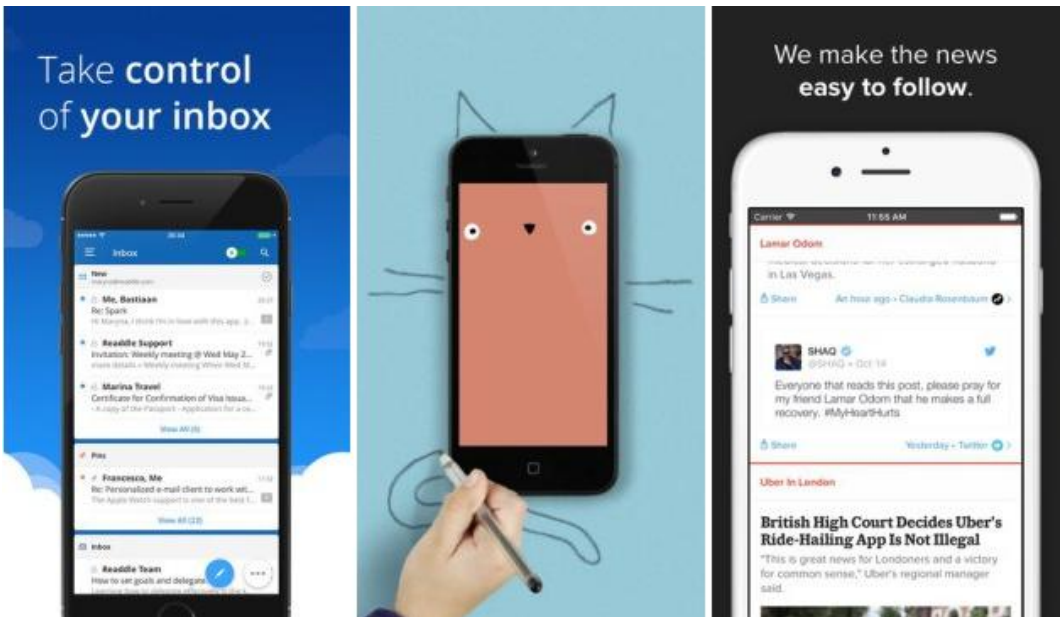


Figure 4.2 - Example of quality screenshots

4.3 Business models of the application monetization

Free application with ads. This is one of the most common ways to earn income from your program. There are no restrictions on downloading. The goal of the developer is to recruit as many users as possible. Data on their behavior is then analyzed and made available to advertisers willing to pay for advertising.

Pros of this model:

– a free mobile application can very quickly gain a solid base of users who are attracted to the free program;

– the market for mobile advertising has already exceeded the market for traditional advertising on the radio, in magazines, newspapers;

– a popular application is able to quickly gather data about user behavior patterns, which can be useful to many advertisers;

– the model is effective if targeted advertising is used.

Cons of this model:

– this is a very well-known monetization strategy, and many users are annoyed by advertising, which can lead to their outflow;

– mobile advertising is limited by the screen size of the device.

– this method of monetization does not work for niche programs that are designed to help the user perform a limited number of functions (advertising may look unnatural).

Bottom line: you will earn money by providing an ad slot in your application [49].

Freemium. As a rule, a Freemium application has a set of basic and additional functions that a user can already purchase for money. The premise of the model is to attract people to the application with a set of basic functions. Over time, some users want more, and they begin to buy additional features. The main thing here is to recruit a large number of users and interest them in your application for a long time.

A great example now is the game Angry Birds. Rovio provides the application

for free, but some features are hidden until payment, including additional levels, expanding the capabilities of birds and others. The game is very exciting, so users play it for months and some buy additional features for money.

Advantages of this model:

– this method of monetizing the application allows you to collect a user base in a relatively short time, and many of them will work with the program for months;

– people who try out the opportunities before buying are likely to become loyal users for a long time;

– a very flexible solution, can be used in the vast majority of applications;

– can be combined using an advertising model.

The disadvantages of this model are:

– If you offer too few basic functions, users will leave;

– If there are too many basic functions, users will not buy additional ones.

Bottom line: Freemium allows you to attract users with the basic version of the application so that fans of the program buy additional features for money.

Paid apps. Another very common business model, which involves payment to gain access to the application. The cost can be very different - from $ 0.99 to $ 999.99, and the developer receives money from each new user. The key to success is the ability of the developer to present the capabilities of the application in a favorable light, so as to interest the buyer at the very beginning. In the catalog, as a rule, the “killer features” of the program are indicated; this is, in the literal sense of the word, an offer that is impossible (ideally) to refuse.

Advantages of this model:

– funds are credited to the user’s account with each new program download;

– people who paid for the application are likely to use it often - because the money has already been spent;

– in a paid application, there is usually no advertising that can annoy users;

– this model motivates developers to pay maximum attention to promotion and simplifies the calculation of ROI.

Cons of this model:

– selling programs is quite difficult, because competition in application directories is very high;

– 90% of paid programs are downloaded less than 500 times a day.

Bottom line: this business model professes the principle of “pay, then take”, it is worth using for those teams that are confident that they can convince users to pay for the program.

Last year, the developer Economics Research Agency published an infographic listing the characteristics of each of the popular ways to monetize an application [50].

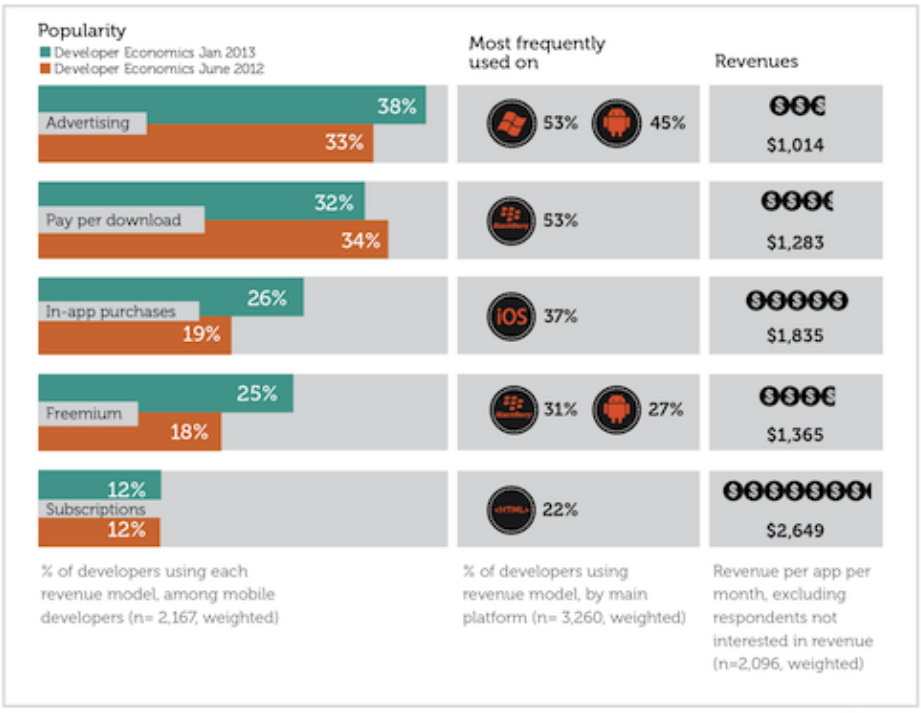


Figure 4.3 – Characteristics of each of the popular monetization methods

As it turned out, the most used business model is the demonstration of advertising in the application, and the most profitable way of monetization is subscription.

Now the ecosystem of applications is becoming more complex and diverse, so we can expect the emergence of new methods of monetization, some of which will be a certain combination of the above business models.

For this application, the optimal business model for monetization would be a mixed model consisting of Freemium and in-built advertising. Such an approach will be able to give maximum profit and at the same time give users to evaluate and test the application without the need to buy it.

CONCLUSIONS

This diploma work is devoted to the problem of automated determination of indicator characteristics of profiles of social network participants on mobile platforms and devices. This problem is relevant nowadays, and despite a great deal of research it is still not fully disclosed.In this paper the general concepts of social network theory, features of data search in them, notions of syntactic analysis and examples of its application, as well as examples of models of social networks (graph and algebraic) are considered.

The relevance of this workis substantiated on the examples of problems that will be solved in the course of the work.

The object of research and research methods are identified and describes, as well as the practical value of this work.

The software packages that solve problems in the field of social network analysis are analyzed, their peculiarities and disadvantages are identified and compared. The advantages of the developed solution on the example of problems that can be solved with its help were substantiated and fully described.In this paper, the main approaches to creating programs on the android platform are considered as well as the principles and patterns of designing applications, tools and libraries that will be used in the course of work.

A system that implements the process of automated determination of indicator characteristics of social network profiles, as well as determining the response of social network participants to certain eventshas been developed and tested on the real-world examples and data. The developed system showed that proposed approach has substantially improved the accuracy and the amount of the characteristics of user profiles inside the Twitter social network by using the approaches that combine various natural language processing technologies and approaches.

REFERENCES

1. КонцевойА. О. Автоматизованевизначенняіндикаторниххаарктеристикпрофілівучасниківсоціальнихмереж / А. О. Концевой // МатеріалидоповідейXLVIII науково-технічноїконференціїпідрозділівВінницькогонаціональноготехнічногоуніверситету, 21–23 березня. – Вінниця: ВНТУ, 2019.

2. Wasserman S. Social Network Analysis: Methods and Applications / S. Wasserman, K. Faust. — Cambridge: Cambridge University Press, 1994 — 857 p.

3. Martino F. Social Network Analysis: A brief theoretical review and further perspectives in the study of Information Technology / F. Martino, A. Spoto // Psychology Journal. — 2006. — Vol. 4, No 1. — P. 53—86.

4. Губанов Д. А. Социальные сети: модели информационного влияния, управления и противоборства / Д. А. Губанов, Д. А. Новиков. — М.: изд-во физико-математической литературы, 2010. — 228 с.

5. Butts C. T. Social network analysis. A methodological introduction // C. T. Butts // Asian Journal of Social Psychology. — 2008. — Vol. 11. — P. 13—41.

6. Borgatti S. P. Notions of Position in Social Network Analysis / S. P. Borgatti, M. G. Everett // Sociological Methodology. — 1992. — Vol. 22. — P. 1—35.

7. Carrington J. Models and Methods in Social Network Analysis / J. Carrington, S. Wasserman, G. Roberts. — Cambridge: Cambridge University Press, 2005 — 345 p.

8. Borgatti S. P. Identifying sets of key players in a social network / S. P. Borgatti // Computational & Mathematical Organization Theory. — 2006. — Vol. 12. — P. 21—34.

9. Breiger R. L. The Analysis of Social Networks / R. L. Breiger, edited by M. Hardy, A. Bryman. // Handbook of Data Analysis. — London : SAGE Publication., 2004. — P. 505—526.

10. Mislove A. Measurement and Analysis of Online Social Networks / A.

Mislove, M. Marcon, K. P. Gummadi, P. Druschel, B. Bhattacharjee// IMC '07 Proceedings of the 7th ACM SIGCOMM conference on Internet measurement — ACM New York, NY, USA., 2007. — P. 29—42.

11. Бреер В. Стохастические модели социальных сетей // Управление большими системами. – 2009. – Вып.27. –204с.

12. Wellman B. Social Structures: A Network Approach / B.Wellman, S.Berkowitz, M.Granovetter. – Cambridge University Press, 1999. – 513p.

13. Pattison P. Algebraic Models for Social Structure / P.Pattison. – Cambridge University Press, 2004. — 310p.

14. Freeman L. Research Methods in Social Network Analysis / L.Freeman, D.White, A.Romney. – Transaction Publishers, 2002. – 530p.

15. Crano W, Principles and Methods of Social Research / W.Crano. – Lawrence Erlbaum Associates, 2005. – 416p.

16. Domingos P. Mining the Network Value of Customers / P. Domingos, M. Richardson // KDD '01 Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. — ACM New-York, NY, USA., 2001. — P. 57—66.

17. МазуренкоВ.В. Оглядмоделейаналізусоціальнихмереж / В.В. Мазуренко, С.Д. Штовба // Вінницькийнаціональнийтехнічнийуніверситет "ВНТУ". – 2017. – 20 с.

18. Bokka K. Deep Learning for Natural Language Processing / K. Bokka, S. Hora, T. Jain. — Packt Publishing, 2019 — 674 p.

19. Куликова А. А. Подход к классификации пользователей социальных сетей / А. А. Куликова // Восточно-европейский журнал передовых технологий. — 2011. — Т. 3, № 2. — С. 14—18.

20. Губанов Д. А. Концептуальный подход к анализу онлайновых социальных сетей / А. А. Куликова // Восточно-европейский журнал передовых технологий. — 2011. — Т. 3, № 2. — С. 14—18.

21. Scott J. Social network analysis developments, advances, and prospects / J. Scott // Social Network Analysis and Mining. — 2011. — Vol. 1, Issue 1. — P. —

26.

22. Baym N. Social interactions across media / N.Baym // New Media & Society. - 2004 - Vol 6. - P.299–318.

23. Adamic L. How to search a social network / L.Adamic // Social Networks. - 2005 - Vol 27. - P.187–203.

24. Винник В. Социальные сети как феномен организации общества: сущность и подходы к использованию и мониторингу // Философия науки. – 2012. – №4 (55). – 146с.

25. Кондратенко Н. Р. Нечеткие модели принятия решений в задачах прогнозирования взаимоотношений в социальных группах / Н. Р. Кондратенко, С. В. Лужецкий // Наукові праці ВНТУ. — 2009 — № 2.

26. Manning C. Foundations of Statistical Natural Language Processing / C.Manning, H.Schütze. – The MIT Press, 2001. – 620 p.

27. Jurafsky D. Speech and Language Processing, 2nd Edition / D.Jurafsky, J.Martin. – Prentice Hall, 2008. – 1032 p.

28. Рудченко Д.В. Аналіз даних соціальних мереж для забезпечення безпеки / Д.В. Рудченко // Вісник національного технічного університету "ХНУ". – 2017. – 50 с.

29. Mitkov R. The Oxford Handbook of Computational Linguistics / R.Mitkov. – Oxford University Press, 2005. – 806 p.

30. Meier R. Professional Android: 4th Edition / R. Meier, I. Lake. — Wrox, 2018 — 928 p.

31. Fusco M. Java 8 in Action: Lambdas, Streams, and functional-style programming / M. Fusco, A. Mycroft. — Manning Publications, 2014 — 424 p.

32. Moscala M. Android Development with Kotlin: Enhance your skills for Android development using Kotlin / M. Moscala, I. Wodja. — Packt Publishing, 2017 — 440 p.

33. Fowler M. Patterns of Enterprise Application Architecture 1st Edition / M. Fowler. — Addison-Wesley Professional, 2002 — 560 p.

34. Buschmann F. Pattern-Oriented Software Architecture / F. Buschmann,

R. Meunier, H. Rohnert. — Wiley, 1996 — 476 p.

35. Jordan L. Practical Android Projects (Books for Professionals by Professionals) 1st ed. Edition / L. Jordan. — Apress, 2011 — 240 p.

36. Griffiths D. Head First Android Development: A Brain-Friendly Guide 1st Edition / D. Griffiths. — O'Reilly Media, 2015 — 734 p.

37. Murphy M. The Busy Coder's Guide to Advanced Android Development / M. Murphy. — CommonsWare, LLC, 2011 — 630 p.

38. Darwin F. Android Cookbook: Problems and Solutions for Android Developers 2nd Edition / F. Darwin. — O'Reilly Media, 2017 — 772 p.

39. Schildt H. Java: A Beginner's Guide, Eighth Edition, 8th Edition / H.Schildt. — McGraw-Hill Education, 2018 — 720 p.

40. Skeen J.Kotlin Programming: The Big Nerd Ranch Guide / J.Skeen. — Big Nerd Ranch Guides, 2018 — 384 p.

41. McCalister J. Mobile Apps Made Simple: The Ultimate Guide to Quickly Creating, Designing and Utilizing Mobile Apps for Your Business, 2nd Edition / J.McCalister. — CreateSpace Independent Publishing Platform, 2014 — 170 p.

42. Lee V. Mobile Applications: Architecture, Design, and Development / V.Lee. — Prentice Hall, 2004 — 368 p.

43. Iversen J. Learning Mobile App Development: A Hands-on Guide to Building Apps with iOS and Android, 1st Edition / J.Iversen, M.Eierman. — Addison-Wesley Professional, 2013 — 464 p.

44. Vasic M. Mastering Android Development with Kotlin: Deep dive into the world of Android to create robust applications with Kotlin / M.Vasic. — Packt Publishing, 2017 — 378 p.

45. Leiva A. Kotlin for Android Developers: Learn Kotlin the easy way while developing an Android App, 1st Edition / A.Leiva. — CreateSpace Independent Publishing Platform, 2016 — 240 p.

46. KamathU.Deep Learning for NLP and Speech Recognition/ U.Kamath, W.Whitaker. — Springer, 2018 — 621 p.

47. Grenadiuk A. Mobile App Marketing And Monetization / A.Grenadiuk. — Semantic Valley LLC, 2014 — 151 p.

48. Berney P. Mobile Marketing: Lessons from Global Brand Leaders on How to Make a Success of the Mobile Channel / P.Berney. — Kogan Page, 2019 — 224 p.

49. Hoss O. App Store Optimization: A Step-by-Step Guide to Boosting your App's Organic Downloads / O.Hoss. — CreateSpace Independent Publishing Platform, 2019 — 309 p.

50. Greene R. App Marketing: Top Mobile App Monetization and Promotion Strategies / R.Greene. — Tru Nobilis Pubilishing, 2017 — 104 p.

ATTACHMENTS

Appendix A(mandatory).

Technical task

APPROVED BY
Head of the department of the AIIT VNTU,
Dr. of Sci. Prof
_____R. Kvetny
“__” _____ 2019

TECHNICAL TASK
for the master's qualification work
«Information system of intellectual analysis of natural-language content of socialnetworks.Part 1. Subsystem of definition of indicator characteristics of profiles of participants of social networks "»
08-02.MQW.008.00.000 TT

Supervisor of Masters qualificationwork
Doctor of Science of
the AIITdepartmentO. Bisikalo
“___” _____ 2019
Performer: st.of the group 3ACIT-18m
A. Kontsevoi
“___” _____ 2019

Vinnytsia 2019

1. Name and field of application

1.1. Title –«Information system of intellectual analysis of natural-language content of social networks.Part 1. Subsystem of definition of indicator characteristics of profiles of participants of social networks "»

1.2. Field of application –Mobile applications for social networks analysis.

2. The basis for the development.

Theme of master's qualification was approved by order VNTU № ____ from "_____”

3. Purpose and designation.

The purpose of master's thesis is to create the mobile application that would analyze user profiles and their reactions to the news inside social networks

4. Initial data for development.

Master's qualification work is carried out for the first time. During the development, the following documents should be used:

1. Vasic M. Mastering Android Development with Kotlin: Deep dive into the world of Android to create robust applications with Kotlin / M.Vasic. — Packt Publishing, 2017 — 378 p.
2. Kamath U. Deep Learning for NLP and Speech Recognition / U. Kamath, W.Whitaker. — Springer, 2018 — 621 p
3. Mitkov R. The Oxford Handbook of Computational Linguistics / R.Mitkov. – Oxford University Press, 2005. – 806 p.

5. Requirements for development.

5.1. List of main functions:

- Visualization of information (indicators of user profile characteristics and their reactions to the events inside social network).
- Ability to fetch data from Twitter.
- Syntax analysis and natural language processing of user tweets.
- Possibility of choosing the type of analysis.

5.2. Basic technical requirements for development.

5.2.1. Requirements for the software platform: Android OS.

5.2.2. Operating system conditions:

- work on mobile devices based on Android 6.0 or higher;;

-natural language processing must happen on the device rather than in.

6. Stages of development.

6.1 Explanatory note:

1	Analysis of the characteristics and social network models	
2	Analysis of the current tasks and problems connected with social network analysis and user characteristics identification	
3	Practical implementation and analysis of the results	
4	Approval of research results	
5	Publications	
6	Designing an explanatory note, graphic material and presentation	
7	Master’s work defense	

7. The order of control and acceptance.

7.1. The progress of master's qualification work is supervised by the head of the work. The border control should be conducted until _____.

7.2. The project certification is carried out on preliminary protection. Preliminary defense of master's qualification work is to be held until _____.

7.3. The final decision on the assessment of the quality of the master's qualification work is taken at a meeting of the DEC. The defense of master's qualification work is to hold _____.

AppendixB (mandatory).
List of graphical materials

Head of the department of the AIIT	_____	Dr. of Sci. Prof R. Kvetny
	(signature)	(name, surname, degree, academic status)
Scientific supervisor	_____	Dr. of Sci. Prof O. Bisikalo
	(signature)	(name, surname, degree, academic status)
Technical control	_____	Dr. of Sci. Prof O. Bisikalo
	(signature)	(name, surname, degree, academic status)
Regulatory control	_____	Dr. of Sci. Prof O. Bisikalo
	(signature)	(name, surname, degree, academic status)
Reviewer	_____	PhD.,As. Prof O. Kovalyuk
	(signature)	(name, surname, degree, academic status)
Student of the 3ACIT-18m	_____	A. Kontsevoi
	(signature)	(name,surname)

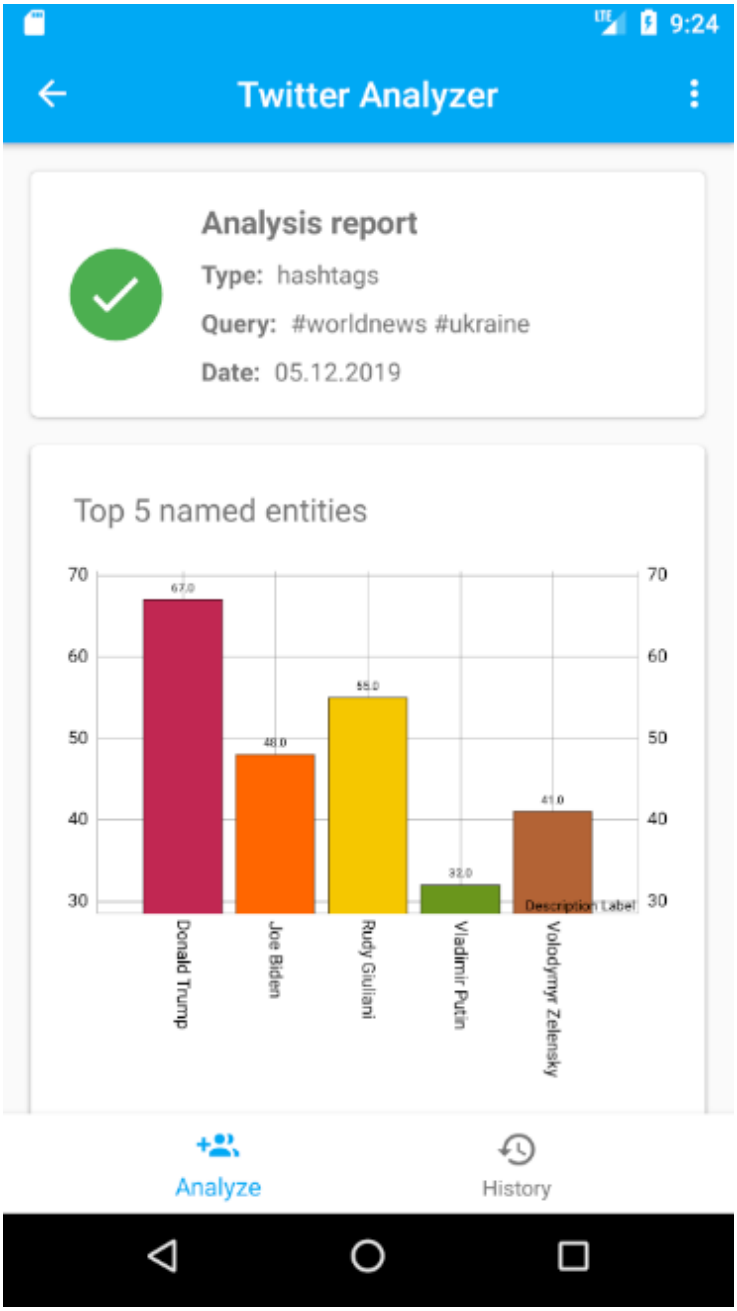


Figure B.1 – Example of named entities recognition when analyzing user reactions to the events in social network

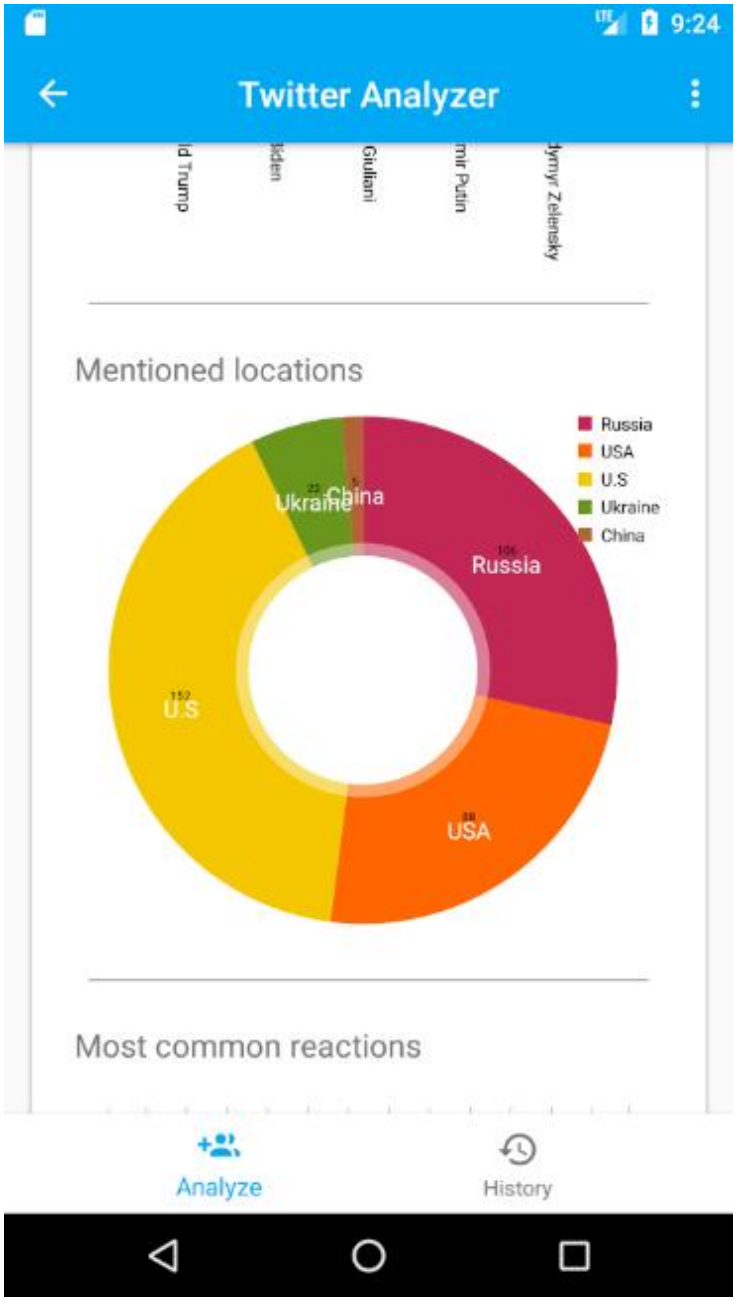
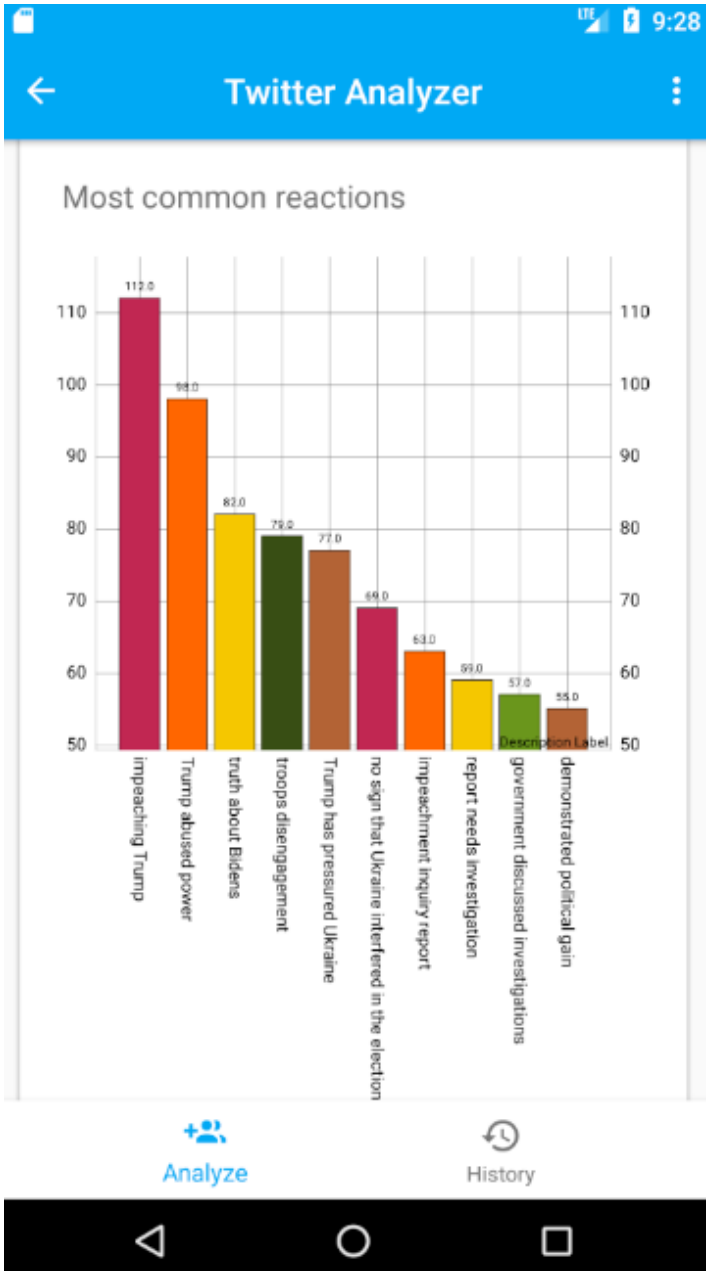


Figure B.2 – Example of location names recognition when analyzing user reactions to the events in social network



FigureB.3 – Example of most common phrases used when users were discussing the even in the social network

Figure B.4 – Example of user profile analysis report

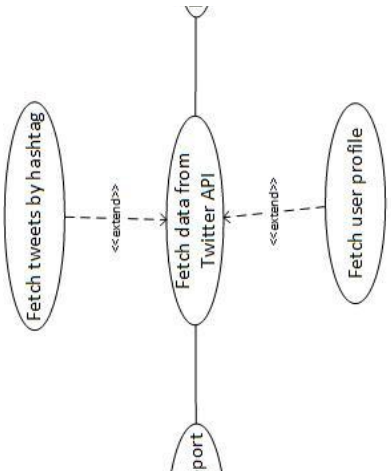


Figure B.5 – Use-case UML diagram of the developed information system

The continuation of the attachment B

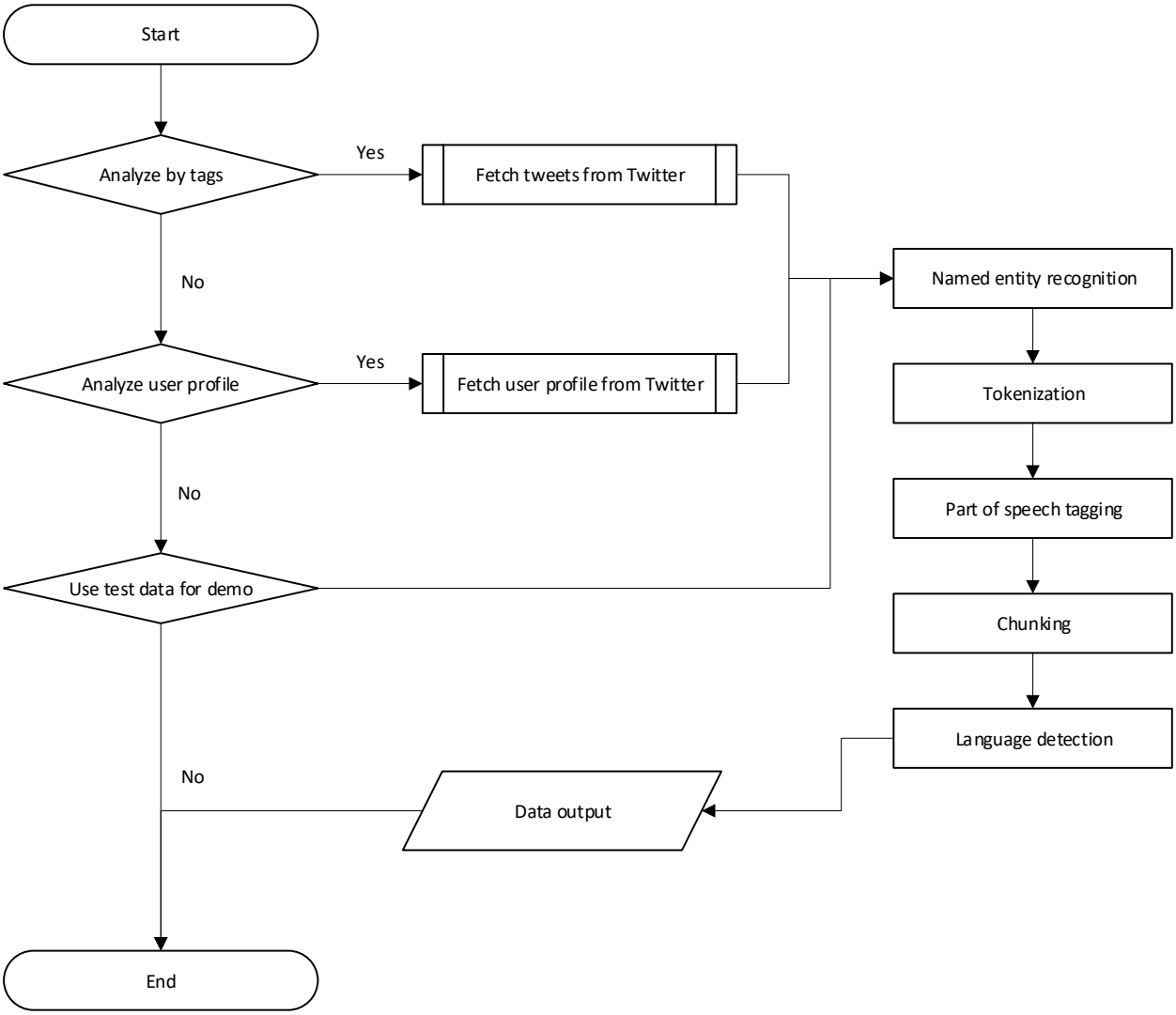


Figure B.6 – Diagram of the developed information system flow

AppendixC (mandatory).

Software code listing

TwitterApiHelper.kt

```
package com.example.twitteranalyzer.businesslogic

import twitter4j.*
import twitter4j.conf.ConfigurationBuilder

class TwitterAPI private constructor() {

    private val twitter: Twitter

    fun getTweets(searchQuery: String): List<Status>? {

        return try {

val query = Query(searchQuery)
val result: QueryResult

            result = twitter.search(query)
result.tweets

        } catch (te: TwitterException) {

te.printStackTrace()
            null

        }

    }

    fun searchUsers(searchQuery: String): ResponseList<User>? {

        return try {

            var page = 1
            var users: ResponseList<User>?

            do {

                users = twitter.searchUsers(searchQuery, page)
                page++

            } while (users?.size != 0 && page < 50)
            users
```

```
        } catch (te: TwitterException) {

te.printStackTrace()
            null

        }

    }

    fun getFollowerIDs(user: String): IDs? {

        var ids: IDs? = null

        if (user.isNotEmpty()) {
            ids = twitter.getFollowersIDs(user, 0)
        }

        return ids

    }

    fun getUserListStatuses(user: String): ResponseList<Status>
{

val page = Paging(1)
        var statuses: ResponseList<Status>

        do {
            statuses =
twitter.getUserListStatuses(user.toInt().toLong(), page)

page.page = page.page + 1

        } while (statuses.size> 0 &&page.page<= 10)

        return statuses
    }

    fun getTweetsByGivenLocation(latitude: Double, longtitude:
Double, radius: Double): List<Status>? {

        return try {

            var query = Query()
query.geoCode(GeoLocation(latitude, longtitude), radius, "km")

            var result: QueryResult
            result = twitter.search(query)

val tweets = result.tweets
```

```
            try {
Thread.sleep(500)
            } catch (e: InterruptedException) {
e.printStackTrace()
            }

            return tweets

        } catch (te: TwitterException) {

te.printStackTrace()
            null

        }

    }

    companion object {

        var instance: TwitterAPI? = null

get() {
                if (field == null) {
                    field =
TwitterAPI()
                }

                return field
            }

            private set
    }

init {

valcb = ConfigurationBuilder()

cb.setDebugEnabled(true)
.setOAuthConsumerKey(TwitterApiCredentials.consumerKey)
.setOAuthConsumerSecret(TwitterApiCredentials.consumerSecret)
.setOAuthAccessToken(TwitterApiCredentials.accessToken)
.setOAuthAccessTokenSecret(TwitterApiCredentials.accessTokenSecr
et)

valtf = TwitterFactory(cb.build())

        twitter = tf.instance
    }
}
```

OpenNlpHelper.kt

```
package com.example.twitteranalyzer.businesslogic;

import org.intellij.lang.annotations.Language;

import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Arrays;

import opennlp.tools.chunker.ChunkerME;
import opennlp.tools.chunker.ChunkerModel;
import opennlp.tools.namefind.NameFinderME;
import opennlp.tools.namefind.TokenNameFinderModel;
import opennlp.tools.postag.POSModel;
import opennlp.tools.postag.POSTaggerME;
import opennlp.tools.sentdetect.SentenceDetectorME;
import opennlp.tools.sentdetect.SentenceModel;
import opennlp.tools.tokenize.SimpleTokenizer;
import opennlp.tools.tokenize.TokenizerME;
import opennlp.tools.tokenize.TokenizerModel;
import opennlp.tools.util.InputStreamFactory;
import opennlp.tools.util.MarkableFileInputStreamFactory;
import opennlp.tools.util.ObjectStream;
import opennlp.tools.util.PlainTextByLineStream;
import opennlp.tools.util.Span;
import opennlp.tools.util.TrainingParameters;

public class OpenNlpHelper {

    public ArrayList<String>splitEngTextIntoSentences(String
inputText) throws Exception {

ArrayList<String> sentences = new ArrayList<>();

InputStream is = getClass().getResourceAsStream("/models/en-
sent.bin");

        if (is != null) {

SentenceModel model = new SentenceModel(is);
SentenceDetectorMEsentenceDetector = new
SentenceDetectorME(model);

            String detectedSentences[] =
sentenceDetector.sentDetect(inputText);

sentences.addAll(Arrays.asList(detectedSentences));

        }
```

```
        return sentences;

    }

    public ArrayList<String>findNames(String paragraph) throws
IOException {

InputStreaminputStream = getClass().getResourceAsStream("/en-
ner-person.bin");
TokenNameFinderModel model = new
TokenNameFinderModel(inputStream);

NameFinderMEnameFinder = new NameFinderME(model);
String[] tokens = tokenize(paragraph);

        Span nameSpans[] = nameFinder.find(tokens);

ArrayList<String> names = new ArrayList<>();

for(Span span: nameSpans) {
names.add(tokens[span.getStart()]);
        }

        return names;
    }

    public ArrayList<String>findLocations(String paragraph)
throws IOException {

InputStreaminputStreamNameFinder =
getClass().getResourceAsStream("/en-ner-location.bin");
TokenNameFinderModel model = new
TokenNameFinderModel(inputStreamNameFinder);

NameFinderMElocFinder = new NameFinderME(model);
String[] tokens = tokenize(paragraph);

        Span locationSpans[] = locFinder.find(tokens);

ArrayList<String> locations = new ArrayList<>();

for(Span span: locationSpans) {
locations.add(tokens[span.getStart()]);
        }

        return locations;

    }

    public ArrayList<String>chunkText(String text) throws
IOException {
```

```
SimpleTokenizer tokenizer = SimpleTokenizer.INSTANCE;
String[] tokens = tokenizer.tokenize(text);

InputStreaminputStreamPOSTagger =
getClass().getResourceAsStream("/models/en-pos-maxent.bin");
POSModelposModel = new POSModel(inputStreamPOSTagger);
POSTaggerMEposTagger = new POSTaggerME(posModel);

        String tags[] = posTagger.tag(tokens);

InputStreaminputStreamChunker =
getClass().getResourceAsStream("/models/en-chunker.bin");
ChunkerModelchunkerModel  = new
ChunkerModel(inputStreamChunker);

ChunkerMEchunker = new ChunkerME(chunkerModel);

String[] chunks = chunker.chunk(tokens, tags);

return  newArrayList<String>(Arrays.asList(chunks));
    }

    public ArrayList<String>predictLanguages(String text) throws
IOException  {

InputStreamFactorydataIn = new MarkableFileInputStreamFactory(
new File("src/main/resources/models/DoccatSample.txt"));
ObjectStreamlineStream = new PlainTextByLineStream(dataIn, "UTF-
8");

LanguageDetectorSampleStreamsampleStream = new
LanguageDetectorSampleStream(lineStream);

TrainingParameters params = new TrainingParameters();
params.put(TrainingParameters.ITERATIONS_PARAM, 100);
params.put(TrainingParameters.CUTOFF_PARAM, 5);
params.put("DataIndexer", "TwoPass");
params.put(TrainingParameters.ALGORITHM_PARAM, "NAIVEBAYES");

LanguageDetectorModel model =
LanguageDetectorME.train(sampleStream, params, new
LanguageDetectorFactory());

LanguageDetectorld = new LanguageDetectorME(model);
Language[] languages = ld.predictLanguages(text);

ArrayList<String>languagesAsStrings = new ArrayList<>();
        for (Language language: languages) {
languagesAsStrings.add(language.toString());
        }

        return languagesAsStrings;
```

```
    }

    public String[] tokenize(String sentence) throws
IOException{
InputStreaminputStreamTokenizer =
getClass().getResourceAsStream("/en-token.bin");
TokenizerModeltokenModel = new
TokenizerModel(inputStreamTokenizer);
TokenizerME tokenizer = new TokenizerME(tokenModel);
        return tokenizer.tokenize(sentence);
    }

}
```