

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Тилемисова Жанерке Касымқызы

Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелерді құру  
алгоритмін әзірлеу

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд, профессор

Молдагулова А.Н. Молдагулова

" 20 " 05 2022 ж.

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

Тақырыбы: «Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелерді құру алгоритмін әзірлеу»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы

Орындаған

Тилемисова Жанерке Касымқызы

Рецензент

PhD, қауымдастырылған профессор

Самбетбаева М.А. Самбетбаева

" 18 " 05 2022 ж.

Ғылыми жетекші

PhD, т.ғ.к., қауымд. профессор

Еримбетова А.С. Еримбетова

" 19 " 05 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

**БЕКІТЕМІН**

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд., профессор

А.Н. Молдагулова

" 20 " 05 2022 ж.

Дипломдық жобаны орындауға

**ТАПСЫРМА**

Білім алушы Тилемисова Жанерке Касымқызына

Тақырыбы: «Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелерді құру алгоритмін әзірлеу»

Академиялық мәселелер жөніндегі проректоры бұйрығының № 489-170/24 " 12 " 2021 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі " 24 " 05 2022 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма, жоба диаграммалары түрінде ақпаратты жинау, деректер қорына сақтау, тестілеу, тексеруге арналған программалық қаматамаларды жасау жүргізілген.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

- а) тақырып бойынша талдау және есептің қойылымын жасау;
- б) жобаны жобалау және пәндік сала бойынша талдау;
- в) пайдаланушы интерфейсін жобалау және дамыту;
- г) бағдарламаны құру, кітапханаларды қосу, деректерді қосу және тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):

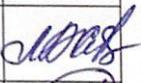
презентация 20 слайдпен берілген құжат түрінде ұсынылған.

Ұсынылған негізгі әдебиеттер саны: 20

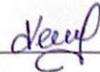
Дипломдық жобаны орындау  
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Дипломдық жобаның жоспарын құру	14.01.2022	орындалды
2. Тапсырма қойылымы және программалау ортасын таңдау	18.01.2022	орындалды
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім бойынша есеп беру жазбасын дайындау	01.02.2022	орындалды
4. Дипломның екінші және үшінші бөлімдерін, жобалау сызбаларын дайындау	15.02.2022	орындалды
5. Жобаның веб-қосымшасын тестілеуден өткізу	18.03.2022	орындалды
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2022	орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Жекамбаева М.Н. PhD, қауымдастырылған профессор	20.05.22	
Бағдарламалық бөлім	Марғұлан Қ. тех.ғыл.магистірі, лектор	20.05.22ж	

Ғылыми жетекші \_\_\_\_\_  Еримбетова А.С.

Тапсырманы орындауға алған білім алушы \_\_\_\_\_  Тилемисова Ж.К

Күні «17» 11 2021 ж

## АНДАТПА

Дипломдық жұмыс тақырыбы «Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелерді құру алгоритмін әзірлеу».

Дипломдық жұмыс: кіріспеден, абзацтарға бөлінген үш тараудан, қорытындыдан және пайдаланылған әдебиеттер тізімінен тұрады.

Кіріспе зерттеу тақырыбының өзектілігін, зерттеу мақсаты мен міндеттерін негіздейді.

Бірінші тарауда әлеуметтік желілерде мәтіндерді классификациялау, мәтінді классификациялау мәселесінің ресми тұжырымы, мәтінді классификациялаудың түрлері мен классификация әдістерінің сипаттамасы, сондай-ақ қолданыстағы әлеуметтік желілердің жаңа мүмкіндіктері мен кемшіліктері қарастырылған.

Екінші тарау әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелер мен әлеуметтік желілердің функцияларына талдау жасалынып, талдау деректерін алу үшін ВКонтакте әлеуметтік желісінің API интерфейсін пайдалану арқылы құрылымдарына зерттеу жүргізілді.

Үшінші тарауда ақпараттық жүйені құру алгоритмі әзірленіп, бағдарламалау тілін таңдау негіздемесі жүргізілді. Сонымен қоса, мобильдік қосымшаны құру жолдары мен оны ісқу асыру жолдарын қамтиды.

Қорытынды бөлімінде зерттеу нәтижесінде алынған негізгі қорытындылар беріледі.

## АННОТАЦИЯ

Тема дипломной работы «Разработка алгоритмов создания информационных систем классификации текстов в социальных сетях».

Дипломная работа состоит из введения, трех глав, разделенных на параграфы, заключения и списка литературы.

Во введении обосновывается актуальность темы исследования, цели и задачи исследования.

В первой главе речь идет о классификации текстов в социальных сетях, официальной постановке задачи классификации текстов, описании типов классификации текстов и методов классификации, а также о новых особенностях и недостатках существующих социальных сетей.

Во второй главе анализируются функции информационных систем и социальных сетей для классификации текстов социальных сетей, а также для изучения структур социальной сети ВКонтакте с использованием интерфейса API для получения аналитических данных.

В третьей главе разработан алгоритм создания информационной системы и обоснован выбор языка программирования. Он также включает способы создания мобильного приложения и способы его реализации.

В заключительном разделе представлены основные результаты исследования.

## ANNOTATION

The theme of the diploma work is "Development of algorithms for creating information systems for classifying texts in social networks."

The thesis consists of an introduction, three chapters, divided into paragraphs, a conclusion and a list of references.

The introduction substantiates the relevance of the research topic, goals and objectives of the research.

The first chapter deals with the classification of texts in social networks, the official formulation of the problem of text classification, a description of the types of text classification and classification methods, as well as new features and shortcomings of existing social networks.

The second chapter analyzes the functions of information systems and social networks for classifying social network texts, as well as for studying the structures of the social network VKontakte using the API interface to obtain analytical data.

In the third chapter, an algorithm for creating an information system is developed and the choice of a programming language is substantiated. It also includes how to create a mobile application and how to implement it.

The final section presents the main results of the study.

## МАЗМҰНЫ

	Кіріспе	9
1	Әлеуметтік желілерде мәтіндерді классификациялау	11
1.1	Мәтінді классификациялау мәселесінің ресми тұжырымы	11
1.2	Классификация әдістерінің сипаттамасы	13
1.3	Мәтінді классификациялаудың түрлері	16
1.4	Қолданыстағы әлеуметтік желілердің жаңа мүмкіндіктері мен кемшіліктері	17
2	Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелерге талдау	24
2.1	Әлеуметтік желілердің функцияларын анализдеу	24
2.2	Әлеуметтік желілерде мәтіндердің классификациялық талдауы	32
2.3	Талдау деректерін алу үшін ВКонтакте әлеуметтік желісінің API интерфейсін пайдалану	33
3	Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелердің қосымшасын құру және оны ісқосу асыру	39
3.1	Бағдарламалау тілін таңдау негіздемесі	39
3.2	Жүйенің интерфейсін Angular фреймворгінде жобалау	40
	Қорытынды	50
	Пайдаланылған әдебиеттер тізімі	51
	А қосымшасы. Техникалық тапсырма	53
	Б қосымшасы. Бағдарлама мәтіні	54

## КІРІСПЕ

**Дипломдық жұмыстың өзектілігі** - қазіргі әлемде көптеген электронды құрылғылар, сонымен қатар байланыс құралы болып табылатын веб және мобильді қосымшалар бар. Адамдар интернет-форумдарды, чаттарды пайдаланады, блогтар жасайды, әлеуметтік желілер арқылы бір-бірімен араласады. Осындай өзара әрекеттесу нәтижесінде пайдаланушылардың өздері туралы, олардың басқа адамдарға, олардың өмірінде және әлемде болып жатқан оқиғаларға қатынасы туралы ақпараттың үлкен көлемі пайда болады. Мұндай ақпарат қоғамда болып жатқан процестерді модельдеуде, оған қатысушылардың мінез-құлқын болжауда және т.б. пайдалы болуы мүмкін. Сондықтан виртуалды әлеуметтік желілерде немесе форумдарда қамтылған ақпаратты жинау және талдау әдістері зерттеу үшін үлкен қызығушылық тудырады.

Интернеттегі ақпараттың көпшілігі табиғи тілде мәтін түрінде беріледі. Бұл оны өңдеуді қиындатады және есептеу лингвистика әдістерін қолдануды талап етеді. Осыған байланысты лингвистикалық зерттеулердің өзектілігі, құрылымдалмаған мәтіндік ақпарат массивтерінен фактілерді алудың жаңа тиімді бағдарламалық жүйелерін әзірлеу, желідегі хабарламалардың өзін талдауға да, таратылатын ақпарат көздерін анықтауға бағытталған ақпаратты классификациялау және кластерлеу қазіргі уақытта артып келеді.

**Дипломдық жұмыстың мақсаты** - әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық платформа құру. Және де платформа құру барысында әлеуметтік желі ретінде арнайы Вконтакте әлеуметтік желісін таңдадым. Яғни, платформа Вконтакте әлеуметтік желісіндегі топтарда жазылатын ақпараттаға анализ жасау болып табылады. Платформада арнайы бір постты таңдап, сол постта не жайлы жазылған, және де қанша бағалау мен пікір қалдырылғаны туралы анализ жасап, білуге болады.

Әлеуметтік желілер пайдаланушыларды қай жерде болса да өз ойларымен және әсерлерімен бөлісу мүмкіндігімен пайда болғаннан бері танымал. Жыл сайын әлеуметтік желіні пайдаланушылар саны артып келеді, адамдар әртүрлі ресурстарда көбірек аккаунттар жасайды, әртүрлі дүкендердегі тауарлар туралы өз пікірлерін жиі білдіреді, жаңа гаджеттерге жаңа шолулар үнемі жазылады, әлемде болып жатқан әртүрлі оқиғалар талқыланады. Осылайша, әлеуметтік желілерде әртүрлі аспектілер бойынша ақпараттың үлкен базасы қалыптасады. Адамдардың өнімдер немесе оқиғалар туралы пікірлерін білу үлкен артықшылық болуы мүмкін, мысалы, маркетингтік ұйымдар, ірі корпорациялар және т.б. Әлеуметтік желідегі адамдардың пікірлеріне қарап белгілі бір мәселелер бойынша қолданушылардың мәлімдемелеріне талдау жасауға болады.

Сонымен қатар, соңғы онжылдық әртүрлі әлеуметтік жүйелердің танымалдылығының өсуімен сипатталады: блогтар (мысалы: Livejournal, Twitter), форумдар (тақырыптық қауымдастықтың үлкен саны, мысалы:

Tripadvisor - саяхатшылар форумы, Cyberforum - форум. бағдарламашыларға арналған), әлеуметтік желілер (мысалы: ВКонтакте, Facebook, Instagram), белгілі бір объект туралы пікірлерді жинақтайтын интернет қызметтері (мысалы: Yandex.Market, Kinopoisk, Amazon). Күн сайын мұндай ресурстарды пайдаланушылар көптеген хабарламалар, материалдар орналастырады, белгілі бір объект туралы пікір білдіреді. Осындай пікірлерге сүйене отырып, адам қызықтыратын қызметті пайдалану немесе қолданбау, қалаған тауарды сатып алу немесе алмау туралы қорытынды жасай алады. Қазіргі уақытта пікірлерді бақылаудың бұл тәсілінің пайдалылығына қарамастан, бірқатар елеулі кемшіліктер бар: деректердің үлкен көлемін қолмен өңдеу, пікірлерді табу және оларды эмоционалды бағалау және нәтижені ыңғайлы пішінге келтіру қиындықтары.

Жоғарыда айтылғандардың негізінде пікірлерді автоматты түрде табу және талдау жүйесін құру қажеттілігі туындады. Ұқсас тапсырма ақпаратты іздеу мен компьютерлік лингвистиканың қиылысында орналасқан пәнде қойылады – ол мәтіндік пікірді талдау болып табылады. Әлеуметтік желілердегі мәтіндердің тональдылығы – мәтінде айтылған қандай да бір объектіге мәлімдеме авторының эмоционалды қатынасы, сондай-ақ пікір бойынша біз бір нәрсені эмоционалды бағалауды түсінеміз.

# 1 Әлеуметтік желілерде мәтіндерді классификациялау

## 1.1 Мәтінді классификациялау мәселесінің ресми тұжырымы

Мәтіндерді классификациялау – ол мәтіннің мазмұнына қарай мәтінді бірнеше категориялардың біріне жатқызудан тұратын ақпаратты іздеу міндеттерінің бірі.

Жіктеуді қолмен жасалған ережелер жинағын автоматты түрде немесе машиналық оқыту әдістерін пайдалану арқылы толығымен қолмен жасауға болады.

Мәтіндердің классификациясын кластерлеуден ажырата білу керек, соңғы жағдайда мәтіндер де белгілі бір критерийлер бойынша топтастырылады, бірақ алдын ала анықталған категориялар жоқ.

Мәтінді классификациялаудың бірнеше тәсілдері бар.

Бірінші тәсіл – ол классификация әрқашан компьютермен орындала бермейді. Мысалы, кәдімгі кітапханада тақырыптық тақырыптарды кітапханашы кітаптарға қолмен тағайындайды. Мұндай қолмен жіктеу қымбатқа түседі және үлкен жылдамдықпен құжаттардың үлкен санын жіктеу қажет болған жағдайда қолданылмайды.

Екінші тәсіл - мәтінді бір немесе басқа санатқа тағайындауға болатын ережелерді жазу. Мысалы, осы ережелердің бірі келесідей болуы мүмкін: «егер мәтінде туынды және теңдеу сөздері болса, онда оны математик ретінде жіктеңіз». Пән саласын жақсы білетін және тұрақты сөз тіркестерін жазуға дағдыланған сарапшы оларды жіктеу үшін кіріс құжаттарға автоматты түрде қолданылатын ережелер жинағын жасай алады. Бұл тәсіл алдыңғыға қарағанда жақсы, өйткені жіктеу процесі автоматтандырылған, сондықтан өңделген құжаттардың саны іс жүзінде шексіз. Сонымен қатар, қолмен құрастыру ережелері машиналық оқытуға қарағанда жақсы классификация дәлдігін бере алады. Дегенмен, ережелерді құру және сақтау маманның тұрақты күш-жігерін қажет етеді.

Соңғы тәсіл машиналық оқытуға негізделген. Бұл тәсілде ережелер жиынтығы немесе жалпы алғанда мәтіндік классификатордың шешім критерийі оқу деректерінен автоматты түрде есептеледі (басқаша айтқанда, классификатор оқытылады). Оқыту деректері әр сыныптың жақсы үлгі құжаттарының саны болып табылады. Машиналық оқыту қолмен белгілеу қажеттілігін. Бірақ белгілеу ережелерді жазуға қарағанда оңайырақ жұмыс. Сонымен қатар, таңбалау жүйені пайдаланудың қалыпты режимінде орындалуы мүмкін. Мысалы, электрондық пошта бағдарламасы хабарламаларды спам ретінде белгілеу мүмкіндігіне ие болуы мүмкін, осылайша классификатор - спам сүзгісі үшін оқу жинағын құрайды. Осылайша, машиналық оқытуға негізделген мәтінді жіктеу бақылаудағы оқытудың мысалы болып табылады, мұнда мұғалім сыныптар жинағын анықтайтын және оқу жинағын белгілейтін тұлға болып табылады.

Классификацияны кластерлеуден ажырату керек. Құжаттарды жіктеу кезінде санаттар алдын ала белгіленеді, кластерлеу кезінде олар белгіленбейді, тіпті олардың саны туралы ақпарат болмауы мүмкін.

Формальды түрде жіктеу есебінің тұжырымын былайша жазуға болады.

$D = \{d_1, \dots, d_{|D|}\}$  құжаттар жинағы және  $C = \{c_1, \dots, c_{|C|}\}$  мүмкін санаттар (сыныптар) жинағы бар. Белгісіз мақсат функциясы  $\Phi: D \times C = \{0, 1\}$  формуламен берілген.

$$\Phi(d_j, c_i) = \begin{cases} 0, & \text{если } d_j \notin c_i, \\ 1, & \text{если } d_j \in c_i \end{cases} \quad (1)$$

$\Phi'$ -ге барынша жақын,  $\Phi$  классификаторын құру талап етіледі. Мәселені осылай тұжырымдағанда, құжаттың өзінен алуға болатынын қоспағанда, санаттар мен құжаттар туралы қосымша ақпарат жоқ екенін атап өткен жөн.

Егер классификатор нақты жауап берсе:

$$\Phi': D \times C = \{0, 1\}, \quad (2)$$

онда классификация дәл деп аталады.

Егер жіктеуіш құжаттың ұқсастық дәрежесін (категоризация күйінің мәні) анықтаса:

$$CSV: D = [0, 1], \quad (3)$$

онда классификация шекті деп аталады.

Жалпы жағдайда мұғаліммен оқыту процесі (прецеденттер бойынша оқыту, жетекшілік ететін оқыту) келесідей. Жүйе бұрын белгісіз заңдылыққа қатысты мысалдар жиынтығымен ұсынылған. Бұл жиынды кейде оқу жиыны  $L$  деп те атайды. Ол классификаторды жаттықтыру және классификатор ең жақсы нәтиже беретін оның параметрлерінің мәнін анықтау үшін қолданылады. Әрі қарай жүйеде шешім қабылдау ережелері әзірленеді, оның көмегімен мысалдар жиынтығы берілген сыныптарға бөлінеді. Бөлу сапасы  $T$  мысалдарының сынақ жинағымен тексеріледі. Бұл жағдайда шарттар қажет

Тест мысалдар жинағы  $T$ . Бұл жағдайда шарттар қажет

$$L \cap T = \emptyset, \quad (4)$$

$$\Omega = L \cup T \subset C \times D \quad (5)$$

Көптеген мысалдар үшін мақсат функциясының  $\Phi$  мәндері белгілі.

Егер әрбір құжаттың тапсырмасында бір ғана категория сәйкес келуі мүмкін болса, онда бірімәнді жіктеу, ал егер категориялардың ерікті саны болса, онда анық емес жіктеу болады.

Бір мәнді жіктеу жағдайында екілік жіктеу бар, бұл кезде құжаттар жиынтығы екі бөлек санатқа бөлінуі керек. Мысалы, мәлімдемелердің көңіл-күйін анықтау міндеті (оң немесе теріс бояу) немесе спамды анықтау тапсырмасы (хабарлама спам болып табылады ма) екілік классификатордың көмегімен шешіледі.

Жіктеу мәселесін шешу келесі төрт кезеңнен тұрады:

- құжаттарды алдын ала өңдеу және индекстеу;

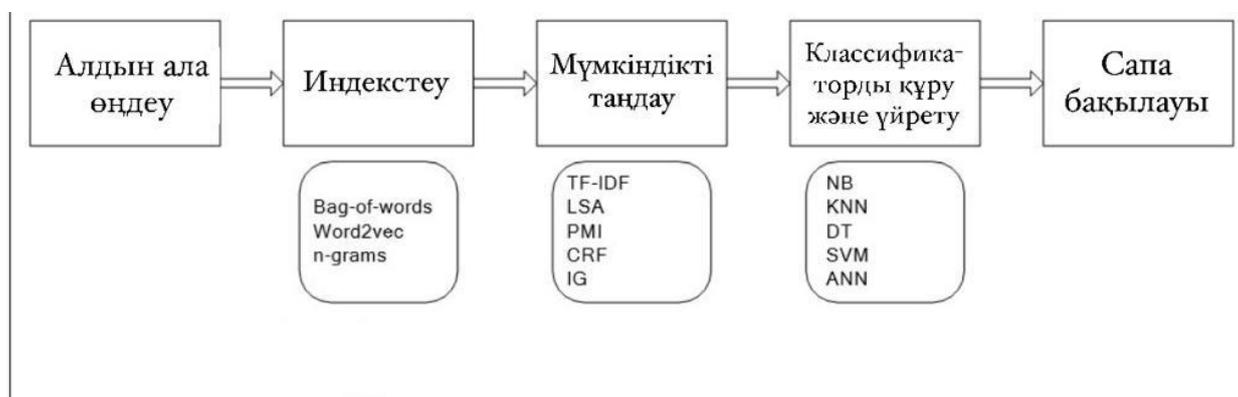
- призма өлшемін кішірейту;
- машиналық оқыту әдістерін қолдана отырып, классификаторды құру және оқыту;
- жіктеу сапасын бағалау.

Белгілі бір жіктеу алгоритмін таңдаған кезде олардың әрқайсысының ерекшеліктерін ескеру қажет. Жіктеу белгілерінің жиынтығын, олардың санын және салмақты есептеу әдістерін анықтау мәселесі әлі де шешімін таппаған. Терең оқыту алгоритмдерінде жіктеу дәлдігі дұрыс өлшемдегі оқу үлгісінің болуына өте тәуелді. Мұндай үлгіні дайындау өте қиын процесс. Осы уақытқа дейін оқыту кезеңінде кейбір алгоритмдердің параметрлерін таңдау мәселесі де ашық күйінде қалып отыр.

Әрі қарай кезеңдердің әрқайсысы егжей-тегжейлі қарастырылады, классификаторларды құрудың әртүрлі алгоритмдері мен әдістері, олармен жүргізілген тәжірибелер және осы тәжірибелердің нәтижелері сипатталады.

## 1.2 Классификация әдістерінің сипаттамасы

1.1-суретте жіктеу процесінің жалпы схемасы берілген. Оның әрбір кезеңін қарастырайық.



### 1.1-сурет – Мәтінді автоматты классификациялау процесінің кезеңдері

*Мәтінді алдын ала өңдеу және индекстеу.* Мәтінді алдын ала өңдеу токенизацияны, функционалды сөздерді алып тастауды (біріккен сөздер, көсемшелер, мақалалар және т.б. сияқты мағыналық бейтарап сөздер) қамтиды. Әрі қарай морфологиялық талдау жүргізіледі (сөйлеу бөліктері бойынша таңбалау және стемматтау жүргізіледі). Бұл кеңістіктің өлшемін айтарлықтай азайтуға мүмкіндік береді. Нәтижесінде мәтінде кездесетін барлық маңызды сөздер мәтіннің белгілері ретінде әрекет етеді.

Мәтінді индекстеу – мәтінді әрі қарай өңдеуге ыңғайлы ұсынуға аударатын кейбір сандық мәтін үлгісін құру.

Мысалы, сөз қалтасының моделі мәтінді сөздердің көп өлшемді векторы және олардың мәтіндегі салмағы ретінде көрсетуге мүмкіндік береді. Басқаша айтқанда, әрбір мәтін көпөлшемді кеңістіктегі вектор болып табылады, оның координаталары сөздердің сандарына сәйкес келеді, ал координаттардың мәндері салмақтардың мәндеріне сәйкес келеді.

Басқа кең таралған индекстеу моделі Word2vec. Ол әрбір сөзді контекстік (байланысты) сөздер туралы ақпаратты қамтитын вектор ретінде көрсетеді.

Басқа индекстеу моделі n-граммдарды, яғни іргелес символдар тізбегін есепке алуға негізделген.

Оқу және сынақ мәтіндері үшін бірдей индекстеу әдісін қолдану керек екені анық.

*Мүмкіндік кеңістігінің өлшемін азайту.* Әртүрлі жіктеу әдістерінің есептеу күрделілігі мүмкіндіктер кеңістігінің өлшеміне тікелей байланысты. Сондықтан классификатордың тиімді жұмыс істеуі үшін олар жиі қолданылатын белгілердің (терминдердің) санын азайтуға жүгінеді.

Терминдер кеңістігінің өлшемін азайту арқылы классификатор маңызды және маңызды емес, оқыту деректерінің кездейсоқ немесе қате сипаттамаларына назар аударатын құбылыс - артық сәйкестендірудің әсерін азайтуға болады. Қайта оқытылған классификатор оқытылған жағдайларда жақсы жұмыс істейді және сынақ деректерінде әлдеқайда нашар жұмыс істейді. Шамадан тыс қондырмау үшін оқыту мысалдарының саны қолданылатын терминдер санына сәйкес болуы керек. Кейбір жағдайларда мүмкіндіктер кеңістігінің өлшемін 10 есеге (тіпті 100 есеге) азайту тек классификатордың жұмысының аздап нашарлауына әкелуі мүмкін.

Мәтін мүмкіндіктерінің салмағын анықтаудың бірнеше жолы бар. Ең көп тарағаны TF-IDF функциясын есептеу. Негізгі идеясы - белгілі бір мәтінде жиілігі жоғары және басқа мәтіндерде жиі кездесетін сөздерге көбірек салмақ беру.

*Машиналық оқыту әдістерін қолдана отырып, классификаторды құру және оқыту.* Келесі жіктеу әдістерін ажыратуға болады:

- ықтималдық (мысалы, NB);
- метрикалық (мысалы, KNN);
- логикалық (мысалы, DT);
- сызықтық (мысалы, SVM);
- логистикалық регрессия;
- жасанды нейрондық желілерге негізделген әдістер (мысалы, FFBP, RNN, DAN2, CNN).

*Шешім ағаштары әдісі (Decision Trees, DT)* жіктеудің логикалық әдістеріне жатады.

Шешім ағашы - бұл функциялар жиынтығымен сипатталған объектілерді (біздің жағдайда мәтіндік құжаттар) жіктейтін циклдік график. Ағаштың әрбір түйінінде белгілердің біріне сәйкес тармақталу шарты болады. Әрбір түйінде таңдалған атрибут үшін қанша мән болса, сонша тармақ болады. Жіктеу процесінде бір түйіннен екіншісіне бірізді ауысулар объектінің белгілерінің

мәндеріне сәйкес жүзеге асырылады. Ағаштың жапырақтарының біріне (соңғы түйіндеріне) жеткенде классификация аяқталды деп саналады. Бұл жапырақтың мәні қарастырылып отырған нысанның қай сыныпқа жататынын анықтайды. Тәжірибеде әдетте екілік шешім ағаштары қолданылады, оларда жиектер бойынша жылжыту туралы шешім құжатта мүмкіндіктің болуын жай ғана тексеру арқылы жүзеге асырылады. Егер төлсипат мәні болса белгілі бір мәннен аз болса, бір тармақ таңдалады, егер үлкен немесе тең болса, екіншісі таңдалынып алынады.

Бұрын ұсынылған басқа тәсілдерден айырмашылығы, шешім ағашы тәсілі символдық (яғни сандық емес) алгоритмдерге жатады.

Екілік шешім ағашын құру алгоритмі келесі қадамдардан тұрады.

Ағаштың бірінші түйіні жасалады, ол барлық қол жетімді мүмкіндіктермен ұсынылған барлық құжаттарды қамтиды. Әрбір құжат үшін мүмкіндік векторының өлшемі  $n$  болады, себебі  $d = (t_1, \dots, t_n)$ .

Ағаштың ағымдағы түйіні үшін ең қолайлы  $t_k$  мүмкіндігі және оның ең жақсы шекаралық мәні  $v_k$  таңдалады.

Таңдалған мүмкіндіктің шекаралық мәніне сүйене отырып, оқыту үлгісі екі бөлікке бөлінеді. Әрі қарай таңдалған белгі осы бөліктердегі фрагменттердің сипаттамасына қосылмайды, яғни бөліктердегі фрагменттер  $n - 1$  өлшемі бар вектормен бейнеленеді.

Нәтижедегі ішкі жиындар олардың әрқайсысында тек бір сыныптың мәтіндері немесе мәтіндерді ажырататын мүмкіндіктер болмайынша, ұқсас өңделеді.

Ең қолайлы белгіні таңдау туралы айтқанда, олар, әдетте, жиілік белгісін, яғни мәтіннің мәтінде оның кездесу жиілігін табу мүмкіндігін беретін кез келген ерекшелігін білдіреді. Бөлу үшін ең жақсы мүмкіндік - осы қадамда санаттар туралы максималды ақпаратты беретін мүмкіндік. Мәтінге арналған мұндай белгі, мысалы, түйінді сөз болуы мүмкін. Осы тұрғыдан алғанда кез келген жиілік ерекшелігін айнымалы деп санауға болады. Содан кейін ең қолайлы екі ерекшеліктің арасындағы таңдау екі айнымалының байланыс дәрежесін бағалауға дейін төмендейді. Сондықтан қолайлы белгіні таңдау үшін тәжірибеде гипотезаларды тексерудің әртүрлі критерийлері қолданылады, яғни өзара сәйкестікке қойылған екі айнымалының байланыс дәрежесін сандық бағалау критерийлері, мұнда  $0$  айнымалылардың толық тәуелсіздігіне сәйкес келеді, ал  $1$  олардың максималды тәуелділігіне.

Екі айнымалы шама арасындағы байланысты зерттеу үшін бұл айнымалылардың бірлескен таралу көрінісін күтпеген кесте түрінде (факторлық кесте немесе ерекшелік жиілік матрицасы) пайдалану ыңғайлы. Бұл статистикалық қатынастарды зерттеудің ең әмбебап құралы, өйткені ол кез келген өлшем деңгейімен айнымалыларды көрсете алады. Кездейсоқтық кестелері әртүрлі статистикалық тестілердің көмегімен екі белгі арасындағы байланыстың гипотезасын тексеру үшін жиі қолданылады: Фишер сынағы (Фишердің дәл сынағы), Пирсонның жақсылық сынағы (хи-квадрат сынағы), Крамер сынағы, Студент тесті ( $t$ -Студент критерийі), т.б.

*Әдістің артықшылықтары:*

- алгоритмді салыстырмалы түрде қарапайым программалық жүзеге асыру;
- алгоритм нәтижелерін оңай түсіндіру.

*Әдістің кемшілігі:*

- бастапқы мәліметтердегі шектен тыс көрсеткіштерге қатысты алгоритмнің тұрақсыздығы және нақты нәтижелерді алу үшін мәліметтердің көптігі.

*Қолдау векторлық машинасы (SVM) – сызықтық жіктеу әдісі.* Қазіргі уақытта бұл әдіс ең жақсылардың бірі болып саналады. Жіктеуді қажет ететін мәтіндер жинағын қарастырыңыз. Онымен  $|D|$  өлшемді кеңістіктегі нүктелер жиынын байланыстырайық.

Әртүрлі класстарға жататын нүктелерді гипержазықтықтың көмегімен бөлуге болатын болса (екі өлшемді жағдайда түзу гипержазықтық болып табылады) нүктелер жиынын сызықтық бөлінетін деп атайды. Бұл жағдайда есепті шешудің айқын жолы – бір класстың барлық нүктелері оның бір жағында, ал екінші класстың барлық нүктелері екінші жағында жататындай түзу жүргізу. Содан кейін белгісіз нүктелерді жіктеу үшін олардың түзудің қай жағы болатынын қарау жеткілікті болады.

Жалпы жағдайда біздің шартымызды қанағаттандыратын гипержазықтықтардың (сызықтардың) шексіз жиынын салуға болады. Қол жетімді нүктелерден мүмкіндігінше алыс түзу сызықты таңдаған дұрыс екені анық. Тірек векторлық машинада түзу мен нүктелер жиынының арақашықтығы түзу мен жиыннан оған ең жақын нүкте арасындағы қашықтық болып табылады. Дәл осы қашықтық бұл әдісте максималды болады. Екі параллель гипержазықтыққа дейінгі қашықтықты барынша арттыратын гипержазықтық бөлу деп аталады (2-суретте ол  $L$  әрпімен белгіленген). Параллель гипержазықтықтарға жақын нүктелер тірек векторлары деп аталады, олар арқылы үзік сызықтар өтеді. Басқаша айтқанда, алгоритм осы параллель гипержазықтықтар арасындағы айырмашылық немесе қашықтық неғұрлым үлкен болса, классификатордың орташа қатесі соғұрлым аз болады деген болжаммен жұмыс істейді, өйткені сыныптар арасындағы алшақтықты барынша арттыру сенімдірек жіктеуге ықпал етеді.

Тәжірибеде деректер құрылымы жиі белгісіз және бөлуші гипержазықтықты құру өте сирек мүмкін, бұл үлгінің сызықтық бөлінуіне кепілдік беру мүмкін емес дегенді білдіреді. Алгоритм бір сыныпқа тағайындайтын құжаттар болуы мүмкін, бірақ іс жүзінде олар керісінше болуы керек. Мұндай деректер шеткі мәндер деп аталады, олар әдіс қатесін тудырады, сондықтан оларды елемей жақсы болар еді. Бұл сызықтық бөлінбеу мәселесінің мәні.

### **1.3 Мәтінді классификациялаудың түрлері**

Әлеуметтік желідегі пайдаланушы ақпаратының және пайдаланушы мазмұнының үнемі өсіп келе жатқан көлеміне байланысты, оны әлеуметтік және коммерциялық салаларда тиімді пайдалана алу үшін бұл ақпаратты өңдеу қажеттілігі өсуде. Сондықтан қазіргі уақытта жалпы табиғи тілдегі мәтіндерді өңдеу мәселелеріне және оның ішінде мәтінді жіктеудің қосалқы міндетіне көп көңіл бөлінеді. Мәтінді жіктеу әдістері әртүрлі практикалық мәселелерді шешу үшін қолданылады, мысалы:

- спамды сүзу;
- контекстік жарнаманы таңдау және көрсету;
- автоматты абстракциялау (аннотациялар);
- мәтіндерді автоматты түрде аударуда түсініктеме беру;
- іздеу жүйелеріндегі іздеу аймағын шектеу;
- жаңалықтар мен іздеу нәтижелерін жекелендіру;
- мәтіннің кодталуы мен тілін анықтау;
- мәтіннің эмоционалды бояуын, яғни тональділігін анықтау.

#### **1.4 Қолданыстағы әлеуметтік желілердің жаңа мүмкіндіктері мен кемшіліктері**

Әлеуметтік желі - бұл өзіңіз туралы ақпаратты орналастыруға және басқа пайдаланушылармен, фотосуреттермен, хабарламалармен және әртүрлі файлдармен ақпарат алмасуға болатын, сонымен қоса бағалау мен пікір білдіре алатын Интернет сайты.

Әлеуметтік желілердің мүмкіншіліктері:

- достармен қашықтықтан сөйлесу мүмкіндігі;
- жоғалған контактілер мен жаңа таныстарды іздеу;
- әлеуметтік желілерді өзін-өзі дамыту, шет тілдерін үйрену, қызықты кітаптар оқу, музыка тыңдау құралы ретінде пайдалануға болады. Қызығушылықтар топтары әлеуметтік желілерде құрылады, онда сізді қызықтыратын ақпаратты таба аласыз, мысалы, фитнес сабақтарының немесе гитара сабақтарының бейнелері;
- әлеуметтік желілер оқу процесіне көмектесе алады. Олардың көмегімен сіз дәріс жазбаларын, зертханалық тапсырмаларды және басқа да пайдалы ақпаратты бөлісе аласыз. Сондай-ақ белгілі бір тақырыптың қауымдастығына қосылып, тарих мәселелерін егжей-тегжейлі зерттеуге немесе шет тілінде білім алуға болады. Ол үшін қажетті әдебиеттерге, фото және бейнематериалдарға сілтемелер бар, топтың басқа мүшелерімен проблемалық мәселелерді талқылауға болады;
- әлеуметтік желілерде сіз достарыңызбен ғана емес, бейтаныс адамдармен де сөйлесе аласыз.

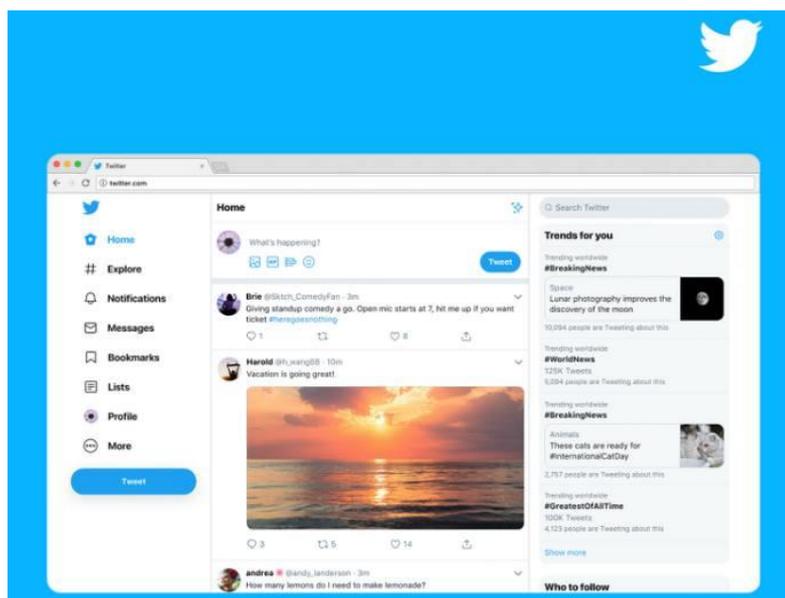
Әлеуметтік желілердің кемшіліктері:

- желіде сіз қысқа нөмірге SMS төлегеннен кейін ұсынылатын әртүрлі қызметтерді кездестіре аласыз. Көбінесе бұл алаяқтық. Сондықтан бейтаныс адамдардан файлдарды қабылдауға, күмәнді сілтемелерді ашуға болмайды, өйткені осының нәтижесінде компьютеріңізді вирустармен жұқтыруыңыз мүмкін;
- әлеуметтік желілерде пайдасыз ақпараттардың жеткілікті болуына байланысты интернетке кететін уақыт бірнеше есеге артады. Мұның бәрі әл-ауқатқа, физикалық және психологиялық денсаулыққа теріс әсер етеді. Ақпараттың көлемі жүйке жүйесіне теріс әсер етеді, ашуланшақтық пен агрессивтілік пайда болады;
- интернет пен әлеуметтік желілерге тәуелділік адамның гормоналды фонын өзгертуі мүмкін. Уақыт өте келе шынайы қарым-қатынас дағдылары жоғалады. Барлық мәселелерді желіде шешу адамды асоциалды тұлғаға айналдырады;
- әлеуметтік желілер сөйлеу мен емлеге теріс әсер етеді. Нашар сөздік, сөйлеу және грамматикалық қателер – барлығы желідегі қарым-қатынастың нәтижесі;
- профильді жиі тексеруден туындайтын тәуелділік тағы бір кемшілік болып табылады. Статистикаға сәйкес, бұл жасөспірімдер арасында жиі кездеседі және психологиялық бұзылыс болып табылады.

Ендеше, қолданыстағы танымал әлеуметтік желілердің жаңа мүмкіндіктері мен кемшіліктерін қарап шығайық.

### *Twitter.*

Twitter ең жылдам дамып келе жатқан әлеуметтік желілердің біріне айналды. Жалпыға ортақ қысқа хабарламаларды жылдам бөлісу мүмкіндігіне баса назар аударылады. Твиттерді бүгінде қарапайым адамдар ғана емес, сонымен қатар танымал адамдар, бизнесмендер, тіпті президенттер де пайдаланады.



**1.2-сурет – Twitter әлеуметтік желісі**

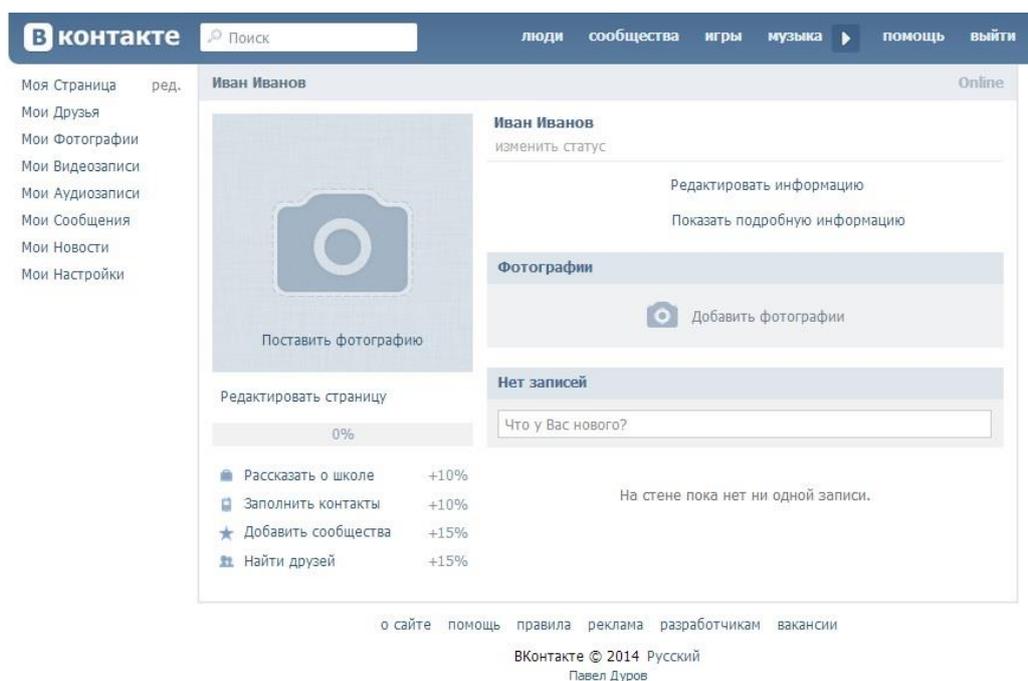
Twitter пайдаланушыларға төмендегідей мүмкіндіктер береді:

- әлемдегі оқиғалар туралы әртүрлі адамдардың көзқарасы бойынша ақпарат алуға болады;
- сүйікті жұлдыздарыңыздың өмірі мен жұмысын бақылауға болады;
- соңғы жаңалықтарды жылдам оқуға болады;
- демалуға және көңіл көтеруге;
- кез келген басқа қызмет сияқты, Twitter-де де кемшіліктер бар. Оларға:
- әкімшіліктің саяси астарлы шешімдері;
- уақыт алады. Басқа әлеуметтік желілер сияқты, Twitter бос уақытты өлтіреді;
- жарнамалық твиттер. Көбінесе пайдаланушылар қажетсіз жарнамаларға тап болады;
- көптеген боттар. Көптеген тіркелгілер тек бір мақсат үшін жасалады - жазылушылар санын «жою».

Кәдімгі әлеуметтік желілерге балама қолданып көріңіз. Facebook, Vkontakte және басқа да әлеуметтік желілер кейбір аспектілері бойынша ерекшеленсе де, көп жағынан бір-біріне ұқсас. Адамдар ол жерге тек сөйлесу үшін ғана емес, сонымен қатар музыка тыңдау, фильм көру, ойын ойнау және т.б. үшін келеді. Twitter, керісінше, пайдаланушыларға өздерінің микроблогтарын жүргізуге мүмкіндік береді. Адамдар арасындағы негізгі қарым-қатынас қысқа мәтіндік басылымдар төңірегінде жүзеге асады. Кейбіреулер үшін бұл қарым-қатынас тәсілі ыңғайлырақ көрінеді.

*ВКонтакте.*

ВКонтакте – Mail.Ru Group компаниясына тиесілі әлеуметтік желі. ВКонтакте пайдаланушыларға бір-біріне хабарламалар жіберуге, өз беттері мен қауымдастықтарын құруға, суреттермен, аудио және бейне жазбалармен алмасуға, ақша аударуға, браузер ойындарын ойнауға мүмкіндік береді. Ол сонымен қатар шағын қолданбаларды пайдалана отырып, бизнесті алға жылжытуға және күнделікті тапсырмаларды шешуге арналған платформа ретінде көрсетеді.



### 1.3-сурет – Вконтакте әлеуметтік желісі

Бұл әлеуметтік желінің көптеген артықшылықтары бар:

- 1) Ескі байланыстарды сақтау және жаңа танысу мүмкіндігі. Пайдаланушылар мыналарға қол жеткізе алады: байланыс, ақпарат алмасу, медиа файлдар, қауымдастықтар мен қызығушылық клубтарын құру. Адамдар әртүрлі жарты шарларда болса да, нақты уақыт режимінде сөйлесе алады. Хабарламалар мен қоңыраулар жіберу тегін.
- 2) Аты, тегі, жасы және тұрғылықты жері бойынша адамдарды оңай іздеу. Барлық параметрлерді орнатқаннан кейін платформа көрсетілген деректерге сәйкес келетін тіркелгілерді таңдауды қамтамасыз етеді.
- 3) Барлық медиа файлдарға ашық қол жетімділік: ойындар, қолданбалар, музыка, бейнелер, кітаптар және фотосуреттер. Сіз оларды жеке хат алмасуда, пікірлерде, топтарда және чаттарда алмастыра аласыз. Әлеуметтік желіден компьютерге бейне және аудио файлдарды жүктеп алуға болады.
- 4) Қауымдастықты құру. Оны қысқа мерзімде ақылы да, тегін әдістермен де, көп уақыт пен ресурстарды жұмсамай-ақ алға жылжытуға болады.
- 5) Бизнес жобаларды немесе жарнама арқылы табыс әкелетін ақпараттық қоғамдастықтарды, сондай-ақ мейрамханалар, сауда нүктелері, кинотеатрлар және т.б. туралы интернет-дүкендер мен тақырыптық көпшілікке ақша табыңыз. Мұнда API қызметі негізінде қосымшаларды жасап, оларды жалпыға қолжетімді ете аласыз. Бұл жағдайда шот «дауыс» алады - ВКонтakte ішкі валютасы.

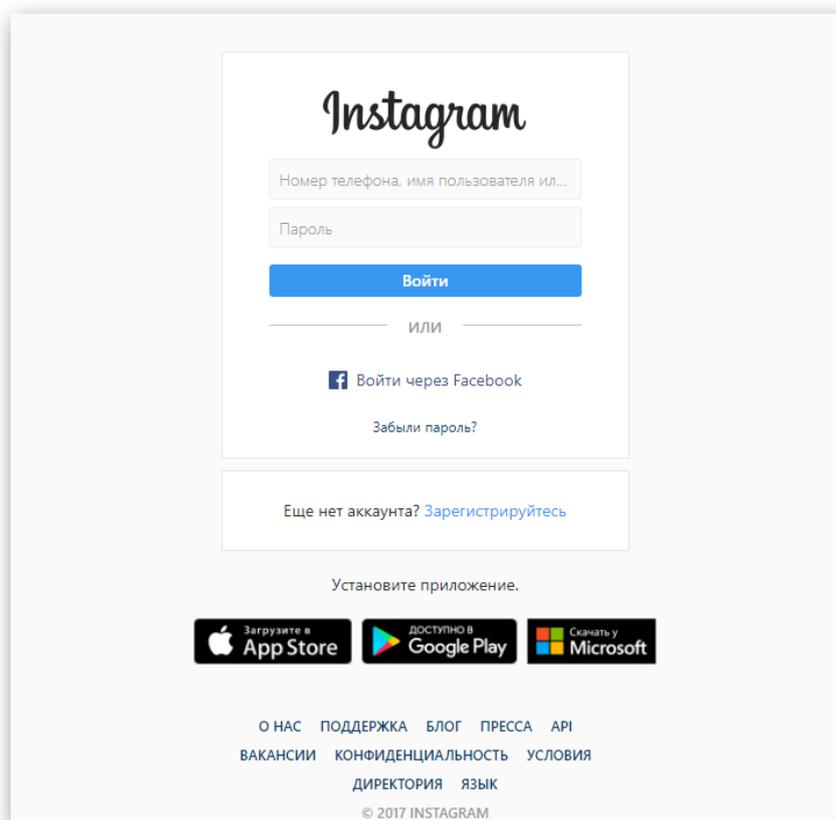
- 6) Мұнда орнатылған виджеттің арқасында топты сайтыңызға байланыстыру мүмкіндігі. Осылайша, келушілер бір түймені басу арқылы топқа жазыла алады.
- 7) Жақсы сайтты оңтайландыру және элементтердің жұмысын шамадан тыс жүктеу және баяулатудың болмауы.
- 8) Дизайн және техникалық бөлікті үнемі жаңарту және өзектілендіру.
- 9) Жарнаманың аз мөлшері, мазмұнды қабаттастыратын қалқымалы баннерлер жоқ. Жарнамалар мақсатты және таргетинг арқылы аудиторияның қажеттіліктеріне бейімделген.
- 10) Жеке хабарламалардағы жіберулер үшін жазылушылар базасын жинау мүмкіндігі. Ол үшін ВКонтакте арнайы қосымшалары қолданылады. Бұл бизнесті жеңілдетеді және көбірек адамдарға қол жеткізу арқылы сатылымды арттыруға мүмкіндік береді.

Бұл әлеуметтік желінің негізгі кемшіліктеріне мыналар жатады:

- 1) Жеке деректерді өзгерту қиындықтары. Сіз мобильді нұсқада қарауға өтініш беріп, жауапты күтуіңіз керек. Бұл деректерді өңдеу және тексеру қолмен жүзеге асырылады.
- 2) Бүркеншік есімді пайдалану мүмкін емес, сондықтан мұндай қажеттілік туындаса, бұрыннан бар есептік жазбаны жойып, жаңасын бастауға тура келеді.
- 3) Ересектерге арналған мазмұнның болуы.; Жеке хабарламаларға вирустар мен спам жіберу. Желі қолданушылары интрузивті жарнамаларға шағымданады. Мұндай аккаунттарды қадағалап, бұғаттау үшін «Шағым беру» түймесін басу керек.
- 4) Құпиялылықтың болмауы. Арнайы параметрлерді орнатпасаңыз, барлығы құжаттары бар бөлімге қол жеткізе алады.
- 5) Нашар қолдау қызметі. Мәселені шешу туралы сұрау тек жазбаша түрде жіберіледі және мәселенің шешілетініне кепілдік бермейді. Қызмет баяу, жауаптар әрқашан келе бермейді.

### *Instagram.*

Instagram - фотосуреттер мен бейнелерді бөлісуге арналған американдық әлеуметтік желі. Қолданба пайдаланушыларға сүзгілер арқылы өңдеуге және хэштегтермен және геотегтермен ұйымдастыруға болатын медиа файлдарды жүктеп салуға мүмкіндік береді. Хабарларды жалпыға ортақ немесе алдын ала бекітілген жазылушылармен бөлісуге болады. Пайдаланушылар басқа пайдаланушылардың мазмұнын тегтер мен орын бойынша шолып, тренді мазмұнды көре алады. Пайдаланушылар фотосуреттерді ұнатып, басқа пайдаланушыларға мазмұндарын жеке арнаға қосу үшін бақылай алады.



#### 1.4-сурет – «Instagram әлеуметтік желісі»

Инстаграмның артықшылығы мен кемшіліктері

Instagram өзінің жағымды қасиеттеріне байланысты өте танымал, бірақ, әрине, оның белгілі бір кемшіліктері де бар. Бұл әлеуметтік желінің басқалардан айырмашылығы неде және оның қандай артықшылықтары бар екенін көруге тырысайық.

Инстаграмның артықшылығы:

- суреттерді интернетке жылдам жүктей аласыз. Шын мәнінде, сіз өз өміріңізді желіде тарата аласыз. Жаңа аяқ киім, қызықты мүсінше, күлкілі жазу немесе әдемі тағам - олардың барлығы жаһандық желіде бірден пайда болады;
- қызығушылықтар бойынша фотосуреттерді қарапайым іздеу. Ол үшін тегтер (хэштегтер) қолданылады. Фотосуреттердегі қалалардың атаулары немесе объектілердің белгілері бар хэштегтер бар;
- басқа пайдаланушылардың есептік жазбаларына қарапайым жазылу. Мұнда «достар» жоқ, сіз жай ғана адамның рұқсатынсыз оның суретін көре аласыз. Мұны жұлдыздардың жанкүйерлері мен сайтта көп болатын БАҚ өкілдері пайдаланады.

Инстаграмның кемшіліктері. Кез келген заманауи әлеуметтік желі сияқты, Instagram-дың да кемшіліктері бар. Олар:

- құпиялылық параметрлері бойынша төмен мүмкіндіктер. Фотосуретті «тізім астында» жабу мүмкін емес, тек профиль толтырылған;

- фотосуреттерді жүктеуге арналған толыққанды сайттың болмауы. Бұл суреттерді бөлісуге қарсы болмайтын, бірақ оларды ұялы телефон арқылы жүктей алмайтын аудиторияның бөлігін қысқартады;
- фотосуреттерді жүктеп алу мүмкін емес. Дәлірек айтқанда, мүмкіндік бар, бірақ үшінші тарап қызметтерін пайдалану арқылы ғана тіркелгіңізге кіруге рұқсат беру керек;
- техникалық көмек.

## 2 Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелерге талдау

### 2.1 Әлеуметтік желілердің функцияларын анализдеу

*Әлеуметтік желілердің деректерін талдау.* Әлеуметтік желі ұғымын әлеуметтанушылар өткен ғасырдың 20-жылдарында әртүрлі қауымдастықтар мүшелері арасындағы қарым-қатынасты зерттеу үшін қолданған. Психолог және психотерапевт Джейкоб Морено социограммаларды ұсынды, онда индивидтер нүктелер түрінде, ал олардың арасындағы байланыстар сызықтар түрінде бейнеленген. Адамдар арасындағы қарым-қатынастар мен қарым-қатынастарды зерттеу үшін графикалық теория аппаратын пайдалану идеясын әлеуметтану, психология, антропология, саясаттану, экономика саласындағы мамандар алды - Әлеуметтік желіні талдау бағыты осылай қалыптасты, графиктер мен желілер түрінде үлгіленген әлеуметтік қатынастардың құрылымдық қасиеттерін зерттейтін. Мұндай зерттеудің маңызды, бірақ өте көп уақытты қажет ететін кезеңі баспа көздерінен алынған әртүрлі деректерге, қосымша сауалнамалар мен сауалнамаларға негізделген модельді құру болды.

Заманауи әлеуметтік желілер сұрақтың тұжырымын айтарлықтай өзгертті - бүгінгі күні зерттеушілерде зерттеу үшін «тегін» ресурс бар, ал әлеуметтік онлайн қызметтерінің жылдам таралуы және Big Data технологияларының дамуы әлеуметтік желілердегі ақпаратты пайдалануға қызығушылықты тудырды. әртүрлі салаларда. Құрылымдық және мазмұндық деректерді бөлісу әлеуметтік түрде әлеуметтік желілерді бизнес тапсырмаларының кең ауқымын шешу үшін пайдалануға мүмкіндік береді: алаяқтықпен күресу, брендті басқару, тауарлар мен қызметтерді жарнамалау, жаңа тарату арналарын құру және т.б.

Әлеуметтік желілерде, форумдарда, жаңалықтар мен ойын-сауық порталдарында және блогтарда көптеген құнды материалдар бар, олардан адамдар мен компаниялардың артықшылықтары мен сипаттамалары туралы ақпаратты алуға болады. Мұны істеу үшін, ең алдымен, әрбір дереккөздегі клиентті анықтау қажет, оны барлық ресурстар жасауға мүмкіндік бермейді - олардың көпшілігінде адамдар тіркелмейді немесе сәйкестендіру деректері жеткіліксіз. Клиентті анықтау үшін деректер жеткілікті болған жағдайда да ол туралы пайдалы қосымша ақпарат болмауы мүмкін. Осыған байланысты әлеуметтік желілер клиентті анықтауға арналған ақпаратты да, қалауы, отбасылық жағдайы, білімі, әлеуметтік ортасы және т.б. туралы қосымша мәліметтерді қамтитын ең қолайлы көз болып табылады.

Тұтастай алғанда, тұтынушы профилін байыту міндеті келесідей. Компания өз тұтынушылары туралы негізгі деректерді (аты, тегі, туған күні, қаласы) ұсынады және олардың негізінде қосымша ақпаратты табу қажет: қызығушылықтар шеңбері, әлеуметтік мәртебе, кәсіби қызмет саласы, музыкалық қалаулар және т.б. .Бұл мәселені шешу үшін әлеуметтік желілерден

тұтынушы деректерін жинап, оны анықтау, деректерді байыту және әрбір тұтынушы үшін бір профильді қалыптастыру қажет.

Деректерді жинаудың ең оңай жолы - көптеген көздерден деректерді жинайтын және үнемі жаңартып отыратын мамандандырылған компаниялардың қызметтерін пайдалану. Мұндағы басты артықшылық - үлкен клиенттік базамен және әртүрлі әлеуметтік желілерді пайдаланумен қажет ақпаратты алу жылдамдығы. Кемшілігі - деректерді жаңартуға ақылы жазылу.

Келесі әдіс - барлық танымал әлеуметтік желілер ұсынатын API интерфейстерін пайдалану. Әртүрлі желілер үшін API интерфейстері қолжетімді деректер жиынтығымен, сұраулар санына шектеулермен және интерфейстерге қол жеткізу құнымен ерекшеленеді. Мысалы, егер Vkontakte желісінің бағдарламалау интерфейсін пайдалансаңыз, сіз пайдаланушы туралы толық ақпаратты ала аласыз, онда Facebook пайдаланушы туралы «нөлге» дерлік ақпаратты қайтаратын API ұсынады. Бұл әдістің кемшіліктері бір уақыттағы сұраулар санына және қолданба уақыт бірлігінде жасай алатын қоңыраулар санына шектеуді қамтиды. Сонымен қатар, API-дегі өзгерістерді үнемі қадағалап отыру және деректерді жинау қосымшасын жаңарту қажет, ал кейбір әлеуметтік желілер маңызды деректерді тек ақылы түрде береді. Әдістің артықшылықтары құрылымдық пішінде (JSON немесе XML) бір клиент туралы деректерді алу мүмкіндігі, сондай-ақ API қоңырауларын жеке қолданбаңызға біріктірудің қарапайымдылығы болып табылады.

Тағы бір әдіс - әлеуметтік желінің веб-беттерін қолмен талдау, сондай-ақ кейінгі талдау арқылы деректерді жинау үшін дайын тексеріп шығушыларды пайдалану. Бұл жағдайда барлық ашық деректерге қол жетімділік бар және оларды жинау жылдамдығына ешқандай шектеулер жоқ. Кемшіліктер іске асырудың күрделілігін қамтиды - әрбір әлеуметтік желінің веб-парағы бірегей, сондықтан әр уақытта жеке талдау ережелерін әзірлеуге тура келеді. Кемшіліктер - бұл қолдаудың күрделілігі және үлкен есептеу ресурстарының қажеттілігі, алайда бұл процесс жақсы параллельді.

Тұтынушыны сәйкестендіру - әлеуметтік желілерде белгілі бір тұтынушыны білдіретін барлық профильдерді анықтау. Төлқұжат деректері іздеу үшін бастапқы деректер ретінде қызмет ете алады, бірақ қосымша ақпарат пайдалы болады. Шеңберді тарылту және адамды табуға көмектесу үшін ол жұмыс істейтін компанияның аты, телефон нөмірі, пошта мекенжайы, оқу орны және достарының тізімі сияқты ақпарат көмектеседі.

Анықтаудың ең оңай жолы - клиенттің барлық белгілі сипаттамаларының дәл сәйкестігін іздеу, алайда әлеуметтік желілердегі сәйкес сипаттамалар белгілі бір дәрежеде ғана сенімді екенін ескеру қажет - олар жоқ болуы мүмкін, әдейі болуы мүмкін. жалған немесе әртүрлі емле жазуға рұқсат етіңіз. Сондықтан, сәйкестендіруді жүзеге асырмас бұрын деректерді тазарту және қалыпқа келтіру, сондай-ақ профильде көрсетілген параметрлердің дұрыстығын тексеру қажет - мысалы, пайдаланушының қаласы оның жазылымдарын, жазбаларын талдау негізінде анықталуы мүмкін күйлерді жатқызуға болады.

Кейбір параметрлерді пайдаланушының немесе оның достарының профилін талдау арқылы қалпына келтіруге болады. Мысалы, әйелдер көбінесе туған жылын көрсетпейді, ал университетті немесе мектепті бітірген жылы бар.

Сәйкестендіруде қолданылатын әрбір сипаттаманың белгілі бір салмағы бар - барлық салмақтардың қосындысы, егер барлық параметрлер сәйкес келсе, біреуге тең болуы керек. Сонымен, тегі, аты және жынысы сәйкестендіру кезінде маңызды параметрлердің бірі болып табылады және егер бұл деректер қате болса, жоғары ықтималдықпен бұл пайдаланушыны анықтау мүмкін болмайды. Екінші орында туған күні мен айы. Бұл деректерді қалпына келтіруге болады, бірақ онсыз сәтті сәйкестендіру мүмкіндігі де өте төмен. Ең аз салмақ туған қаласы мен жылы болып табылады. Дегенмен, дәл осы параметрлер басқа деректер негізінде қалпына келтіруге жақсы мүмкіндік береді.

Желі қолданушылары өз профилінде нақты көрсететін деректерден басқа, жазбаларды, келесі топтар мен фотоларды талдау арқылы көп нәрсені білуге болады. Сонымен қатар, бұл құрылымдалмаған ақпараттан алуға болатын қосымша фактілер қызығушылық тудырады. Мысалы, қабырғадағы жазбалардың көпшілігі кино тәжірибелері туралы болса, пайдаланушының фильмдерге қызығатыны анық.

Мәтінді автоматты талдау лингвистикалық технологияларсыз мүмкін емес. Сонымен қатар, статистикалық әдістер, машиналық оқыту технологиялары және деректерді терең талдау (деректерді өндіру) көптеген мәселелерді шешу үшін де пайдалы. Статистикалық зерттеулер мен табиғи тілмен жұмыс әдетте қандай да бір дәлсіздікпен байланысты - статистикада әрқашан толық орындала бермейтін белгілі бір болжамдар, эвристикалық жорамалдар болады, ал табиғи тілде мәлімдемелер мен қорытындыларды екі жақты түсіндіру мүмкіндігі әрқашан болады. Лингвистикалық және статистикалық тәсілдердің дұрыс үйлесуі нәтиженің сапасын және оның сенімділік деңгейін арттырады. Мәтіндік деректерді байытудағы әртүрлі әдістердің мүмкін болатын корреляциясын көрсету үшін бірнеше мысалдарды қарастырайық.

Қолданушының футболға қызығушылығы бар-жоғын анықтау керек делік. Оның қабырғасындағы мәтіндерде сәйкес терминдердің қаншалықты жиі кездесетінін және олардың пайда болуының белгілі бір деңгейіне жеткенде белгілі бір қорытындылар жасауға болатынын анықтайық. Бұл байыту әдісі үшін белгілі бір пән саласына арналған сөздіктерден немесе тезаурилерден алуға болатын терминологияны білу қажет. Сонымен қатар, сіз сондай-ақ қолдану санын дұрыс есептей білуіңіз керек - бір сөздің әртүрлі формаларын түсіну үшін. Осылайша, бұл мысал үшін тек тілдік құралдар жеткілікті.

Екінші мысал лингвистикалық өңдеуден басқа, машиналық оқыту әдістері қажет болған жағдайға қатысты. Пайдаланушының толық туған күні жоқ және ол жазған мәтіндер негізінде жас тобын анықтау керек делік. Ең алдымен, жасы белгілі пайдаланушылардың мәтіндер жинағы қалыптасады. Содан кейін бұл жиынтық үшін машиналық оқыту алгоритмдерін қолдана отырып, әрбір жас

тобына арналған мәтіндердің ерекшеліктері ашылады және еркін мәтін үшін оның авторының жасын бағалауға мүмкіндік беретін қандай да бір формальды модель қалыптасады. Машиналық оқыту алгоритмдері әдетте құрылымдық деректерге арналған, сондықтан оларды қолданбас бұрын мәтіндер оларда кездесетін сөздер жиынымен немесе осы мәтіндерді сипаттайтын тақырыптар жиынтығымен ауыстырылады. Ол үшін лингвистикалық алгоритмдер мәнді сөздерді бөлектеу, оларды нормалау, мәтіннің лексикалық профилін құру, тақырыптарды анықтау және т.б.

Бір клиент сәйкестендіру деректерін ескере отырып, сәйкестендіру сенімділігінің жеткілікті жоғары деңгейімен әлеуметтік желілерде көптеген әртүрлі пайдаланушыларға ие болуы мүмкін.

Бұл жағдайда бірнеше пайдаланушылардың деректерін бір клиенттік профильге «біріктіру» міндеті туындайды. Деректерді қалай дәл біріктіру нақты тапсырмаға байланысты - мысалы, қызығушылықтардың жалпы тізімін қалыптастыру үшін әрбір пайдаланушы үшін бар қызығушылықтарды ғана таңдауға болады. Немесе берілген жалға алушының барлық пайдаланушыларының мүдделерін біріктіріп, мүдделердің кеңейтілген жинағын пайдалануға болады.

Nadoor, RCO лингвистикалық өңдеу құралдары және R тілі негізінде Force әзірлеген ForSMedia платформасы жоғарыда аталған әдістер мен тәсілдердің барлығын жүзеге асырады. Шешімнің маңызды ерекшелігі - автоматты режимде тұтынушы профильдерінің үлкен саны үшін деректерді жаппай байыту мүмкіндігі. Платформаны пайдаланушы серверлерінде орналастыруға немесе бұлттық қызмет ретінде жеткізуге болады.

Әлеуметтік желілер кез келген компанияның тұтынушылары туралы қосымша деректердің жаңа пайдалы көзі ретінде қызмет етеді. Бұл дереккөзді пайдалану оңай емес және осы жолда туындайтын мәселелер арнайы технологиялар мен құралдарды қажет етеді. Nadoor және басқа Big Data технологияларына негізделген ForSMedia жүйесі тұтынушы профильдерін жаппай өңдеу режимінде әлеуметтік желілерде нақты көрсетілген деректермен ғана емес, сонымен қатар көптеген хабарлама мәтіндерінде, жазбаларда, жазылу топтарында жасырын түрде көрсетілген ақпаратпен автоматты түрде байытуға мүмкіндік береді.

Адамдар топтарындағы әлеуметтік байланыстарды зерттеу 19 ғасырдың соңы – 20 ғасырдың басында қарастырыла бастады. Осылайша, Эмиль Дюркгейм барлық әлеуметтік құбылыстарды жеке тұлғалардың қасиеттерімен түсіндіруге болмайтынын алға тартты. Георг Зиммель еркін байланысқан желілік әлеуметтік құрылымдарды қарастыра отырып, өз еңбектерінде әлеуметтік желілерді талдаудың негізін қалады. Осы кездегі әлеуметтік философтар «қарым-қатынас торы» терминін қолданды.

1930 жылдары Дж.Морено тұлғааралық және топаралық қатынастарға арналған социометрия бойынша бірқатар еңбектер жариялады. Мореноның ғылыми еңбектерінің негізгі жаңалығы социограммалар болып саналады - шағын әлеуметтік топтағы тұлғааралық қатынастар құрылымының схемалық

көрінісі. Оның эксперименттерінің бірінде қыздарға арналған балалар үйінің тәрбиеленушілерін бір-біріне жанашырлары ғана ортақ коттеджде болатындай етіп бөлек коттеждерге орналастыру, ал бірін-бірі ұнатпайтын қыздардың бірге тұруына тыйым салу міндеті қойылған. Зерттелушілерден кімді ұнататынын және кіммен жұмыс істегісі келетінін немесе бос уақытын бірге өткізгісі келетінін сұрады. Нәтижелер матрицалар қатары түрінде ұсынылды, мұнда топтың әрбір мүшесін топтың басқа мүшелері бағалады. Әрі қарай, жеке және топтық индекстер есептелді, социограммалар графиктер түрінде құрастырылды, оларда топ мүшелеріне белгілі бір позициялар тағайындалды, таңдаулар (оң және теріс) көрсеткілермен белгіленді - осылайша топтың «құрылымы» көрсетілген. топтағы қалаулар көрсетілді. Дж.Морено өз тәжірибелерімен шын мәнінде әлеуметтік желілерді аналитикалық зерттеудің негізін қалады.

Кейінірек Алекс Бавлас (1948) мен Гарольд Левитт (1951, 1963) теорияның дамуында тағы бір маңызды қадам жасады: олар желіні жеке тұлғалардың емес, позициялардың жиынтығы деп түсінуді ұсынды. Тәжірибе барысында олар алған позициялар арасындағы қарым-қатынастардың соңғы үлгісі құрылымның негізі немесе түрі сияқты көрінді. А.Бейвлас еңбектерінде алғаш рет орталық (коммуникациялар орталық позиция арқылы жүзеге асырылған кезде белгілі бір міндеттер жақсырақ және жылдам орындалатын) туралы айтылып, позициялар арасындағы байланыстар туралы идея ұсынылды. ресурстар ағындары болып табылады.

XX ғасырдың екінші жартысында. әлеуметтік желілердің жүйелік талдауы Р.Соломонов, С.Берковиц, С.Боргетти, Р.Берт, К.Карли, К.Фауст, Д.Нок, П.Марсен, Н.С. сияқты зерттеушілердің еңбектерінде айтарлықтай кеңейді. Маллинс, А.Рапопорт, С.Вассерман, Б.Веллман, Д.Уайт, У.Харрисон және т.б. 1950 жылдары Ағылшын антропологтары Дж.Барнс пен Э.Ботттың еңбектері әлеуметтік желілерге жататындығымен байланысты күрделі әлеуметтік құбылыстарға назар аударды. «Әлеуметтік желі» терминін 1954 жылы социолог Джеймс Барнс енгізген [20]. Сол кездегі көптеген антропологиялық еңбектердің ішінен С.Найдельдің «Әлеуметтік құрылым теориясы» (1957) еңбегін ерекше атап өту керек, онда «құрылым» мен «функцияның» нақты бөлінуі жасалған, бұл қазіргі заманғы желілік талдау принциптері. Найдель үлкен популяциялардағы «құрылымдарды» зерттегендіктен (шағын эксперименттік топтарға қарағанда), оның жұмысы психологияда да, социологияда да сұранысқа ие болды. Ғалым өз еңбегінде: «Біз қоғамның құрылымын нақты популяция және мінез-құлық үлгілері немесе субъектілердің бірлескен және өзара бағытталған рөлдерін орындау арқылы алған қарым-қатынас желілері (немесе жүйелері) арқылы анықтаймыз» деп жазды.

1959–1968 жж Венгриялық математиктер Пол Эрдос пен Альфред Реньи әлеуметтік желілердің қалыптасу принциптерін сипаттайтын мақалалар сериясын жариялады. Ғалымдар алғаш рет әлеуметтік желілерді құру принципін көрсету үшін математикалық теорияны қолданды. Ұсынылған

кездейсоқ графиктер теориясы құрылыстың айқын принциптері жоқ күрделі желілерді сипаттауға мүмкіндік берді [21]. График төбелерінің кездейсоқ біркелкі қосылуы кезінде  $P(k)$  ( $k$  – төбелерге кіретін жиектер саны) үлестірімі биномдық, ал үлкен график өлшемі шегінде Пуассон болғандықтан, мұндай желілер де аталады. Пуассонның кездейсоқ желілері. Алайда, 21 ғасырдың басында Эрдос-Реньи моделі әлеуметтік интернет желілерінің нақты графиктерімен жақсы сәйкес келмейтіні белгілі болды.

1969 жылы американдық психологтар Стэнли Милграмм мен Джеффри Траверс алты қол алысу теориясын ұсынды, оған сәйкес жер бетіндегі кез келген екі адамды орта есеппен тек бес деңгейлі өзара танысу және сәйкесінше алты байланыс деңгейі ("деп аталатын") ажыратады. жақын дүние құбылысы»). Американың екі қаласында өткізілген эксперимент жобасы бойынша Омаха (Небраска) тұрғындарына 300 конверт таратылды, олар Бостонда тұратын белгілі бір адамға беріледі. Конверттерді тек таныстары мен туыстары арқылы беруге рұқсат етілді. Нәтижесінде Милграмм есептеулерді жүргізгеннен кейін, орта есеппен әрбір келген конверт бес адамнан өткенін анықтады. Сипатталған тәжірибе әртүрлі вариацияларда бірнеше рет қайталанды. 1998 жылы Колумбия университетінің ғалымдары электронды пошта арқылы зерттеуді ұқсас схемамен растады. Интернетте алты рет басу арқылы бір сайттан екіншісіне өтуге болатыны белгілі болды. 2008 жылы Microsoft екі жылдық зерттеу нәтижелерін жариялады, онда қызметтің 240 миллион пайдаланушысының әрқайсысы орта есеппен 6,6 «қол алысу» арқылы екіншісіне «жүре» алады. Милан университеті мен Facebook әлеуметтік желісі де Facebook желісінің әлеуметтік графикалық деректеріне сүйене отырып, алты қол алысу теориясын эксперименталды түрде зерттеді. Зерттеу нәтижесінде кез келген екі Facebook пайдаланушысын 4,74 байланыс деңгейі ажырататыны белгілі болды.

«ВКонтакте» әлеуметтік желісінде «Достар тізбегі – алты қол алысу теориясы» (<https://vk.com/fchain>) арнайы қосымшасы желі қолданушылары арасындағы таныстарды табуға мүмкіндік береді. Вконтакте аудиториясы бұрынғы ТМД елдерінде ең көп болғандықтан, Украина (немесе Ресей) аумағынан гипотетикалық бейтаныс адамға кірген адамдар 3-4 адамның нәтижесін алады. Осылайша, әлеуметтік интернет желілерінде алты қол алысу теориясын тәжірибе жүзінде растау әрқашан мүмкін бола бермейді. Қатені де былай түсіндіруге болады. Жаңа пайдаланушы тіркеліп, жеке деректерді енгізген кезде, әлеуметтік желі ол біле алатын адамдар жинағын (мысалы, электрондық пошта контактілерін талдау арқылы) немесе ортақ мүдделері, ұқсас жұмысы және т.б. бойынша адамдарды таңдауды ұсынады. ол өзі жиі араласатын адамдарды таңдай алады. Осылайша, мұғалім өз әріптестерін таба алады, оқушылары мен студенттері пікірлес адамдармен (бір жастағы, қызығушылықтары ұқсас, ұқсас университеттер және т.б.) сөйлесе алады [23].

1998 жылы Стивен Строгац пен Дункан Уоттс тұрақты, периодтық торлар мен Пуассон кездейсоқ желілерінің қасиеттерінің суперпозициясына негізделген «шағын әлем» желілерінің математикалық моделін ұсынды [24]. Зерттеушілер көрсеткендей, «шағын әлем» қасиеттеріне бағынатын желілерде

кездейсоқ қосылыстардың аз санының қосылуы желінің диаметрін азайтады, яғни. оның түйіндері арасындағы ең қысқа жолға дейінгі ең ұзын жол.

1970 жылдары Американдық әлеуметтанушы Марк Грановеттер [25] әлеуметтік желілерде «әлсіз» байланыстар (мысалы, көршілеріміз, таныстарымыз, таныстарымыздың таныстары, жұмыстағы ресми байланыстар) маңыздырақ екенін анықтады.

«Бұл ақпараттың «әлсіз» байланыстар арқылы тезірек және кеңірек таралуымен түсіндіріледі. Оның пікірінше, «әлсіз» байланыстар «пайдаланушыларға қоғамдастықпен өзара әрекеттесу және өзара әрекеттесу мүмкіндігін беру үшін маңызды, ал «күшті» байланыстар жергілікті байланыс жасайды». Жұмыспен қамту мысалында ол жұмыс табу және өмірдегі басқа да қажеттіліктер тұрғысынан біз онша білмейтін адамдармен байланыс неғұрлым пайдалы екенін көрсетті. Бұл «күшті» қосылымдар арқылы адамдар деректердің немесе ресурстардың шектеулі көлемін бөлісетіндігіне байланысты, «күшті» қосылымдар ақпараттық артық болып табылады, сондықтан олардың бір-біріне пайдалылығы азырақ. Осылайша, жұмыс табудың артықшылығы - бір-біріне сәйкес келмейтін байланыстардың үлкен желісі бар адам. М.Грановеттер бұл әсерді «әлсіз байланыстардың күші» деп атады.

«Әлсіз» байланыстардың күші туралы тезистің теориялық негіздемесін Рональд Берт өзінің «құрылымдық саңылаулар» теориясында ұсынды, ол коммуникациялық байланыссыз кеңістіктердің бар екендігін көрсетеді. Теория «әлсіз» байланыстардың беріктігі, әлеуметтік желідегі делдалдықтың маңыздылығы, эксклюзивті айырбас теориясы туралы идеяларға, сондай-ақ капиталдың үш формасын - қаржылық, адами және әлеуметтік айырмашылықтарға негізделген. Көптеген «әлсіз» байланыстардың болуы шынымен де адамға ақпарат алуда артықшылық береді.

Дункан Уоттс пен Стивен Строгач әлеуметтік желілер теориясын дамытты және көптеген басқа жаңалықтардың қатарында кластерлеу коэффициенті түсінігін енгізді - гетерогенді топтар арасындағы жақындық дәрежесін зерттеп, дамуға зор үлесін қосты.

Желілік технологиялардың дамуымен және жаңа байланыс құралдарының пайда болуымен 80–90 жж. «әлеуметтік желілер» терминін қолдану бірте-бірте әлеуметтанудан информатика саласына ауысуда. 1984 жылы АҚШ-тың Ұлттық ғылыми қоры кең университет аралық NSFNet желісін құрды. 1980 жылдардың екінші жартысынан бастап. заманауи әлеуметтік желілердің алғашқы прототиптері іске асырылуда - CompuServe, Prodigy және The Well. Жылдам хабар алмасу жүйелері - IRC және ICQ дүниеге келді. 1995 жылы Американдық Classmates порталы пайда болды. 2000 жылдардың басы әлеуметтік интернет желілері дәуірінің гүлденген кезеңі болып есептелді – 2003-2004 ж.ж. LinkedIn, MySpace және Facebook желілері іске қосылды. Ал егер LinkedIn іскерлік байланыстарды орнату және қолдау үшін құрылған болса, MySpace және Facebook иелері ең алдымен адамның өзін-өзі көрсету қажеттілігін қанағаттандыруға сенді. 2006 жылы Twitter жобасы іске қосылды, сонымен қатар ТМД елдеріндегі танымал әлеуметтік желі Вконтакте, бүгінгі таңда ең

үлкен бейне және аудио хостингі бар танымал әлеуметтік желілер болып табылады.

Желілік құрылымдарды зерттеуде соңғы жылдардағы маңызды нәтижелер теориялық физикада алынды. Атап айтқанда, 1999 жылы Альберт-Лассо Барабаши тұжырымдаған масштабсыз желілер теориясы пайда болды. Масштабсыз желілер - бұл төбелердің қосылымдарының санының таралуы экспоненциалды (Пуассон желілеріндегідей) заңмен емес, қуат заңымен сипатталатын график, сонымен қатар қуат заңы бойынша бөлінген объектілер жиі иерархиялық болады, ал желінің негізгі қасиеттері желінің өлшеміне байланысты емес болады. Бұл атау желілердің бұл түрі үшін арнайы ойлап табылған жоқ, бірақ критикалық құбылыстар теориясынан алынған, мұнда критикалық күйлердегі ауытқулар да қуат заңына бағынады, ал масштабсыз желілер теориясының өзі бір желі ретінде қарастырыла бастады. күрделі жүйелердің критикалық күйге шығу сценарийлері. Зерттеулер көрсеткендей, жанды және жансыз табиғаттағы желілердің көпшілігі (ақпараттық, экологиялық, генетикалық, адам миындағы функционалдық байланыстар, зат алмасу, әлеуметтік, технологиялық, сөздік, WWW құжаттары және т.б.) масштабсыз графиктермен жақсы модельденеді.

Жетекші әлеуметтанушы, Калифорния университетінің профессоры Мануэль Кастеллстің пікірінше, бүкіл қоғам тұтастай алғанда «желіге» айналуға, бұл коммуникацияның сапалық жаңа құралдарымен байланысты. «...Жаңа коммуникация жүйесі барған сайын әмбебап цифрлық тілде сөйлей отырып, бір мезгілде біздің мәдениетіміздегі сөздерді, дыбыстарды және бейнелерді шығару мен таратуды әлемдік ауқымда біріктіреді және оларды жеке адамдардың жеке талғамы мен көңіл-күйіне бейімдейді. Интерактивті компьютерлік желілер экспоненциалды түрде өсуде, коммуникацияның жаңа формалары мен арналарын жасайды, өмірді қалыптастырады және бір уақытта өмірмен қалыптасады» (Кастеллс, 2000). «Интернет галактикасы: Интернет, бизнес және қоғам туралы ойлар» деген еңбегінде зерттеуші былай деп атап көрсетеді: «жалпы алғанда, желілер адам әрекетін материалдандырудың өте ескі нысандары болып табылады, бірақ бүгінгі күні олар жаңа өмірге ие болды. олардың интернет арқылы қосылған ақпараттық желілерге айналуының нәтижесі. М.Кастеллс теориясының орталық концепциясы өзін ретінде ашатын «желілік құрылым» болып табылады.

«Бір-бірімен байланысқан түйіндер кешені», ал «әр түйіннің нақты мазмұны қарастырылатын нақты желі құрылымының сипатына байланысты». Қарым-қатынас қарым-қатынас жасау қабілеті арқылы анықталады. «Желілердің өзі ашық құрылымдар болып табылады, олар байланысқа қабілетті болса, жаңа түйіндерді қосу арқылы шексіз кеңейе алады...». Желіге қосу ақпаратқа қол жеткізуді қамтамасыз етеді. Сондай-ақ ғалымның еңбектерінде алғаш рет әлеуметтік процестер мен құбылыстарда желілік логиканың болуының теориялық негіздемесі берілген.

## 2.2 Әлеуметтік желілерде мәтіндердің классификациялық талдауы

Мәтіндік құжаттардың тональділігі бойынша жіктелуі тақырыбында Лукашевич Н.В., Браславский П.И., Котельников Е.В., Турней П., Мохаммад С, Киритченко С, Жу Х., Панг Б., Рубцова Ю.В. және т.б. сияқты ресейлік және шетелдік ғалымдардың еңбектерінде зерттелген.

Мысалы «Әлеуметтік желілердегі мәтіндерді тональдылық бойынша жіктеудің ақпараттық жүйесін құру әдістері мен алгоритмдері» тақырыбында Рубцова Ю.В. зерттеулер жүргізсе, Трусина Т.В. «Әлеуметтік желілердегі мәтіндерді талдау арқылы әлеуметтік тенденцияларды анықтау алгоритмі» тақырыбында зерттеу жасалынған. Жалпы мақала әлеуметтік желілердегі медиамәтіндерді зерттеу арқылы әлеуметтік тенденцияларды анықтауға, сондай-ақ осы мәтіндердің ерекшеліктері мен оларды талдау әдістерін қарастыруға арналған. Автор зерттеу барысында медиамәтіннің ерекшеліктерін ескеруге мүмкіндік беретін теориялық алгоритм әзірлеген.

Осы мақаланы жазу үшін материал ретінде автор әлеуметтік желілерде жарияланған мәтіндерді пайдаланды, яғни оларға қауымдастық жазбалары, пікірлер, микроблогтар жатады.

Сонымен қоса әлеуметтік желілердің медиа мәтіндерінің белгілі бір ерекшеліктері бар екенін атап өткен. Және де бүгінгі таңда әлеуметтік желілердің мәтіндік мазмұнын семантикалық талдау үшін жиілік-семантикалық алгоритм және Портер стемеріне негізделген алгоритм қолданылған.

Жиілік-семантикалық алгоритм мәтіндегі сөздердің кездесу жиілігін санауға негізделген, бұл оның тақырыбын анықтауға мүмкіндік береді. Мәтіндегі әрбір сөз мәтіннің тақырыбы немесе өзегі бола алмайтындықтан, сөздікке енгізілген зат есімдер ғана есепке алынады. Бұл әдісті әлеуметтік желілердің ерекшеліктеріне бейімдеу үшін сөздікке ауызекі лексиканы қосу қажет болып табылады.

Портер стемеріне негізделген алгоритм сөзден жалғаулар мен жұрнақтарды кесіп тастаудан тұрады, сонда қалған бөлігі сөздің барлық грамматикалық формаларына негіз болады. Штамминг аффикстер арқылы флексияны жүзеге асыратын тілдермен ғана жұмыс істей алады (мысалы, орыс және ағылшын). Бұл алгоритмнің басты артықшылығы - сөздіктің жоқтығы. Семантикалық талдау барысында сөз түбірлерінің массиві қалыптасады. Ең көп кездесетін бунақтар мәтіннің тақырыбы болады.

Сипатталған алгоритмдер әлеуметтік желілердегі ағымдағы тақырыптарды қадағалауға арналған қолданыстағы бағдарламалық жасақтамада бірнеше рет енгізілді (Brand Analytics, You Scan, Babkee). Сонымен бірге ғалымдар зерттеу мәтіндік деректер мен әлеуметтік графиктердің құрылымын біріктіріп талдауды қажет етеді деп есептейді. Әлеуметтік желілердің құрылымын кластерлеуге байланысты ашуға болады, яғни. барлық пайдаланушыларды әлеуметтік байланыстары негізінде ішкі топтарға бөлу. Бұл

кластер ішінде тренд аудиториясына тән демографиялық факторларды анықтауға мүмкіндік береді.

Төменде жалпыға қолжетімді ғылыми-практикалық әзірлемелер негізінде автор құрастырған әлеуметтік тенденцияларды анықтау мақсатында әлеуметтік желі мәтіндерін талдаудың жалпы алгоритмі берілген.

- қысқа мерзімді тенденцияларды анықтау;
- әлеуметтік желі кластерлерін бөлу;
- пікір көшбасшыларын табу және оларға басымдық беру;
- сәйкес мәтіндерді табу;
- спамды сүзу;
- метасезімдерді анықтау үшін басым лексемалар қолданылатын контексттерді зерттеу;
- трендтерді картаға түсіру, олардың кластерленуі;
- анықталған тенденцияларды барлармен салыстыру;
- ұзақ мерзімді тенденцияларды анықтау.

### **2.3 Талдау деректерін алу үшін ВКонтакте әлеуметтік желісінің API интерфейсін пайдалану**

API - сайттар, бағдарламалар және қолданбалар арасында ақпарат алмасу тәсілі. Сондай-ақ, бұл мұндай алмасу жүзеге асырылатын ережелер жиынтығы деп айтуға болады.

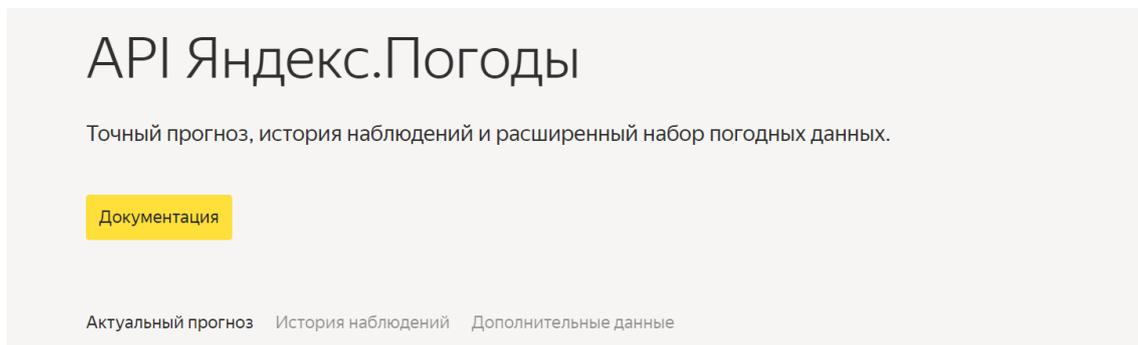
Бағдарламалар мен сайттардың пайдаланушылары тек адамдар ғана емес, оларды басқа сайттар мен қолданбалар да пайдаланады. Олар адамның араласуынсыз бір-бірімен деректер алмаса алады. Бұл дайын шешімді қарызға алу өзіңізді нөлден жасаудан оңайырақ болғанда ыңғайлы.

Spotify ағындық қызметі ойнату тізімдеріне кіруге мүмкіндік беретін ашық API жасады. Оның негізінде әзірлеушілер сервистің мүмкіндіктерін толықтырып, кеңейткен ондаған қызметтерді жасап үлгерді. Міне, кейбір керемет толықтырулардың тізімін атап өтсек:

- электрондық музыканың генеалогиялық ағашы бар қызмет, барлық әндері Spotify API арқылы шығарылады;
- ойнату тізімдері үшін пайдаланушы мұқабаларының дизайнері, содан кейін оларды ресми қолданбада пайдалануға болады;
- 2000-шы жылдар ортасында Spotify ойнату тізімін тыңдауға мүмкіндік беретін Winamp ойнатқышының эмуляторы.

Әзірлеушілер музыкалық мұрағаттарға тегін қол жеткізе алады және олардың негізінде өз идеяларын жүзеге асыра алады; өз кезегінде, қызмет үнемі пайдаланушыларды тартатын әдеттен тыс пайдалануларға ие және пайдаланушылардың музыканы тыңдауға мүмкіндігі бар. Нәтижесінде, API арқасында барлығы ақпараттар қол жетімді болады.

Мысалы, веб-сайтта немесе мобильдік қосымша арқылы ауа-райын көрсету үшін Yandex.Weather API қосуға болады:



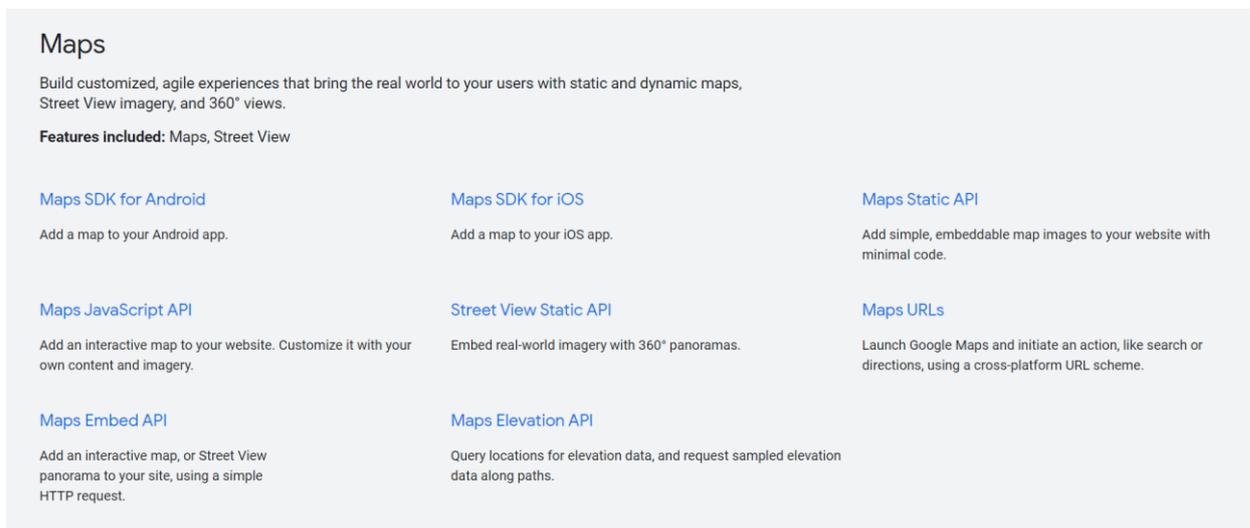
API позволяет получать погодные данные в автоматизированном режиме. Объем предоставляемых данных зависит от выбранного тарифа доступа.

## Актуальный прогноз

Технология Meteum оценивает фактическую погоду и строит прогноз для любых координат на территории России. Компании могут получать и транслировать эти данные множеством способов, начиная с электронных писем и заканчивая виджетами в собственном интерфейсе.

## 2.1-сурет – Yandex.Weather API

Сайт немесе қосымша пайдаланушылары картада орынды көру үшін Google Maps API қосуға болады:



## 2.2-сурет – Google Maps API

Жаһандық шешімдер үшін әзірлеушілер олар қазірдің өзінде іске асырылған жерге жүгінеді, себебі API қосу өз ауа райы қызметін іске қосудан немесе жаңа спутниктік карталарды әзірлеуден оңайырақ. API жай ғана екі

қызметке ақпарат алмасуға көмектеседі, бірақ бұл өзара әрекеттесу пайдаланушыға көрінбейді, себебі ол сыртқы интерфейс арқылы жасырылады.

Қосымша интерфейсi iшкi механизмнiң жұмыс iстеуi үшiн пайдаланушы меңгеруi керек жүйе. Егер сiз автокөликтiң бақылау тақтасындағы белгiлi бiр түйменi бассаңыз, терезе әйнегi төмен қарай айнала бастайды. Жолаушы да, жүргiзушi де терезенi түсiретiн механизмдi көре алмайды, олар тек соңғы нәтиженi көредi. Сол сияқты, қолданба интерфейсiндегi түйменi басу арқылы бiзге қажеттi беттi ашатын механизмдердi көрмеймiз.

Дәл осы принцип бойынша API әзiрлеушiлерге құрылғыларының егжей-тегжейлерi туралы ойланбастан, өздерiнiң және үшiншi тарап бағдарламаларының өзара әрекеттесуiн реттеуге көмектеседi. Сiз Орталық банк веб-сайтының серверi қалай ұйымдастырылғанын бiлмеуiңiз мүмкiн, бiрақ оның API арқылы қызметiңiздегi валюта бағамдарын көрсете аласыз. Мәселен, мысалы, әзiрлеушiлерге арналған «VKontakte API интерфейсiн енгiзу» бетiнде былай делiнген: «Дерекқордың қалай орналасқанын, оның қандай кестелер мен өрiстерден тұратынын егжей-тегжейлi бiлудiң қажетi жоқ, өйткенi API сұрауы бұл туралы «бiледi».

Екi қызметтiң өзара әрекеттесуiн орнату ынтымақтастық шарттарын белгiлейтiн шарттың бiр түрi болып табылады. Әрбiр қызметке арналған API құжаттамасы үшiншi тарап сайттары немесе қолданбалары оның функционалдығын пайдалана алатын шарттарды көрсетедi. Мысалы, сол ВКонтакте әлеуметтiк желiсi әзiрлеушiлерге пайдаланушы идентификаторларын, бейнелердi, құжаттарды, өнiм карталарын үшiншi тарап қызметтерiнде пайдалану мүмкiндiгiн бередi.

API интерфейсiмен дұрыс жұмыс iстеу үшiн құжаттама оларды орындауға болатын арнайы операциялар мен әдiстердi белгiлейдi:

video.add - ВКонтакте ойнатқышынан бейне қосу әдiсi;

likes.delete - «Ұнату» белгiсiн жою әдiсi;

users.get — пайдаланушы туралы ақпарат алу әдiсi.

Әдiстер мазмұн бөлiмi бойынша топтарға бөлiнедi: фотосуреттер, бейнелер, құжаттар, ұнатулар және т.б. Әдiстердiң толық тiзiмiн ВКонтакте желiсiндегi API құжаттамасының арнайы бетiнен табуға болады.

Құжаттамаға сәйкес, әдiстердi қолдана отырып, әрбiр әрекеттi орындау үшiн белгiлi бiр ережелерге сәйкес сұрау салу қажет. Мысалы, ВКонтакте үшiн сiзге мыналарды көрсету керек:

https:// — қосылу протоколы, ол әрқашан бiрдей болады;

api.vk.com/method - API қызметiнiң мекенжайы; қызметке байланысты ол өзгередi, бiрақ ВКонтакте API мекенжайы дәл осылай көрiнедi;

users.get - белгiлi бiр VK API әдiсiнiң атауы;

user\_id=210700286 — деректерiн алу қажет нақты пайдаланушының идентификаторы.

https://api.vk.com/method/users.get?user\_id=210700286&v=5.52

Ключи доступа    Callback API    Long Poll API    [Добавить сервер](#)

---

Настройки сервера    Типы событий    Запросы

---

Название: Сервер 1 · [Изменить](#)

Версия API:

Адрес:  [?](#)

Для получения уведомлений нужно подтвердить адрес сервера. На него будет отправлен **POST-запрос**, содержащий JSON:  
**{ "type": "confirmation", "group\_id": 19340724 }**

Строка, которую должен вернуть сервер: **00a306de** 

[Подтвердить](#)

---

Секретный ключ:  [?](#)

[Сохранить](#)

---

### 2.3-сурет – Вконтакте API

Сұранысқа жауап ретінде сервер сұралған деректері бар JSON нысанын қайтарады, ол келесідей болады:

```
{ "response": [ { "id": 210700286, "first_name": "Lindsey", "fast_name": "Stirling" } ] }
```

Жауапты құрамдас бөліктерге бөлетін болсақ, онда:

- 1) идентификатор өрісінде бұрынғыдай шоттың сәйкестендіру нөмірі сақталады;
- 2) бірінші\_аты өрісінде - пайдаланушы аты бар жол;
- 3) фамилия жолында - пайдаланушының тегі бар жол.

Пайдаланушы туралы ақпаратты алу әдісін, мысалы, ВКонтакте аккаунты арқылы авторизациялау мүмкіндігі бар сайттар пайдаланады. Осылайша, сайт осы әлеуметтік желіде өз қауымдастығының терезесін байланыстыра алады, пайдаланушыға достарының қайсысы оның мүшесі екенін көрсете алады,

парақшаны профильде бөлісуді ұсына алады немесе сайт бетіне тікелей «Ұнатады» батырмасын басу арқылы ұнату қоя алады.

API интерфейстері әзірлеушілерге өз қызметтерінде алдын ала құрастырылған құралдарды пайдалануға мүмкіндік беретіндіктен, бұл әдіс бірден екі стратегиялық артықшылық береді:

- әзірлеу уақытын үнемдейді;
- әзірлеу құнын төмендетеді.

Бағдарламаның қай тілде жазылғаны API үшін маңызды емес. Интерфейс құрылғылардың барлық түрлерімен, операциялық жүйелермен және бағдарламалау тілдерімен жұмыс істейтін әмбебап адаптер болып табылады.

API мүмкін болатын функциялар жиынтығына қатаң шектеулер қоймайды - интерфейске пайдаланушылар немесе үшінші тарап компаниялары үшін пайдалы болатын кез келген функционалдылықты қосуға болады. Ең бастысы, әрбір функцияның қатаң құрылымы бар:

- API көмегімен орындауға болатын операция;
- пайдаланушы сұрайтын деректер;
- пайдаланушы шығыста алатын деректер (мазмұн немесе қате туралы хабар).

Әзірлеушілер API интерфейстеріне ақылы кіруді қамтамасыз етеді. Мысалы, сіз 2GIS карталарына қол жеткізуді, Yandex.Market өнімдері туралы ақпаратты және HeadHunter сайтында жұмыс виджеттерін орналастыру мүмкіндігін сұрауға болады. Twitter, Facebook, VKontakte сияқты әлеуметтік желілердің API арқылы көптеген сайттар авторизациялау жүйесін жүзеге асырады.

*ВКонтакте API арқылы тікелей трансляциялар жасау үшін үшінші тарап әзірлеушілеріне қолжетімділікті ашады.* «ВКонтакте» әлеуметтік желісі барлық үшінші тарап әзірлеушілеріне тікелей хабар тарату технологияларына қол жеткізуді қамтамасыз етеді - олар ағынды екі жақты интеграциялау мүмкіндігіне ие болады. Қоғамдық аударма API (Application Programming Interface) енді веб-сайттарға, шағын қолданбаларға және дербес қолданбаларға ендіру үшін қол жетімді.

Үшінші тарап әзірлеушілері VK бейне платформасында тікелей трансляцияларды құруды және басқаруды автоматтандыру мүмкіндігіне ие болды. API функционалдығы аударма интерфейсінің негізгі мүмкіндіктеріне сәйкес келеді. API арқылы сіз сілтемені алуға және RTMP ағынын хабар таратуға тасымалдауға болады, мысалы, арнайы студиялық бағдарламалық құралда жинақталған, атауды, сипаттаманы орнатуға және құпиялылықты орнатуға, сондай-ақ хабар тарату кезінде көрермендермен өзара әрекеттесу үшін интерактивті механиканы пайдалануға және аудиторияның жауабын қадағалау. Бейнені тарату және таратуды аяқтау үшін RTMP мекенжайын алу әдістері әзірлеуші платформасында бұрыннан бар.

Бейне ағыны үшін API пайдалану бизнеске әлеуметтік желідегі қатысуын әртараптандыруға көмектеседі және контент жасаушыларға қосымша мүмкіндіктер береді - бұл бейне түсіру қызметтерімен интеграцияны

жеңілдетеді және мазмұнмен жұмыс істеу процесін мүмкіндігінше автоматтандырады, бұл әсіресе жиі жариялайтын авторлар мен қауымдастықтарға арналған.

Қоғамдық API кең ауқымды үшінші тарап әзірлеушілеріне өздерінің қосымшаларында Вконтакте мүмкіндіктерін пайдалануға және әлеуметтік желінің механикасымен интеграцияланған жобаларды әзірлеуге арналған шешімдер жасауға көмектеседі. Сервис әзірлеушілері бейне мазмұнмен жұмыс істеудің жаңа құралын алды - API арқылы бейнелерді жүктеп салудан, қауымдастықтардағы оқиғаларды бақылаудан және әзірлеушілерге арналған VK платформасында қол жетімді көптеген басқа функциялардан басқа.

Александр Тобол, Вконтакте және VK видеосының техникалық директоры: «Қоғамдық API бейнемен жұмыс істеуге арналған дайын класстардың, функциялардың және құрылымдардың үлкен жиынтығын ұсынады және платформада мазмұнды орналастыруды автоматтандыруға мүмкіндік береді: бейнелерді жүктеп салу, пайдаланушы бетінде немесе қауымдастықтарда ойнату тізімдерін жасау және өңдеу, бейне деректерін өңдеу, пікірлер қосу және статистиканы жинау. Вконтакте командасы пайдаланушылардың пікірлері мен жиі қолданылатын жағдайларға негізделген API мүмкіндіктерін кеңейтуді жоспарлап отыр.

Вконтакте үшінші тарап әзірлеушілеріне жеке қызметтерді жасау үшін өз өнімдерінің интерфейсін пайдалануға мүмкіндік береді. Әлеуметтік желі VK Mini Apps, кірістірілген кросс-платформалық Вконтакте қолданбаларына арналған платформаны әзірлейді. Олар HTML, JavaScript, CSS стандартты веб-технологиялары негізінде жасалған және барлық әзірлеушілерге қолжетімді.

### 3 Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық жүйелердің қосымшасын құру және оны іске асыру

#### 3.1 Бағдарламалау тілін таңдау негіздемесі

Жүйені жасау барысында бірнеше программалау тілдері мен құралдар қолданылды. Солардың ең маңыздылары мыналар:

Қолданылған тілдер:

- JavaScript;
- TypeScript;
- HTML;
- CSS (SCSS).

Қолданылған құралдар және технологиялар:

- Visual Studio Code;
- HTTP сұраныстары;
- Postman.

**JavaScript** - бұл веб-бетте күрделі әрекеттерді орындауға мүмкіндік беретін тіл - бұл веб-бетте оны статикалық түрде көрсетуден басқа нәрсе болған кезде - мерзімді жаңартылатын мазмұнды немесе интерактивті карталарды көрсету немесе 2D / 3D графикасын анимациялау немесе бейнені айналдыру ойыншы және т. - бұл JavaScriptсіз болмағанына сенімді бола аласыз.

**TypeScript** - Microsoft корпорациясы 2012 жылы енгізген және JavaScript мүмкіндіктерін кеңейтетін веб-қосымшаларды әзірлеу құралы ретінде орналастырылған бағдарламалау тілі.

**Visual Studio Code** - Windows, Linux және macOS үшін Microsoft әзірлеген бастапқы код редакторы. Ол веб және бұлттық қолданбаларды кросс-платформалық әзірлеуге арналған «жеңіл» код редакторы ретінде орналастырылған.

**Postman** - бұл API құруға және тұтынуға арналған API платформасы. Postman API өмірлік циклінің әрбір қадамын жеңілдетеді және жақсырақ API интерфейстерін жылдам құруға мүмкіндік беретін ынтымақтастықты жеңілдетеді.

### 3.2 Жүйенің интерфейсін Angular фреймворгінде жобалау

Жобамызды Angular фреймворгінде жасадық. Angular – бұл клиенттік қолданбаларды құруға арналған Google-дың фреймворгі. Ең алдымен, ол SPA-шешімдерді (Single Page Application), яғни бір беттік қосымшаларды әзірлеуге бағытталған. Осыған байланысты Angular басқа AngularJS фреймінің мұрагері болып табылады. Сонымен қатар, Angular AngularJS-тің жаңа нұсқасы емес, түбегейлі жаңа фреймворк болып табылады.

Angular екі жақты байланыстыру сияқты функционалдылықты қамтамасыз етеді, ол үлгі деректері басқа жерде өзгерген кезде интерфейстің бір жеріндегі деректерді динамикалық түрде өзгертуге мүмкіндік береді, үлгілер, маршруттау және т.б.



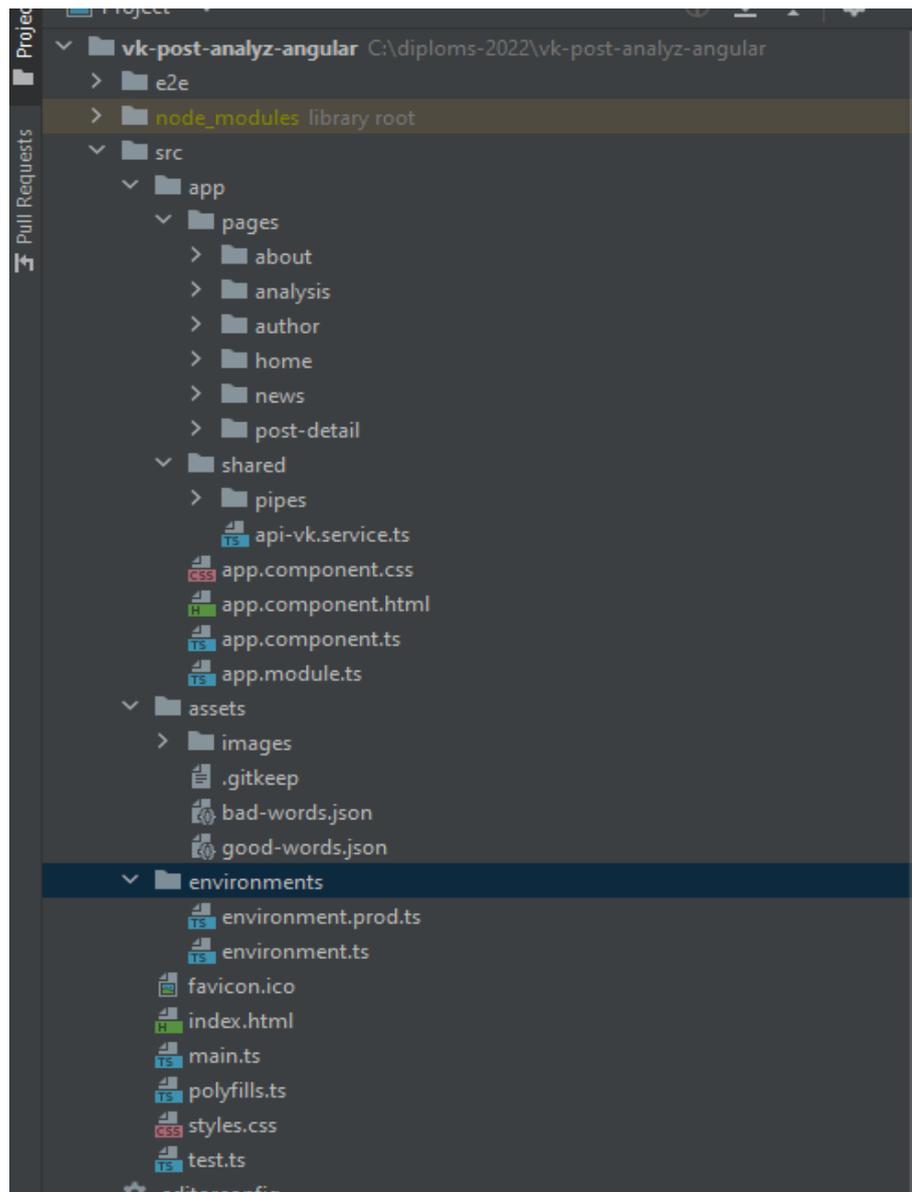
3.1-сурет – Angular-дың логотипі

Angular бағдарламасының негізгі ерекшеліктерінің бірі оның бағдарламалау тілі ретінде TypeScript қолдануы болып табылады. Сондықтан жұмысқа кіріспес бұрын осы тілдің негіздерімен танысу ұсынылады, ол туралы мына жерден оқи аласыз.

Бірақ біз тек TypeScript тілімен шектелмейміз. Қаласаңыз, Dart немесе JavaScript сияқты тілдерді пайдаланып Angular тілінде қосымшаларды жаза аламыз. Дегенмен, TypeScript әлі де Angular үшін негізгі тіл болып табылады.

Angular, Angular 13 соңғы нұсқасы 2021 жылдың қарашасында шығарылды. Github жүйесіндегі рамканың ресми репозиторийі: <https://github.com/angular/angular>. Онда сіз бастапқы файлдарды, сондай-ақ кейбір қосымша ақпаратты таба аласыз.

Жоба Angular-дың 8-ші нұсқасында жасалынды. Ең алдымен жобаны `ng new` командасы арқылы құрылды. Сосын керекті роуттар мен компоненттерді құрылды (3.2-сурет).



### 3.2-сурет – Жобаның құрылымы

Егер біз құрастыру үшін Angular және Angular CLI-мен жұмыс істеу үшін TypeScript қолданатын болсақ, онда бұл package.json, tsconfig.json және angular.json файлдары шын мәнінде әрбір жобада болады. Және оларды жобадан жобаға ең аз өзгерістермен ауыстыруға болады. Мысалы, angular.json файлында жоба атауы болады. Package.json файлында алдыңғы нұсқалар ескірген болса (3.3-сурет), пакеттердің кейбір басқа нұсқаларын көрсету мүмкін болады. Жобаның атын, нұсқасын өзгертуге болады. TypeScript немесе Angular CLI параметрлерін өзгертуге болады, бірақ жалпы ұйым бірдей болады.

```
package.json
13 "dependencies": {
14   "@angular/animations": "~8.0.1",
15   "@angular/cdk": "^8.0.0",
16   "@angular/common": "~8.0.1",
17   "@angular/compiler": "~8.0.1",
18   "@angular/core": "~8.0.1",
19   "@angular/forms": "~8.0.1",
20   "@angular/platform-browser": "~8.0.1",
21   "@angular/platform-browser-dynamic": "~8.0.1",
22   "@angular/router": "~8.0.1",
23   "@nebular/eva-icons": "4.5.0",
24   "@nebular/theme": "^4.5.0",
25   "bootstrap": "^4.3.1",
26   "eva-icons": "^1.1.1",
27   "jquery": "^3.4.1",
28   "moment": "^2.24.0",
29   "rxjs": "~6.4.0",
30   "tslib": "^1.9.0",
31   "zone.js": "~0.9.1",
32   "chart.js": "2.7.2",
33   "ng2-charts": "1.6.0"
34 },
35 "devDependencies": {
36   "@angular-devkit/build-angular": "~0.800.0",
37   "@angular/cli": "~8.0.4",
38   "@angular/compiler-cli": "~8.0.1",
39   "@angular/language-service": "~8.0.1",
40   "@schematics/angular": "~8.0.4",
41   "@types/jasmine": "~3.3.8",
42   "@types/jasminewd2": "~2.0.3",
43   "@types/node": "~8.9.4",
44   "codifyer": "^5.0.0",
45   "jasmine-core": "~3.4.0",
46   "jasmine-spec-reporter": "~4.2.1",
47   "karma": "~4.1.0",
48   "karma-chrome-launcher": "~2.2.0",
49   "karma-coverage-istanbul-reporter": "~2.0.1",
```

### 3.3-сурет – Жобаның package.json файлының құрылымы

Жобада ең керекті мына кітапханаларды қолданылды:

- "bootstrap": "^4.3.1";
- "eva-icons": "^1.1.1";
- "jquery": "^3.4.1";
- "moment": "^2.24.0";
- "rxjs": "~6.4.0";
- "chart.js": "2.7.2".

**Bootstrap** (сондай-ақ Twitter Bootstrap ретінде белгілі) - веб-сайттар мен веб-қосымшаларды құруға арналған тегін құралдар жиынтығы. Типографияға, веб-пішіндерге, түймелерге, белгілерге, шарлау блоктарына және JavaScript кеңейтімдерін қоса, басқа веб-интерфейс құрамдастарына арналған HTML және CSS дизайн үлгілерін қамтиды. Bootstrap CSS және HTML соңғы әзірлемелерін пайдаланады, сондықтан ескі браузерлерге қолдау көрсету кезінде абай болу керек.

**Eva Icons** - бұл жалпы әрекеттер мен элементтерге арналған 480-ден астам әдемі ашық бастапқы код белгішелерінің жиынтығы.

**jQuery** - JavaScript және HTML арасындағы өзара әрекеттесуге бағытталған JavaScript мүмкіндіктері жиынтығы. jQuery кітапханасы кез келген DOM элементіне қол жеткізуді, DOM элементтерінің атрибуттары мен мазмұнына

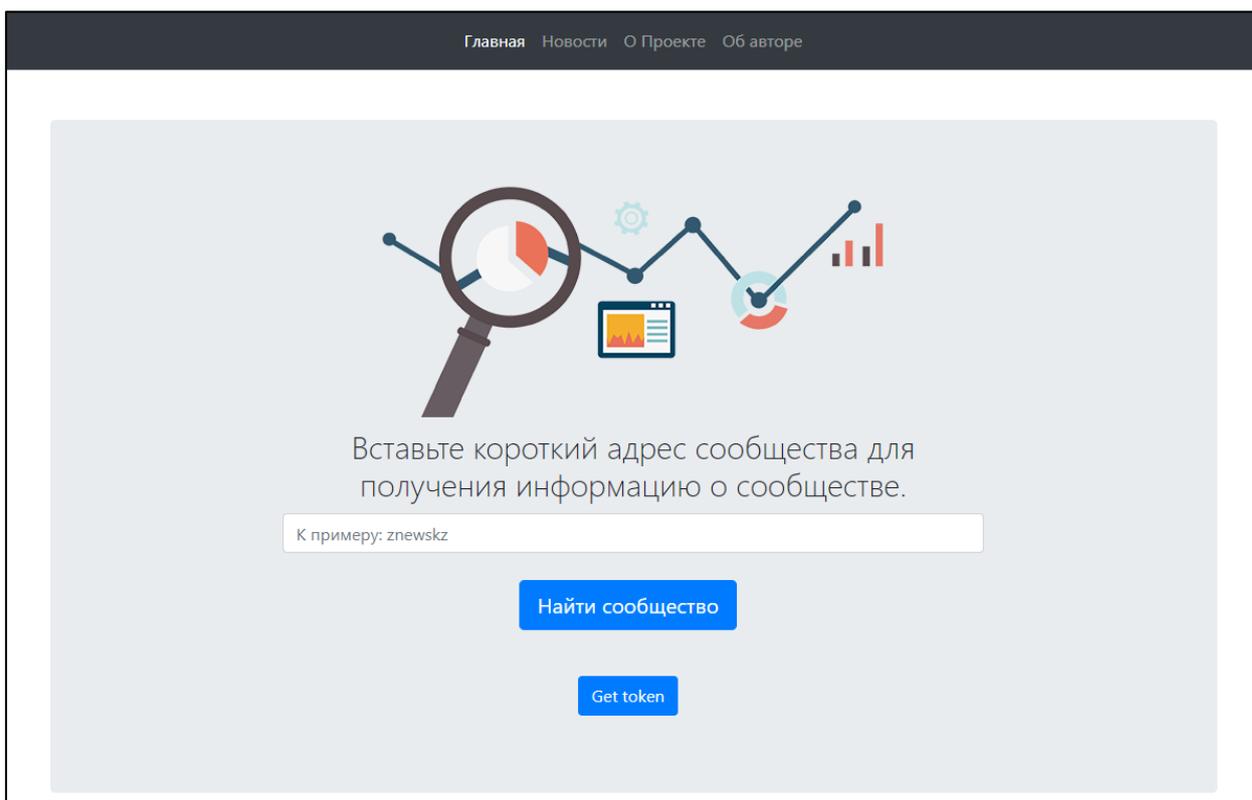
қол жеткізуді және оларды өңдеуді жеңілдетеді. JQuery кітапханасы сонымен қатар AJAX-пен жұмыс істеу үшін ыңғайлы API ұсынады. jQuery-ді қайырылған еріктілер тобы әзірлеген.

**Momentjs** - күндерді талдауға, тексеруге, өңдеуге және пішімдеуге арналған JavaScript күндер кітапханасы.

**RxJS** - бұл асинхронды немесе кері шақыруға негізделген кодты жасауды жеңілдететін Observables көмегімен реактивті бағдарламалауға арналған кітапхана. Бұл жоба API бетін азайтатын кейбір үзіліс өзгерістерімен, негізінен артқа қарай үйлесімді болып қала отырып, жақсырақ өнімділік, жақсы модульдік, жөндеуге болатын қоңыраулар стектері бар Reactive-Extensions/RxJS қайта жазуы.

**Chart.js** - сегіз диаграмма түрін қолдайтын тегін және ашық бастапқы JavaScript деректер визуализациясы кітапханасы: жолақ, сызық, аумақ, дөңгелек, көпіршік, радар, полярлық және шашырау.

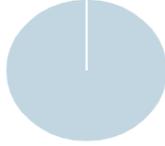
Жоғарыда көрсетілген кітапханалармен жобаның қажетті элементтері жасалынды.



**3.4-сурет – Жобаның бастапқы беті**

Жобаның бұл беті басты беті болып саналады. ВКонтакте әлеуметтік желісіндегі кез-келген ашық топтарды табуға арналған өрісі бар бет (3.4-сурет).

## Группа: РИА Новости

<b>Название сообщества:</b>	РИА Новости		
<b>Описание сообщества:</b>	Официальная страница сайта Ria.ru, МИА "Россия сегодня". Правила общения в сообществе: <a href="http://vk.com/topic-15755094_28764256">http://vk.com/topic-15755094_28764256</a> Почта для связи с командой модераторов <a href="mailto:moderator@rian.ru">moderator@rian.ru</a>		
<b>Короткое название:</b>	ria		
<b>Фото сообщества:</b>			
<b>Доступ сообщества:</b>	открытое		
<b>Тип сообщества:</b>	публичная страница		
<b>Счетчики:</b>	Фотографии: 0 Альбомы: 0 Аудио: 0 Видео: 10229 Топики: 1 Документы: 0	Альбом    Статья    Аудио    Фото Топики    Видео	

### 3.5-сурет – Табылған топ жайлы бет

Получить записи из стены Искать другую

**Всего записей: 196955**

---

А также визовые ограничения в отношении 19 российских бизнесменов, 47 членов их семей и близких.

---

**Просмотров:** 68  
**Лайков:** 1  
**Комментарии:** 1  
**Поделились:** 0  
**Вложения:** 1  
**Опубликован:** Сегодня, в 1:50

Выбрать этот пост

---

500 тысяч долларов.

---

**Просмотров:** 24788  
**Лайков:** 522  
**Комментарии:** 265  
**Поделились:** 296  
**Вложения:** 1  
**Опубликован:** Сегодня, в 1:25

Выбрать этот пост

---

Также породистых кошек, которые выведены в России, нельзя будет зарегистрировать в родословных книгах FIFe.

---

**Просмотров:** 33440  
**Лайков:** 171  
**Комментарии:** 314

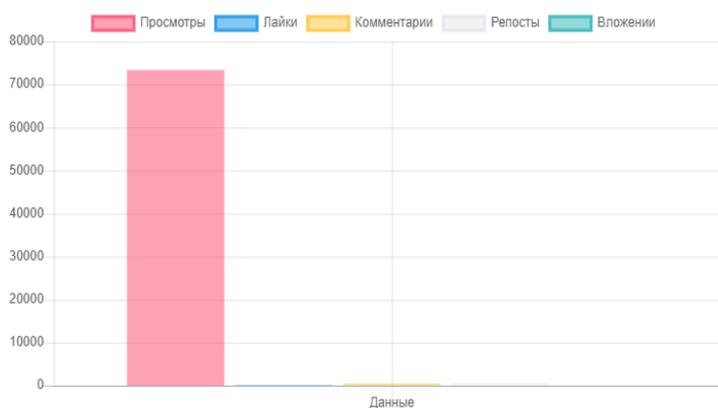
### 3.6-сурет – Топтың жазбалары

## Информация о посте

В ИКЕА в некоторых городах России огромные очереди — 4 марта магазины уже не откроются. Ранее компания сообщила, что приостанавливает работу в России и Белоруссии из-за событий на Украине. Также ИКЕА временно прекращает производство на российской территории.

### Общая статистика

<b>Просмотров:</b>	73365
<b>Лайков:</b>	177
<b>Комментарии:</b>	496
<b>Поделились:</b>	548
<b>Вложения:</b>	1
<b>Опубликован:</b>	Сегодня, в 0:06



### 3.7-сурет – Бір жазбаны тандаған соң шығатын ақпараттар беті

Екатерина Тебецин
Сергей Чистый
Николай Бойко
Елена Осипчук-Сербская
Александр Красин
Света Абрамова
Михаил Аринченков

Roma Serov
Рома Твайс
Дмитрий Шустов
Марина Жердева
Екатерина Крым
Дима Серов

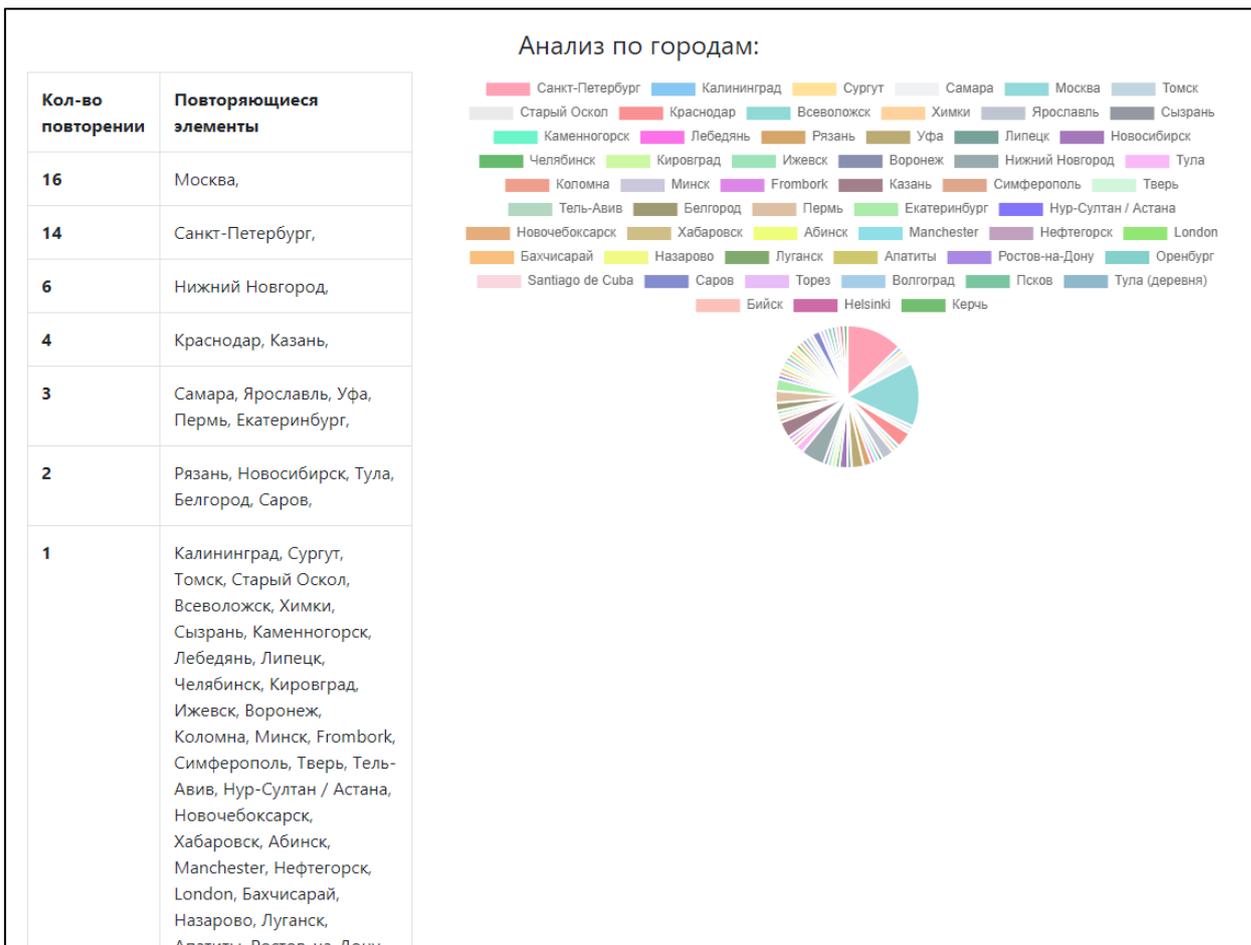
Анализировать пол
Анализировать город
Анализировать возраст

**Анализ по полам:**

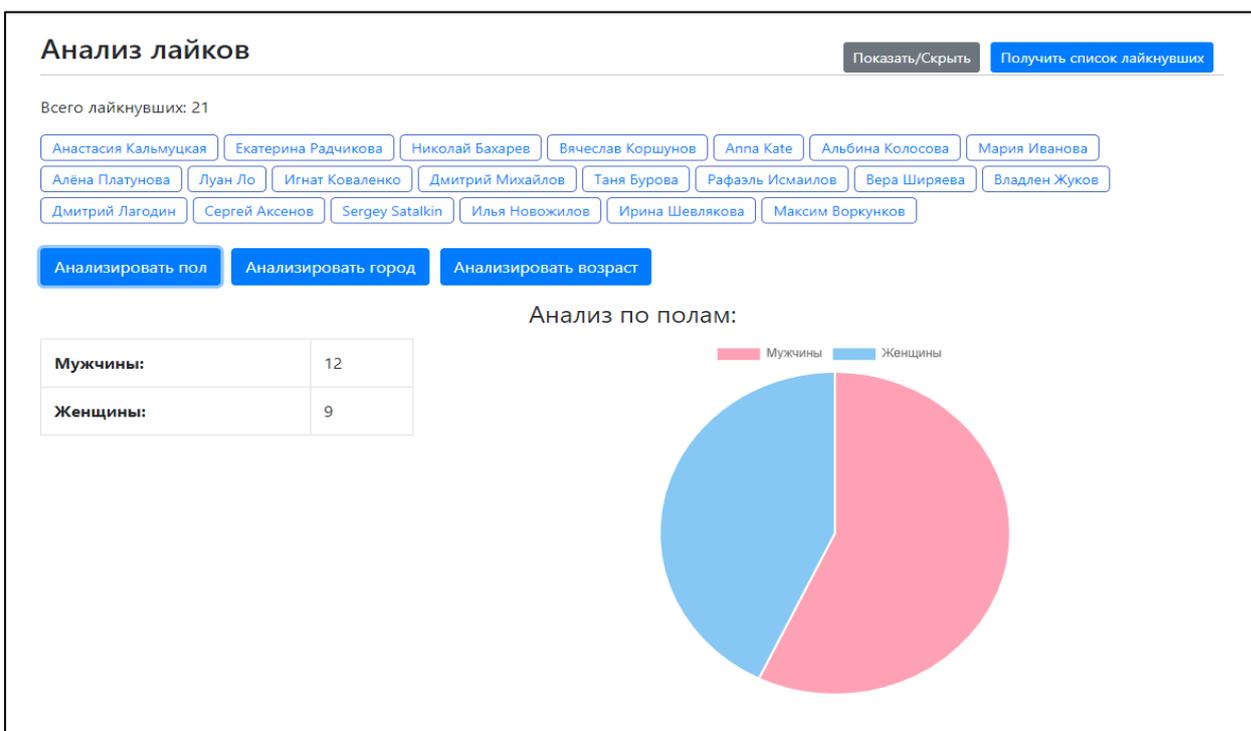
<b>Мужчины:</b>	102
<b>Женщины:</b>	56

Пол	Количество
Мужчины	102
Женщины	56

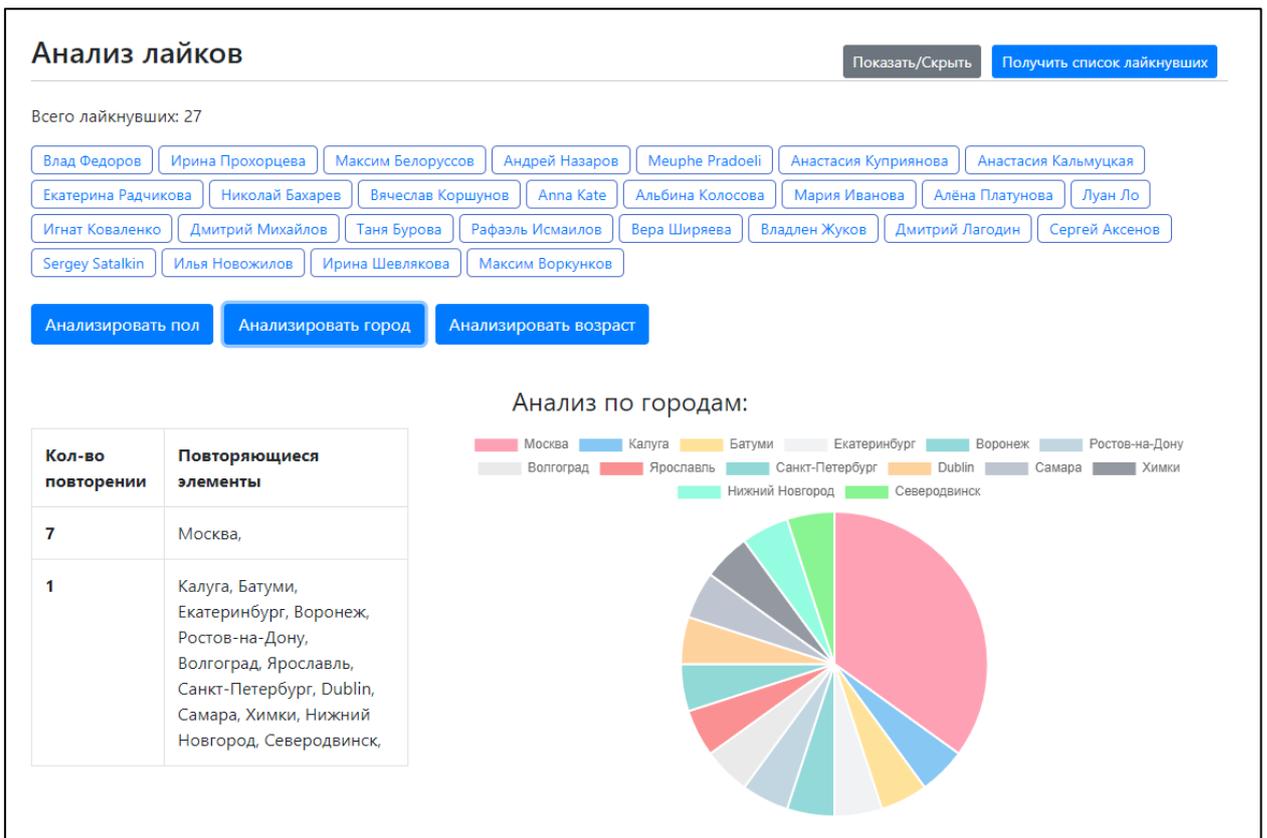
### 3.8-сурет – Пікiрді жазған адамдардың жынысы бойынша статистика



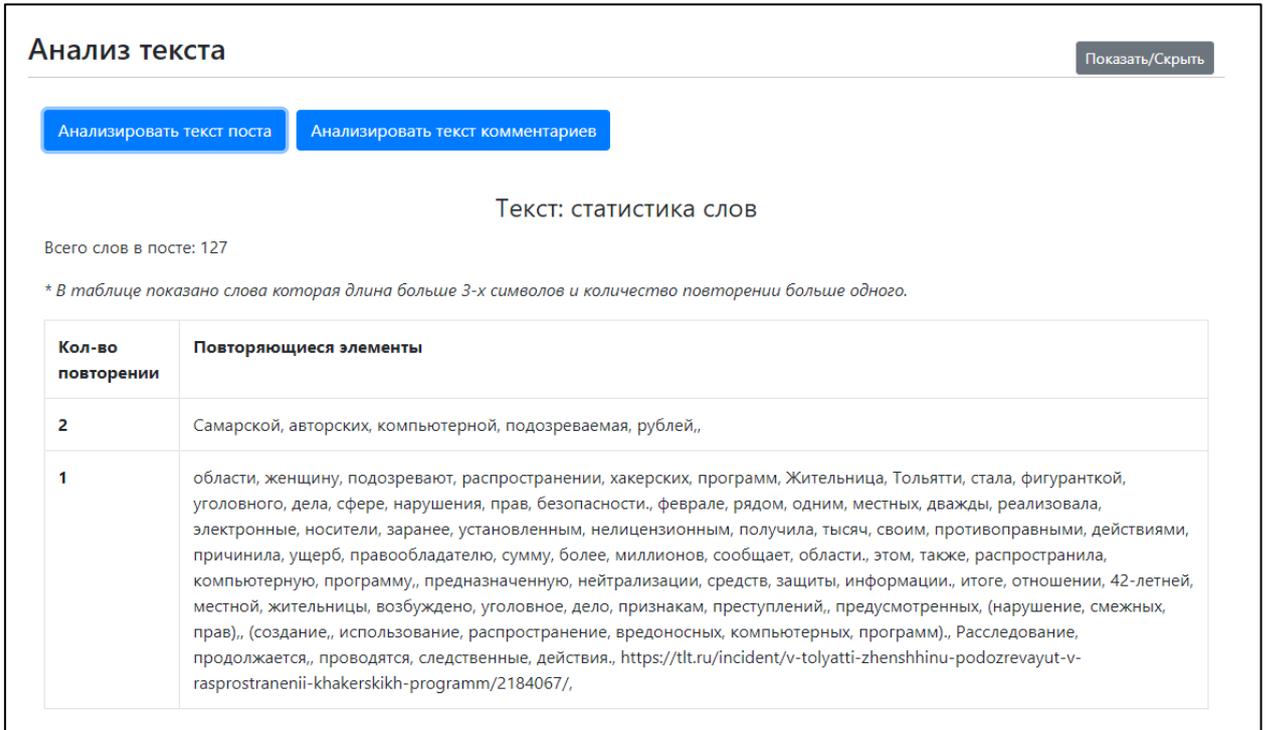
### 3.9-сурет – Пікірді жазған адамдардың қалалары бойынша статистика



### 3.10-сурет – Лайк басқан адамдардың жынысы бойынша статистика



### 3.11-сурет – Лайк басқан адамдардың қалалары бойынша статистика



### 3.12-сурет – Посттың мәтіні бойынша анализ статистикасы

## Текст: статистика слов

Всего слов в посте: 419

\* В таблице показано слова которая длина больше 3-х символов и количество повторении больше одного.

Кол-во повторении	Повторяющиеся элементы
5	санкции,
3	автономном, даже,
2	всего, России», Салехард, запустить, доставку, новость, каждый, устали,
1	Запись, стене, Время, вперед!, 2021, 15:01, «ПОЧТА, РОССИИ», ИСПЫТАЛА, НОВЫЙ, ДРОН, Помните, историю, «дрон, «Почты, России», который, испытаниях, поднялся, воздух, маха, влетел, стену?, Конечно,, дрон, сторонней, компании,, которая, хотела, продемонстрировать, «Почте, свою,, оказалось,, самую, удачную, разработку,, Прошли, годы,, идея, умерла,, сегодня, состоялись, испытания, гораздо, более, серьезной, разработки, вертолётного, типа, компании, «Аэромакс», совершил, несколько, полётов, маршруту, Аксарка, массогабаритным, макетом, груза,, Общая, протяженность, пути, одного, населенного, пункта, другой, составляет, конца, 2024, года, планируется, беспилотную, маршрутам, ЯНАО., Ранее, сообщалось,, «Почта, планирует, экспериментальную, сеть, беспилотной, аэродоставке, грузов,, включающую, маршрутов, четырех, регионах, Чукотском, округе,, Камчатском, крае,, Ямало-Ненецком, округе, Ханты-Мансийском, округе,, Вместе, запуском, 2022, году, нового, авиационного, отделения, обеспечит, оперативную, почты, населенные, пункты, регионов,, которыми, регулярного, транспортного, сообщения,, Источник: ТАСС, #времявперёд, жить, дальше?, 🤔🤔, Прекрасная, 🍌 us, Давно, пора, было, сделать!, Нужно, недвижимость, конфисковать, ещё!, котиков, отошли, опять! 🧑🧑🧑, пусть, поживут, все...ничего, страшного,, возможно, скоро, будут, вводить, всилах, придумать, причину,, ввести, надо, походу, создали, отдел, санкций, себя))), вообще, учитывают?, запуталась, Мечты, сбываются! 😊, Надеюсь, канадские, сфинксы, попадут ☹, Пожар, западном, борделе, продолжался, временем, русские, солдаты, тысячами, погибают, сдаются, плен, день,, <a href="https://youtu.be/UrKaa5WY-YM">https://youtu.be/UrKaa5WY-YM</a> , Адские, санкции,, которых, знал,, закончились,, кошках?, Вернулись, стандартным..., сказали,, россиянен, нём,, Опять, дискриминация, Санкция, Байдену, престарелых, временно,, прям, расстроились 😞, ЗАБЕРИТЕ, СЕБЕ, НАРКОМАНА,ЛЕЧИТЕ, ЕГО!, Ничего,, счас, правительство, поможет, алкаша, соседнего, двора, тоже, введете, санкции?, пытаюсь, выкинуть,, очень, много, хуже,, еще,, интересно, успокоятся, пока, успокоит, Россия.А, какими, методами, можем, успокоить?, Более, жесткими,, Интересно,, сколько, миллионов, российских, солдат, погибнет, версии, ботов,, когда, спецназ, пойдёт, штурм, Киева?), Боже,, одна, лучше, другой), запуталась,, вроде, ввели, против, населения, России, входили, население, ЗЕЛЕНСКИЙ,, ТРЕЗВЫМ,, НУЖЕН,, ещё,?, "megafuckofthefuckbigfuckblocking, sanctions" 😞,

### 3.13-сурет – Пікір жазған мәтіндердің анализ статистикасы

```
rc > assets > { } good-words.json > ...
730     "с иголки",
731     "Смело",
732     "Сногшибательно",
733     "Стильно",
734     "Строго",
735     "Торжественно",
736     "Чудесно",
737     "Шикарно",
738     "Эффектно",
739     "Кумир",
740     "Тронет душу",
741     "Согреть",
742     "Легендарной",
743     "Желаю тебе",
744     "Очаровательную",
745     "Береги",
746     "Чудо",
747     "Вау",
748     "Исцеление",
749     "Бомба",
750     "Влюбляюсь",
751     "Лекарство",
752     "Великолепно",
753     "Родной",
754     "Тёплой",
755     "Добрые",
756     "Улыбка",
```

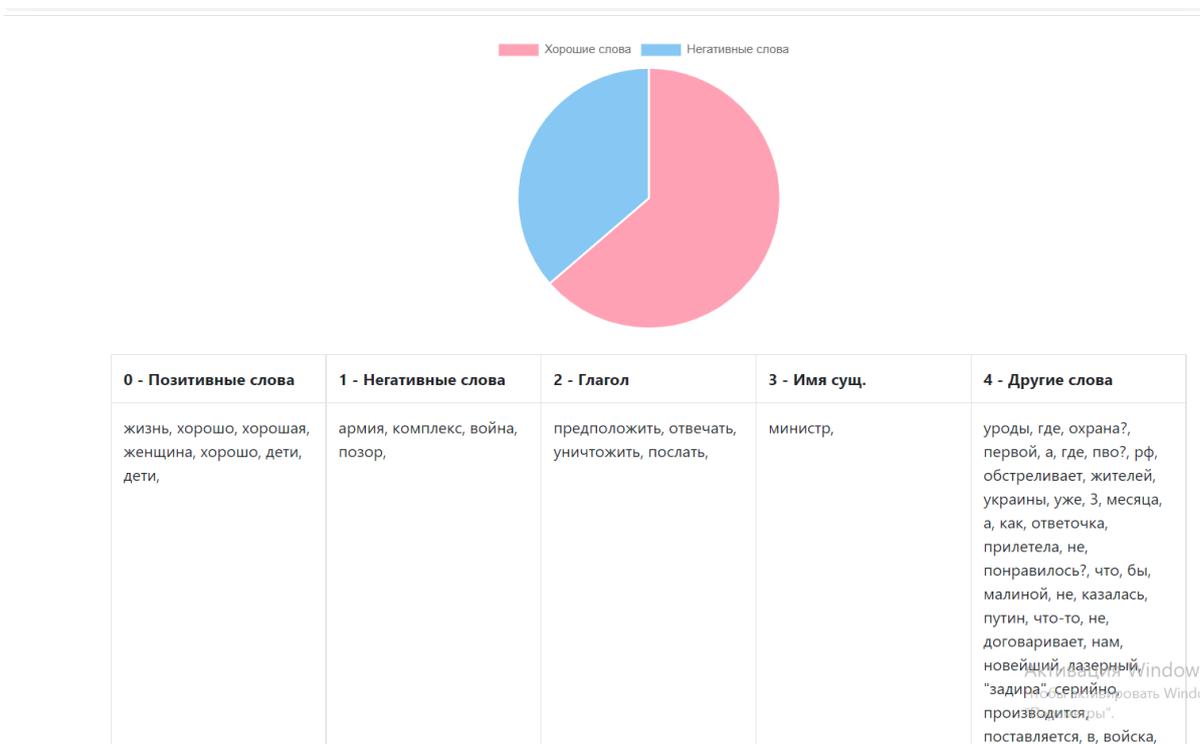
### 3.14-сурет – Пікір мәтінін анализдеуге жақсы сөздері анықтау базасы

```

assets > {} bad-words.json > ...
44 "подавленность",
45 "сессия",
46 "экзамен",
47 "стресс",
48 "грусть",
49 "слёзы",
50 "печаль",
51 "боль",
52 "бессилие",
53 "упадок",
54 "усталость",
55 "одиночество",
56 "ужас",
57 "фобия",
58 "псих",
59 "суицид",
60 "депрессия",
61 "пессимизм",
62 "кризис",
63 "состояние",
64 "скука",
65 "неврох",
66 "психология",
67 "страх",
68 "бессонница",

```

3.15-сурет – Пікір мәтінін анализдеуге жаман сөздері анықтау базасы



3.16-сурет– Пікір мәтінін жақсы және жаман сөздер бойынша анализі

## ҚОРЫТЫНДЫ

Әлеуметтік желілердегі пайдаланушы деректерін талдауға арналған технологиялық стектің негізгі құрамдас бөліктері қарастырылады. Желілік және мәтіндік деректерді талдаудың міндеттері, әдістері мен қолданбалары сипатталған: пайдаланушылардың демографиялық атрибуттарын анықтау, хабарламалар корпусындағы оқиғалардың сипаттамасын іздеу, әртүрлі желілерді пайдаланушыларды анықтау, пайдаланушылар қауымдастығын іздеу және пайдаланушылар арасындағы ақпараттық әсерді өлшеу. Сонымен қатар, талдау үшін бастапқы деректерді алу тәсілдері қарастырылады: әлеуметтік қызметтердің веб-интерфейстеріне қол жеткізу және кездейсоқ әлеуметтік графиктерді құру арқылы нақты деректерді жинау.

Әлеуметтік желілердің әлеуметтік-мәдени феномен ретіндегі дамуындағы басым тенденциялардың бірі адамның әлеуметтік мінез-құлқының ерекшеліктерін тереңірек түсіну және соның нәтижесінде өзін-өзі көрсетудің жаңа құралдарын жасау, сондай-ақ ақпарат алмасу және тәжірибе. Әлеуметтік желілердің пайдаланушы моделі мен функционалдық мүмкіндіктерінің одан әрі кеңеюін күту орынды, бұл әлеуметтік графиктің объектілері мен сілтемелері түріндегі деректердің жаңа түрлерінің пайда болуына және нәтижесінде неғұрлым тиімдірек болу мүмкіндігіне әкеледі. жеке ақпаратты өңдеуге байланысты мәселелерді шешу.

Дипломдық жұмыстың соңында ВКонтакте әлеуметтік желісінің API-мен жұмыс жасайтын жүйе құрастырылды. Бұл жүйе ақрылы әлеуметтік желідегі әр түрлі ақпараттар мен мәтіндерді анализдеуге өте ыңғайлы және қолайлы.

## Пайдаланылған әдебиеттер тізімі

- 1 Джин Ю., Луо Ц., Гуо В., Си Дж., Ву Д., Ван Р. Шартты рефлексияға негізделген мәтінді жіктеу. IEEE рұқсаты. 2019; 7:76712-76719. DOI: <https://doi.org/10.1109/ACCESS.2019.2921976>
- 2 Фещенко А., Гойко В., Можаяева Г., Шиляев К., Степаненко А. Болашақ қатысушыларды іздеу үшін әлеуметтік желілердегі пайдаланушы профильдерін талдау // INTED2017 Proceedings, 11th International Technology, Education and Development Conference. 6–8 наурыз 2017 ж., 5188–5194 беттер. DOI: 10.21125/int.2017.1203
- 3 Степаненко А.А., Шиляев К.С., Резанова З.И. Ғаламшардың тақырыптық топтары мен беттерінің мәтіндеріне негізделген «ВКонтакте» әлеуметтік желілерін пайдаланушылардың кәсіби мүдделерінің атрибуты // 2018. No 52. С. 130–144. DOI: 10.17223/19986645/52/8
- 4 Шешасаайы А., Тайламбал Г. Мәтіндегі жіктеу алгоритмдерін салыстыру // Халықаралық таза және қолданбалы математика журналы. 2017. том. 116, No 22, 425–433 б.
- 5 Сингхал А., Гопалакришнан К., Хайтан С.К. Көлік зерттеулерінің грантына негізделген бюджетті болжау Сипаттама: Мәтінді игеру және машиналық оқытудың барлау талдауы // Азаматтық инженериядағы жұмсақ есептеулер журналы. 2017. № 1–2
- 6 Iandola F.N., Han S., Moskewicz M.W., Ashraf K., Dally W.J., Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv:1602.07360v4 [cs.CV], 2016. 13 p. Available at: <https://arxiv.org/pdf/1602.07360.pdf> (accessed 06.12.2019). (In Eng.)
- 7 Teerapittayanon S., McDanel B., Kung H.T. Distributed Deep Neural Networks Over the Cloud, the Edge and End Devices. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA; 2017. p. 328-339. (In Eng.) DOI: <http://doi.org/10.1109/ICDCS.2017.226>
- 8 Prusa J. D., Khoshgoftaar T. M. Designing a Better Data Representation for Deep Neural Networks and Text Classification. In: 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), Pittsburgh, PA; 2016. p. 411-416. (In Eng.) DOI: <https://doi.org/10.1109/IRI.2016.61>
- 9 Wen Y., Zhang W., Luo R., Wang J. Learning text representation using recurrent convolutional neural network with highway layers. arXiv:1606.06905 [cs.CL], 2016. Available at: <https://arxiv.org/abs/1606.06905> (accessed 06.12.2019). (In Eng.)
- 10 Rajasundari T., Subathra P., Kumar P.N. Performance Analysis of Topic Modeling Algorithms for News Articles. Journal of Advanced Research in Dynamical and Control Systems. 2017; 11:175-183. (In Eng.)
- 11 Liu P., Qiu X., Huang X. Recurrent neural network for text classification with multi-task learning. arXiv:1605.05101, 2016. Available at: <https://arxiv.org/abs/1605.05101> (accessed 06.12.2019). (In Eng.)

- 12 Lai S., Xu L., Liu K., Zhao J. Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15). AAAI Press; 2015. p. 2267-2273. (In Eng.)
- 13 АКТ қолдана отырып оқытудағы әлеуметтік желі: талдау. ескертпе, наурыз 2011 / ЮНЕСКО білім берудегі ақпараттық технологиялар институты. – М.: ИТЕ, 2011. – 12 б. URL: <http://iite.unesco.org/pics/publications/ru/files/3214685.pdf>
- 14 Губанов Д.А.Әлеуметтік желілер: ақпараттық әсер ету, бақылау және қарама-қайшылық модельдері / Д.А.Губанов, Д.А.Новиков, А.Г.Чхартишвили. - М.: Физика-математикалық әдебиеттер баспасы, 2010. - 228с.
- 15 Данич В.Н., Бельченко А.А. «Әлеуметтік желі» ұғымының этимологиясы және дамуы, 2011. - № 2. URL: <http://dspace.snu.edu.ua:8080/jspui/bitstream/123456789/822/1/Danich.pdf>
- 16 Вконтакте API арқылы тікелей трансляциялар жасау үшін үшінші тарап әзірлеушілеріне қолжетімділікті ашады. Электрондық ресурс: [https://www.cnews.ru/news/line/2022-04-08\\_vkontakte\\_otkryvaet\\_dostup](https://www.cnews.ru/news/line/2022-04-08_vkontakte_otkryvaet_dostup)
- 17 Әлеуметтік медиа деректерін талдау. Электрондық ресурс: <https://www.osp.ru/os/2015/03/13046896>
- 18 Рубцова Ю. В. Разработка и исследование предметно независимого классификатора текстов по тональности // Труды СПИИРАН. - 2014. - Т. 5. - №. 36.- С. 59-77.
- 19 Рубцова Ю. Автоматическое построение и анализ корпуса коротких текстов (постов микроблогов) для задачи разработки и тренировки тонового классификатора // Сборник трудов конференции «Инженерия знаний и технологии семантического веба - 2012». - СПб.: НИУ ИТМО, 2012. - С. 109-115.
- 20 Лукашевич Н., Рубцова Ю. Объектно-ориентированный анализ твитов по тональности: результаты и проблемы // Труды Международной конференции DAMDID/RCDL-2015. — Обнинск, 2015. — С. 499-507.

## **А қосымшасы**

### **Техникалық тапсырма**

#### **А.1 Жалпы сипаттама**

Веб-парақша негізінде анализ жасау және мәтіндерді жіктеу ретінде пайдалану үшін жасалған болатын.

#### **А.2 Мақсаты**

Қосымша кез келген тұтынушыға қолжетімді, әрі тегін мақсатында жасалған.Әлеуметтік желілер мәтіндерін жіктеу үшін ақпараттық платформа ретінде пайдалана отырып уақыт үнемдеу және нақты ақпарат алу негізгі мақсат.

#### **А.3 Функционалдық сипаттамаларына қойылатын талаптар**

Веб-қосымша барынша ынғайлы және түсінікті, яғни тұтынушыға қол жетімді.

#### **А.4 Сенімділікке талап**

Қосымшаны қолданушылардың қандай да бір жеке деректері сұралмайды, және де сақталмайды.

#### **А.5 Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар**

- Құрылғыға қойылатын жалпы талаптар:
- ғаламторға қол жетімділік;
  - ақпарат енгізу және басқару құрылғылары.

## Б қосымшасы

### Бағдарлама коды

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';
import { RouterModule } from '@angular/router';
import { AuthorComponent } from './pages/author/author.component';
import { AboutComponent } from './pages/about/about.component';
import { NewsComponent } from './pages/news/news.component';
import { HomeComponent } from './pages/home/home.component';
import { HttpClientModule } from '@angular/common/http';
import { BrowserModuleAnimationsModule } from '@angular/platform-
browser/animations';
import { NbThemeModule, NbLayoutModule } from '@nebular/theme';
import { NbEvalIconsModule } from '@nebular/eva-icons';
import { FormsModule } from '@angular/forms';
import { MomentPipe } from './shared/pipes/moment.pipe';
import { PostDetailComponent } from './pages/post-detail/post-detail.component';
import { ApiVkService } from './shared/api-vk.service';
import { GroupInfoComponent } from './pages/home/group-info/group-
info.component';
import { SearchGroupComponent } from './pages/home/search-group/search-
group.component';
import { WallListComponent } from './pages/home/wall-list/wall-list.component';
import { ChartsModule } from 'ng2-charts';
import { AnalysisCommentComponent } from './pages/analysis/analysis-
comment/analysis-comment.component';
import { AnalysisLikeComponent } from './pages/analysis/analysis-like/analysis-
like.component';
import { AnalysisTextComponent } from './pages/analysis/analysis-text/analysis-
text.component';
import { ShowWordsPipe } from './shared/pipes/show-words.pipe';

@NgModule({
  declarations: [
    AppComponent,
    AuthorComponent,
    AboutComponent,
    NewsComponent,
    HomeComponent,
```

## Б қосымшасының жалғасы

```
MomentPipe,  
ShowWordsPipe,  
PostDetailComponent,  
GroupInfoComponent,  
SearchGroupComponent,  
WallListComponent,  
AnalysisCommentComponent,  
AnalysisLikeComponent,  
AnalysisTextComponent,  
],  
imports: [  
  HttpClientModule,  
  BrowserModule,  
  RouterModule.forRoot([  
    { path: '', pathMatch: 'full', redirectTo: 'home' },  
    { path: 'home', component: HomeComponent },  
    { path: 'news', component: NewsComponent },  
    { path: 'about', component: AboutComponent },  
    { path: 'author', component: AuthorComponent },  
    { path: 'post/:id', component: PostDetailComponent },  
  ]),  
  BrowserModuleAnimationsModule,  
  NbThemeModule.forRoot({ name: 'default' }),  
  NbLayoutModule,  
  NbEvalIconsModule,  
  FormsModule,  
  ChartsModule  
],  
providers: [  
  ApiService,  
],  
bootstrap: [AppComponent]  
)  
export class AppModule { }  
  
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']
```

## Б қосымшасының жалғасы

```
})
export class AppComponent {}

import {Injectable} from '@angular/core';
import {HttpClient, HttpHeaders} from '@angular/common/http';
import {Observable, of} from 'rxjs';
import * as queryString from 'querystring';
import {catchError} from 'rxjs/operators';

@Injectable()
export class ApiVkService {

  public token = null;
  private url = 'https://api.vk.com';
  private versionApi = '5.101';
  private clientId = '7151733';

  constructor(
    private http: HttpClient
  ) {
    const token = localStorage.getItem('token');
    if (token) {
      this.token = token;
    }
  }

  groupsGetById(groupId: string, fields: string = "") {
    const params = {
      group_id: groupId,
      fields,
    };
    return this.getMethod('groups.getById', params);
  }

  wallGet(domain: string, count: number): Observable<any> {
    const params = {
      domain,
      count,
    };
    return this.getMethod('wall.get', params);
  }

  wallGetById(post: string) {
```

## Б қосымшасының жалғасы

```
const params = {
  posts: post
};
return this.getMethod('wall.getById', params);
}
```

```
wallGetComments(ownerId: number, postId: number, count: number = 10) {
  const params = {
    owner_id: ownerId,
    post_id: postId,
    count,
    thread_items_count: 10,
  };
  return this.getMethod('wall.getComments', params);
}
```

```
usersGet(userIds: number, fields: string) {
  const params = {
    user_ids: userIds,
    fields,
  };
  return this.getMethod('users.get', params);
}
```

```
likesGetList(type: string, ownerId: number, itemId: number, extended: number = 0,
count: number = 100) {
  const params = {
    type,
    owner_id: ownerId,
    item_id: itemId,
    extended,
    count,
  };
  return this.getMethod('likes.getList', params);
}
```

```
private getMethod(method: string, params) {
```

## Б қосымшасының жалғасы

```
if (!this.token) {
  alert('У вас нет токена.');
```

```
  return;
}
params.access_token = this.token;
params.v = this.versionApi;

return this.http.get(`${this.url}/method/${method}?`, { params })
  .pipe(
    catchError(() => of([]))
  );
}

getToken(): Observable<string> {
  const params = {
    client_id: this.clientId,
    display: 'page',
    redirect_uri: 'http://localhost:4200',
    response_type: 'token',
    v: this.versionApi
  };
  return of('http://oauth.vk.com/authorize?' + queryString.stringify(params));
}

getBadWords(): Observable<string[]> {
  return this.http.get<string[]>('./assets/bad-words.json');
}

getGoodWords(): Observable<string[]> {
  return this.http.get<string[]>('./assets/good-words.json');
}
}

import {Pipe, PipeTransform} from '@angular/core';

@Pipe({
  name: 'showWords'
})
export class ShowWordsPipe implements PipeTransform {
  transform(value: string[], ...args: any[]): any {
    console.log(value);
  }
}
```

## Б қосымшасының жалғасы

```
    return value.join(',');
  }
}

import { Pipe, PipeTransform } from '@angular/core';
import * as moment from 'moment';

@Pipe({
  name: 'momentPipe'
})
export class MomentPipe implements PipeTransform {

  transform(value: any, ...args: any[]): any {
    moment.locale('ru');
    return moment.unix(value).calendar();
  }

}

import { Component, OnInit } from '@angular/core';
import { ApiVkService } from '../shared/api-vk.service';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  public postData: any = null;
  public domain = "";
  public count = 10;
  public groupInfo: any = null;
  public error = null;

  constructor(private apiVk: ApiVkService) { }

  ngOnInit() {}

  getGroup(domain: string) {
    this.error = "";
  }
}
```

## Б қосымшасының жалғасы

```
this.domain = domain;
const fields = 'description,counters,contacts,wall,verified,status';
this.apiVk.groupsGetById(this.domain, fields).subscribe((group: any) => {
  if (group && group.error) {
    this.error = group.error.error_msg;
  } else if (group && group.response) {
    this.groupInfo = group.response[0];
  } else {
    this.error = 'Нет данных';
  }
});
}

getPosts() {
  this.apiVk.wallGet(this.domain, this.count)
    .subscribe(res => this.postData = res);
}

reset() {
  this.groupInfo = "";
  this.postData = null;
}
}

import {Component, EventEmitter, Input, OnInit, Output} from '@angular/core';
import {ApiVkService} from '../shared/api-vk.service';

@Component({
  selector: 'app-search-group',
  templateUrl: './search-group.component.html',
  styleUrls: ['./search-group.component.css']
})
export class SearchGroupComponent implements OnInit {

  @Input() groupInfo: any;
  @Output() searched = new EventEmitter();
  domain = "";
  tokenLink = "";
  tokenValue = "";

  constructor(private apiVk: ApiVkService) { }
```

## Б қосымшасының жалғасы

```
ngOnInit() {  
}
```

```
getToken() {  
  this.apiVk.getToken().subscribe(e => this.tokenLink = e);  
}
```

```
saveToken() {  
  localStorage.setItem('token', this.tokenValue);  
  this.apiVk.token = this.tokenValue;  
  this.tokenValue = "";  
  this.tokenLink = "";  
  alert('Токен успешно сохранен');  
}
```

```
<div class="jumbotron align-items-center">  
  <div class="col-sm-8 mx-auto align-items-center">  
      
    <h1 class="display-4 text-center title">Вставьте короткий адрес сообщества для  
получения информацию о сообществе.</h1>  
    <form>  
      <div class="form-group">  
        <input  
          [(ngModel)]="domain"  
          name="domain"  
          type="text"  
          class="form-control"  
          id="UrlPost"  
          aria-describedby="emailHelp"  
          placeholder="К примеру: znews kz"  
        >  
      </div>  
      <div class="row">  
        <div class="col-sm-12">  
          <div class="text-center">  
            <button type="submit" class="btn btn-primary btn-lg"  
(click)="searched.emit(domain)">Найти сообщество</button>  
          <br>  
          <br>  
        </div>  
      </div>  
    </div>  
  </div>
```

## Б қосымшасының жалғасы

```
    <button type="button" class="btn btn-primary" (click)="getToken()">Get
token</button>
  </div>
  <div class="col-md-12" *ngIf="tokenLink">
    Token link: <a [href]="tokenLink" target="_blank">get token</a>

    <div class="form-group">
      <input class="form-control" type="text" placeholder="add token"
[(ngModel)]="tokenValue" name="tokenValue">
      <button (click)="saveToken()">save</button>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```

```
import {Component, Input, OnInit} from '@angular/core';
```

```
@Component({
  selector: 'app-group-info',
  templateUrl: './group-info.component.html',
  styleUrls: ['./group-info.component.css']
})
export class GroupInfoComponent implements OnInit {

  public pieChartLabels: string[] = ['Альбом', 'Статья', 'Аудио', 'Фото', 'Топики',
'Видео'];
  public pieChartData: number[] = [];
  public pieChartType = 'pie';

  @Input() groupInfo: any;

  constructor() { }

  ngOnInit() {
    this.pieChartData = this.getDataForChart();
  }

  private getDataForChart() {
```

## Б қосымшасының жалғасы

```
return [  
  this.groupInfo.counters.albums,  
  this.groupInfo.counters.articles,  
  this.groupInfo.counters.audios,  
  this.groupInfo.counters.photos,  
  this.groupInfo.counters.topics,  
  this.groupInfo.counters.videos,  
  ];  
}  
  
}  
  
import {Component, Input, OnInit} from '@angular/core';  
import {concatMap, delay, map, mergeMap} from 'rxjs/operators';  
import {from, of} from 'rxjs';  
import {ApiVkService} from '../shared/api-vk.service';  
  
@Component({  
  selector: 'app-analysis-comment',  
  templateUrl: './analysis-comment.component.html',  
  styleUrls: ['./analysis-comment.component.css']  
})  
export class AnalysisCommentComponent implements OnInit {  
  
  @Input() postData;  
  isVisible = true;  
  
  isAnalysisComment = false;  
  analysisCommentData = {  
    ids: [],  
    commentators: [],  
    girls: 0,  
    boys: 0,  
  };  
  
  isAnalysisCity = false;  
  cities = [];  
  
  isAnalysisAge = false;  
  ages = [];
```

## Б қосымшасының жалғасы

```
groupAgesByCount = [];  
groupCitiesByCount = [];  
  
public sexPieChartLabels: string[] = ['Мужчины', 'Женщины'];  
public sexPieChartData: number[] = [];  
  
public cityPieChartLabels: string[] = [];  
public cityPieChartData: number[] = [];  
  
public agePieChartLabels: string[] = [];  
public agePieChartData: number[] = [];  
  
constructor(private apiVk: ApiVkService) {}  
  
ngOnInit() {}  
  
analysisCommentators() {  
  
    const girls = this.analysisCommentData.commentators.filter(e => e.sex === 1);  
    const boys = this.analysisCommentData.commentators.filter(e => e.sex === 2);  
    this.analysisCommentData.girls = girls.length;  
    this.analysisCommentData.boys = boys.length;  
  
    this.sexPieChartData = [boys.length, girls.length];  
    this.isAnalysisComment = true;  
    console.log('users: ', this.analysisCommentData.commentators);  
}  
  
analysisCity() {  
    this.cities = [];  
    this.analysisCommentData.commentators.forEach((user: any) => {  
        if (user.city) {  
            const findIndex = this.cities.findIndex(e => e.title === user.city.title);  
            if (findIndex > -1) {  
                this.cities[findIndex].count++;  
            } else {  
                this.cities.push({  
                    title: user.city.title,  
                    count: 1,  
                });  
            }  
        }  
    })  
}
```

## Б қосымшасының жалғасы

```
    }
  });
  this.cities.forEach(city => {
    this.cityPieChartLabels.push(city.title);
    this.cityPieChartData.push(city.count);
  });
  this.isAnalysisCity = true;
  this.cities.sort(this.sortByCount);
  // группировка по количеству
  this.groupCitiesByCount = this.groupCities();
}

analysisAge() {
  this.ages = [];
  const bdateHave = this.analysisCommentData.commentators.filter(user =>
user.bdate);
  bdateHave.forEach(user => {
    const bdateArr = user.bdate.split('.');
    if (bdateArr.length === 3) {
      const ageUser = 2020 - (+bdateArr[2]);
      const findIndex = this.ages.findIndex(data => data.age === ageUser);
      if (findIndex > -1) {
        this.ages[findIndex].count++;
      } else {
        this.ages.push({
          age: ageUser,
          count: 1,
        });
      }
    }
  });
  this.ages.forEach(age => {
    this.agePieChartLabels.push(age.age);
    this.agePieChartData.push(age.count);
  });
  this.isAnalysisAge = true;
  this.ages.sort(this.sortByCount);
  // группировка по количеству
  this.groupAgesByCount = this.groupAges();
}
```

## Б қосымшасының жалғасы

```
getCommentators() {
  this.analysisCommentData.ids = [];
  this.analysisCommentData.commentators = [];

  this.apiVk.wallGetComments(
    this.postData.owner_id,
    this.postData.id,
    this.postData.comments.count
  ).pipe(
    map((e: any) => e.response.items),
    map((items: any) => {
      const ids = [];
      items.forEach(item => {
        ids.push(item.from_id);
        if (item.thread.items.length) {
          item.thread.items.forEach(e => ids.push(e.from_id));
        }
      });
      this.analysisCommentData.ids = this.unique(ids);
      return this.analysisCommentData.ids;
    }),
    mergeMap((userIds: any) => this.getUser(userIds)),
  ).subscribe((res: any) => {
    this.analysisCommentData.commentators.push(res);
  });
}

private getUser(userIds: number[]) {
  if (userIds.length) {
    return from(userIds).pipe(
      concatMap(userId => {
        return this.apiVk.usersGet(userId, 'sex,city,bdate').pipe(
          delay(300),
        );
      }),
      map((res: any) => (res && res.response && res.response.length) ?
res.response[0] : of(null))
    );
  } else {
    return of([]);
  }
}
```

## Б қосымшасының жалғасы

```
}

private unique(arr): any[] {
  const result = [];
  for (const str of arr) {
    if (!result.includes(str)) {
      result.push(str);
    }
  }
  return result;
}

toggleVisible() {
  this.isVisible = !this.isVisible;
}

private sortByCount(a: any, b: any) {
  return b.count - a.count;
}

private groupAges() {
  let grouped = [];
  let ages = JSON.parse(JSON.stringify(this.ages));
  ages.forEach(element => {
    if (ages.length) {
      const filter = ages.filter(e => e.count === element.count);
      grouped.push({
        count: element.count,
        data: filter,
      });
      ages = ages.filter(a => a.count !== element.count);
    }
  });
  grouped = grouped.filter(a => a.data.length);
  return grouped;
}

private groupCities() {
  let grouped = [];
  let cities = JSON.parse(JSON.stringify(this.cities));
  cities.forEach(element => {
    if (cities.length) {
```

## Б қосымшасының жалғасы

```
const filter = cities.filter(e => e.count === element.count);
grouped.push({
  count: element.count,
  data: filter,
});
cities = cities.filter(a => a.count !== element.count);
}
});
grouped = grouped.filter(a => a.data.length);
return grouped;
}
}
```

```
import { Component, OnInit, Input } from '@angular/core';
import { ApiVkService } from 'src/app/shared/api-vk.service';
import { map, mergeMap } from 'rxjs/operators';
```

```
@Component({
  selector: 'app-analysis-text',
  templateUrl: './analysis-text.component.html',
  styleUrls: ['./analysis-text.component.css']
})
export class AnalysisTextComponent implements OnInit {
```

```
  @Input() postData;
  isVisible = true;
```

```
  wordStat = [];
  wordLength = 0;
```

```
  isAnalysisPostText = false;
```

```
  groupWordsByCount = [];
```

```
  badWords = [];
  goodWords = [];
```

```
  constructor(private apiVk: ApiVkService) { }
```

## Б қосымшасының жалғасы

```
ngOnInit() { }

toggleVisible() {
  this.isVisible = !this.isVisible;
}

analysisPostText() {
  this.isAnalysisPostText = true;
  this.wordStat = [];
  const words = this.postData.text.replace (/r\n?|n/g, ' ').replace (/ {2,}/g, ' ').replace (/^ /, "").replace (/ $/, "").split(" ");
  this.__analyzeText(words);
}

analysisTextComments() {
  this.apiVk.wallGetComments(
    this.postData.owner_id,
    this.postData.id,
    this.postData.comments.count
  )
  .pipe(
    map((response: any) => response && response.response && response.response.items ? response.response.items : []),
  )
  .subscribe((items: any) => {
    if (items && items.length) {
      const words = items.map(item => item.text).join(' ')
        .replace (/r\n?|n/g, ' ').replace (/ {2,}/g, ' ').replace (/^ /, "").replace (/ $/, "").split(' ');
      this.__analyzeText(words);
      this.analysisTextCommentsNegative(items.map(item => item.text));
      this.isAnalysisPostText = true;
    }
  });
}

private async analysisTextCommentsNegative(data: string[]) {
  const text = data.join(' ')
    .replace (/r\n?|n/g, ' ')
    .replace (/ {2,}/g, ' ')
    .replace (/^ /, ")
```

## Б қосымшасының жалғасы

```
.replace (/ $/, ")
.replace (/,/g, ")
.replace (\/./g, ")
.split(' ');
const badWordsD = await this.apiVk.getBadWords().toPromise();
const goodWordsD = await this.apiVk.getGoodWords().toPromise();

text.forEach(item => {
  const word = item.toLowerCase();
  if (badWordsD.includes(word)) {
    this.badWords.push(word);
  } else if (goodWordsD.includes(word)) {
    this.goodWords.push(word);
  }
});
console.log(this.badWords);
console.log(this.goodWords);
}

private __analyzeText(words: string[]) {
  this.wordStat = [];
  this.wordLength = words.length;
  words.forEach(word => {
    if (word.length > 3) {
      const i = this.wordStat.findIndex(e => e.title === word);
      if (i > -1) {
        this.wordStat[i].count++;
      } else {
        this.wordStat.push({
          title: word,
          count: 1,
        });
      }
    }
  });
  this.wordStat.sort((a: any, b: any) => {
    return b.count - a.count;
  });
  // группировка по количеству
  this.groupWords();
}
```

## Б қосымшасының жалғасы

```
private groupWords() {
  this.groupWordsByCount = [];
  let items = JSON.parse(JSON.stringify(this.wordStat));
  items.forEach(element => {
    if (items.length) {
      const filter = items.filter(e => e.count === element.count);
      this.groupWordsByCount.push({
        count: element.count,
        data: filter,
      });
      items = items.filter(a => a.count !== element.count);
    }
  });
  this.groupWordsByCount = this.groupWordsByCount.filter(a => a.data.length);
}
}
```

```
import { Component, OnInit, Input } from '@angular/core';
import { ApiVkService } from 'src/app/shared/api-vk.service';
import { from, of } from 'rxjs';
import { concatMap, delay, map, mergeMap } from 'rxjs/operators';
```

```
@Component({
  selector: 'app-analysis-like',
  templateUrl: './analysis-like.component.html',
  styleUrls: ['./analysis-like.component.css']
})
export class AnalysisLikeComponent implements OnInit {
```

```
  @Input() postData;
  isVisible = true;
```

```
  users = [];
  sex = {
    girls: 0,
    boys: 0,
  };
  cities = [];
  ages = [];
```

## Б қосымшасының жалғасы

```
groupAgesByCount = [];  
groupCitiesByCount = [];  
  
sexPieChartData = [];  
sexPieChartLabels = ['Мужчины', 'Женщины'];  
  
cityPieChartLabels: string[] = [];  
cityPieChartData: number[] = [];  
  
agePieChartLabels: string[] = [];  
agePieChartData: number[] = [];  
  
isAnalysisSex = false;  
isAnalysisCity = false;  
isAnalysisAge = false;  
  
constructor(private apiVk: ApiVkService) { }  
  
ngOnInit() {  
  
}  
  
toggleVisible() {  
  this.isVisible = !this.isVisible;  
}  
  
likesGetList() {  
  this.users = [];  
  this.apiVk.likesGetList(  
    this.postData.post_type,  
    this.postData.owner_id,  
    this.postData.id  
  ).pipe(  
    mergeMap((res: any) => {  
      if (res.response && res.response.items && res.response.items.length) {  
        return this.getUser(res.response.items);  
      }  
      return of(null);  
    }  
  ),  
  ).subscribe((user: any) => {  
    this.users.push(user);  
  });  
}
```

## Б қосымшасының жалғасы

```
});  
}  
  
private getUser(userIds: number[]) {  
  if (userIds.length) {  
    return from(userIds).pipe(  
      concatMap(userId => {  
        return this.apiVk.usersGet(userId, 'sex,city,bdate').pipe(  
          delay(300),  
        );  
      }),  
      map((res: any) => (res && res.response && res.response.length) ?  
res.response[0] : of(null))  
    );  
  }  
  return of([]);  
}  
  
analysisSex() {  
  const girls = this.users.filter(e => e.sex === 1);  
  const boys = this.users.filter(e => e.sex === 2);  
  this.sex.girls = girls.length;  
  this.sex.boys = boys.length;  
  
  this.sexPieChartData = [boys.length, girls.length];  
  this.isAnalysisSex = true;  
}  
  
analysisCity() {  
  this.cities = [];  
  this.users.forEach((user: any) => {  
    if (user.city) {  
      const findIndex = this.cities.findIndex(e => e.title === user.city.title);  
      if (findIndex > -1) {  
        this.cities[findIndex].count++;  
      } else {  
        this.cities.push({  
          title: user.city.title,  
          count: 1,  
        });  
      }  
    }  
  });  
}
```

## Б қосымшасының жалғасы

```
    }
  });
  this.cities.forEach(city => {
    this.cityPieChartLabels.push(city.title);
    this.cityPieChartData.push(city.count);
  });
  this.isAnalysisCity = true;
  this.cities.sort(this.sortByCount);
  // группировка по количеству
  this.groupCitiesByCount = this.groupCities();
}

analysisAge() {
  this.ages = [];
  const bdateHave = this.users.filter(user => user.bdate);
  bdateHave.forEach(user => {
    const bdateArr = user.bdate.split('.');
    if (bdateArr.length === 3) {
      const ageUser = 2020 - (+bdateArr[2]);
      const findIndex = this.ages.findIndex(data => data.age === ageUser);
      if (findIndex > -1) {
        this.ages[findIndex].count++;
      } else {
        this.ages.push({
          age: ageUser,
          count: 1,
        });
      }
    }
  });
  this.ages.forEach(age => {
    this.agePieChartLabels.push(age.age);
    this.agePieChartData.push(age.count);
  });
  this.isAnalysisAge = true;
  this.ages.sort(this.sortByCount);
  // группировка по количеству
  this.groupAgesByCount = this.groupAges();
}

private groupAges() {
```

## Б қосымшасының жалғасы

```
let grouped = [];  
let ages = JSON.parse(JSON.stringify(this.ages));  
ages.forEach(element => {  
  if (ages.length) {  
    const filter = ages.filter(e => e.count === element.count);  
    grouped.push({  
      count: element.count,  
      data: filter,  
    });  
    ages = ages.filter(a => a.count !== element.count);  
  }  
});  
grouped = grouped.filter(a => a.data.length);  
return grouped;  
}  
  
private groupCities() {  
  let grouped = [];  
  let cities = JSON.parse(JSON.stringify(this.cities));  
  cities.forEach(element => {  
    if (cities.length) {  
      const filter = cities.filter(e => e.count === element.count);  
      grouped.push({  
        count: element.count,  
        data: filter,  
      });  
      cities = cities.filter(a => a.count !== element.count);  
    }  
  });  
  grouped = grouped.filter(a => a.data.length);  
  return grouped;  
}  
  
private sortByCount(a: any, b: any) {  
  return b.count - a.count;  
}  
}
```