In [2]:

```
import nltk
import itertools
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from pymystem3 import Mystem
from kneed import KneeLocator
from string import punctuation
from wordcloud import WordCloud
from nltk.corpus import stopwords
from sklearn.cluster import KMeans
from collections import Counter, defaultdict
from sklearn.metrics import silhouette_score
from sklearn.decomposition import TruncatedSVD
from sklearn.utils.extmath import randomized_svd
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [3]:

```
nltk.download('stopwords')

# get russian stop words
russian_stopwords = stopwords.words("russian")
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\arkhipkin.ma\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Prepocessing data

In [4]:

```
# abbreviations
dic_abbrev = {
    'вуз': 'высшее учебное заведение',
    'азс': 'авто запровочная станция',
    'жби': 'железнобетонные изделия',
    'загс': 'записи актов гражданского состояния',
    'нии': 'научно иследовательский институт',
    'бти': 'бюро технических инвестиций',
    'тсж': 'товарищество собственников жилья',
    'мфц': 'много функциональный центр',
    'мрэо': 'межрайонный регистрационно экзаменационный отдел'}
```

In [5]:

```
df = pd.read_csv('yandex_output.csv')

# replace NaN values
df.fillna('', inplace=True)

# remove uninformative features
df.drop(['company_url', 'company_links', 'source', 'company_name', 'company_features'], a
xis=1, inplace=True)

# to lower case
df.company_categories = df.company_categories.apply(lambda x: x.lower())

# remove duplicates
df.drop_duplicates(['phone_number', 'company_categories'], keep='last', inplace=True)

# convert df to dict(for convenience)
df_to_dict = df.to_dict()
```

```
categories = list(df_to_dict['company_categories'].values())

# add column 'category'
df['category'] = int(0)

df.phone_number = df.phone_number.astype(np.uint64)
df.category = df.category.astype(np.uint16)
```

In [6]:

```
# tokenization for all categories
company_categories = [x.split(';') for x in categories]

# frequency of categories in the form of dict
categories_dict = Counter(itertools.chain.from_iterable(company_categories))

# all categories
categories_List = list(itertools.chain.from_iterable(company_categories))
```

In [7]:

```
# replace abbreviations
for i, val in enumerate(categories_List):
    if val in dic_abbrev.keys():
        categories_List[i] = dic_abbrev.get(val)
```

In [9]:

```
# dfs for vizualize
df_categories = pd.DataFrame.from_dict(categories_dict, orient='index').reset_index()
df_categories.columns = ['words', 'count']
```

In [10]:

```
def preprocess_text(text):
    mystem = Mystem()
    tokens = mystem.lemmatize(text)
    tokens = [token for token in tokens if token not in russian_stopwords\
              and token != " " \
              and token.strip() not in punctuation]
    text = " ".join(tokens)
    return text
```

In [ ]:

```
# let's lemmatize
processed_text = []

for i in categories_List:
    processed_text.append(preprocess_text(i))
```

Building model

In [12]:

```
# apply tf-idf vectoriser
vectorizer = TfidfVectorizer(stop_words=russian_stopwords,
                             max_df=0.5,
                             use_idf=True,
                             ngram_range=(1, 5))
X = vectorizer.fit_transform(processed_text)

terms = vectorizer.get_feature_names()
```

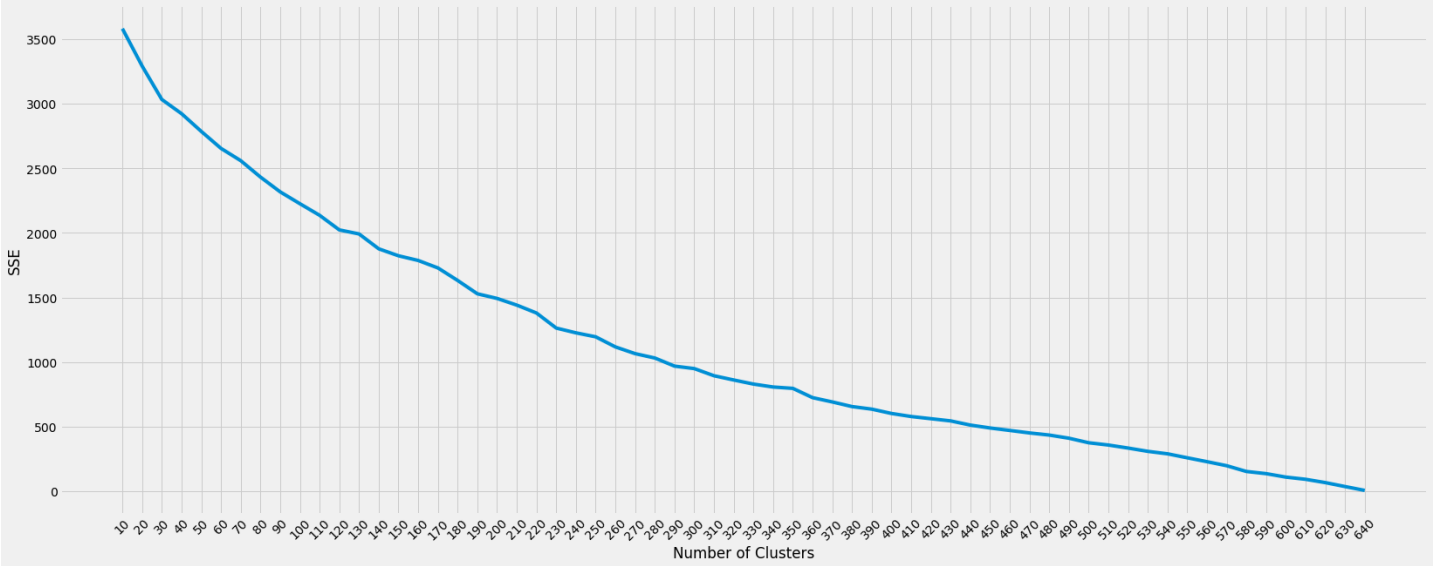
In [13]:

```
# use k-means algorithm
# find 'k' in for loop use Elbow Method
```

```
kmeans_kwargs = {
    "init": "random",
    "n_init": 10,
    "max_iter": 300,
    "random_state": 42,
}
sse = []
max_kmeans = np.unique(processed_text).shape[0]

for k in range(10, max_kmeans, 10):
    kmeans = KMeans(n_clusters=k, **kmeans_kwargs)
    kmeans.fit(X)
    sse.append(kmeans.inertia_)
```

In [21]:

```
# visualize the SSE graph

plt.rcParams["figure.figsize"] = (25,10)

plt.style.use("fivethirtyeight")
plt.plot(range(10, max_kmeans, 10), sse)
plt.xticks(range(10, max_kmeans, 10))
plt.xlabel("Number of Clusters")
plt.xticks(rotation=45)
plt.ylabel("SSE")
plt.show()
```



In [16]:

```
# use the kneed package for to indentify the elbow point programmatically

kl = KneeLocator(range(10, max_kmeans, 10), sse, curve="convex", direction="decreasing")
print(f'Best kmean: {kl.elbow}')
```

```
Best kmean: 290
```

In [25]:

```
# let's look at the Silhouette analysis
# it takes a value from -1 to 1

km = KMeans(n_clusters=kl.elbow)
km.fit(X)
km_silhouette = silhouette_score(X, km.labels_).round(2)

# get glass numbers
clusters = km.labels_.tolist()

print(f'Silhouette analysi for {kl.elbow} clusters: {km_silhouette}')
```

```
Silhouette analysi for 290 clusters: 0.83
```

In [28]:

```
# to represent each and every term and document as a vector
# VT - is a term-concept matrix
# Sigma - is concept-concept matrix
# U - is document-concept matrix

U, Sigma, VT = randomized_svd(X,
                              n_components=kl.elbow,
                              n_iter=300,
                              random_state=42)

# let's look at the first 10 grades and top keaywords
for i, comp in enumerate(VT[:10]):
    terms_comp = zip(terms, comp)
    sorted_terms = sorted(terms_comp, key=lambda x: x[1], reverse=True)[:10]
    print("Category: " + str(i))
    for t in sorted_terms:
        print(t[0], end=', ')
    print('\n')
```

```
Category: 0
клиника, медцентр, медцентр клиника, стоматологический клиника, стоматологический, ветери
нарный клиника, ветеринарный, гинекологический, гинекологический клиника, наркологический
,

Category: 1
автозапчасть автотовары, автотовары, магазин автозапчасть, магазин автозапчасть автотовар
ы, магазин, автозапчасть, интернет магазин, интернет, магазин мебель, мебель,

Category: 2
автосервис, автосервис автотехцентр, автотехцентр, такси, стоматологический клиника, стом
атологический, клиника, ветеринарный, компания, ветеринарный клиника,

Category: 3
поликлиника, взрослый, поликлиника взрослый, больница, детский, больница взрослый, детски
й поликлиника, детский больница, специализированный больница, специализированный,

Category: 4
такси, поликлиника, взрослый, поликлиника взрослый, детский, детский поликлиника, больниц
а взрослый, больница, центр, диагностический,

Category: 5
грузоперевозка, автомобильный грузоперевозка, автомобильный, железнодорожный грузоперевоз
ка, железнодорожный, автомобильный диск, автомобильный диск шина, диск шина, шина, диск,

Category: 6
шиномонтаж, центр, диагностический, диагностический центр, служба, коммунальный, коммунал
ьный служба, такси, аптека, отделение,

Category: 7
центр, диагностический, диагностический центр, реабилитационный, реабилитационный центр,
оздоровительный, оздоровительный центр, торговый центр, торговый, развлекательный,

Category: 8
больница, больница взрослый, взрослый, специализированный больница, специализированный, д
етский больница, медцентр, медцентр клиника, автозапчасть, автозапчасть автотовары,

Category: 9
салон, красота, салон красота, массажный салон, массажный, спа, спа салон, салон связь, с
вязь, материал салон,
```

In [31]:

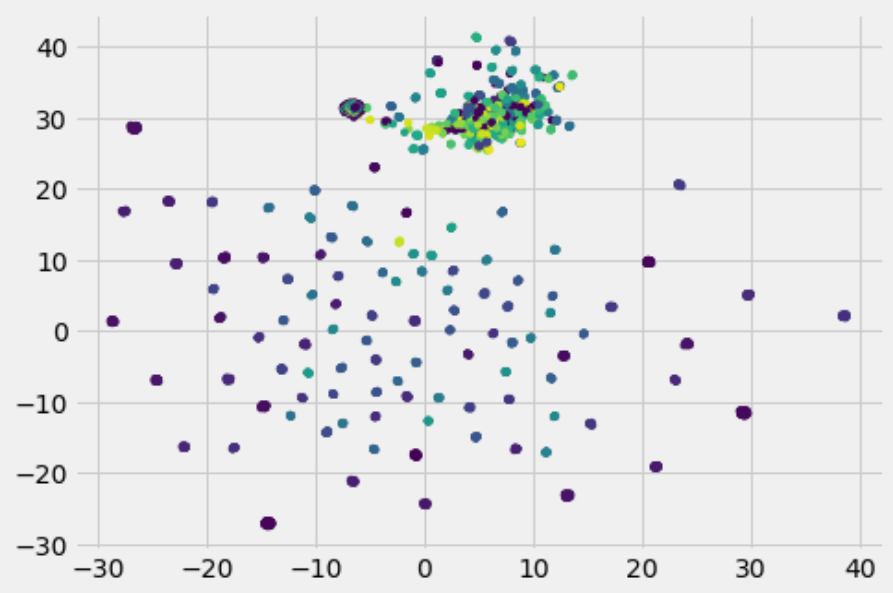
```
# assign the predicted class number to each
# category in the original data set

lst_tmp = []
```

```
for index, value in df_to_dict['company_categories'].items():
    for word in value.split(';'):
        dct_tmp = {}
        dct_tmp['category'] = index
        dct_tmp['word'] = word
        lst_tmp.append(dct_tmp)
df_tmp = pd.DataFrame(lst_tmp)

for category, class_ in zip(categories_List, clusters):
    if len(df_tmp[df_tmp.word == str(category)]) != 0:
        for ind in df_tmp[df_tmp.word == str(category)]['category'].tolist():
            df.loc[ind, 'category'] = class_
```

Vizualization

In [95]:

```
# use UMAP method to petform nonlinear dimensionality reduction

import umap.umap_ as umap

X_topics=U*Sigma
embedding = umap.UMAP(n_neighbors=290, min_dist=0.5, random_state=12).fit_transform(X_top
ics)

plt.figure(figsize=(7,5))
plt.scatter(embedding[:, 0], embedding[:, 1], c=clusters, s=20, edgecolor='none')
plt.show()
```

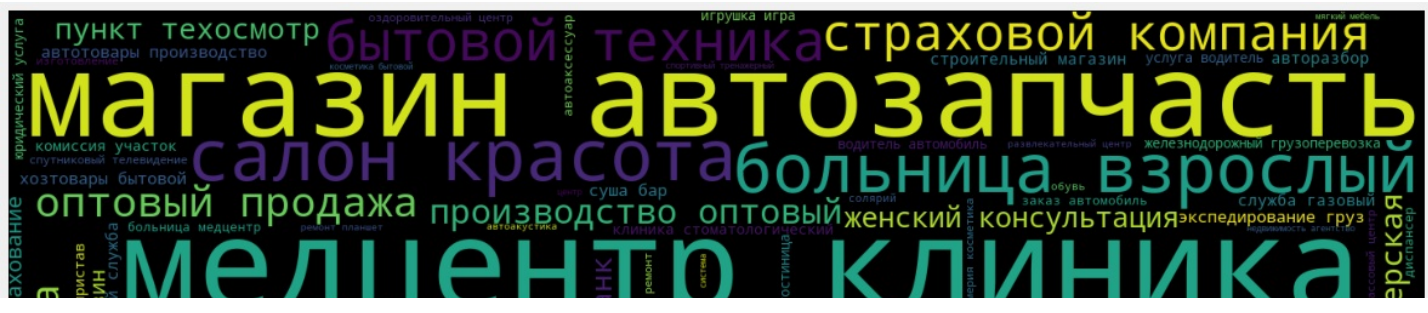


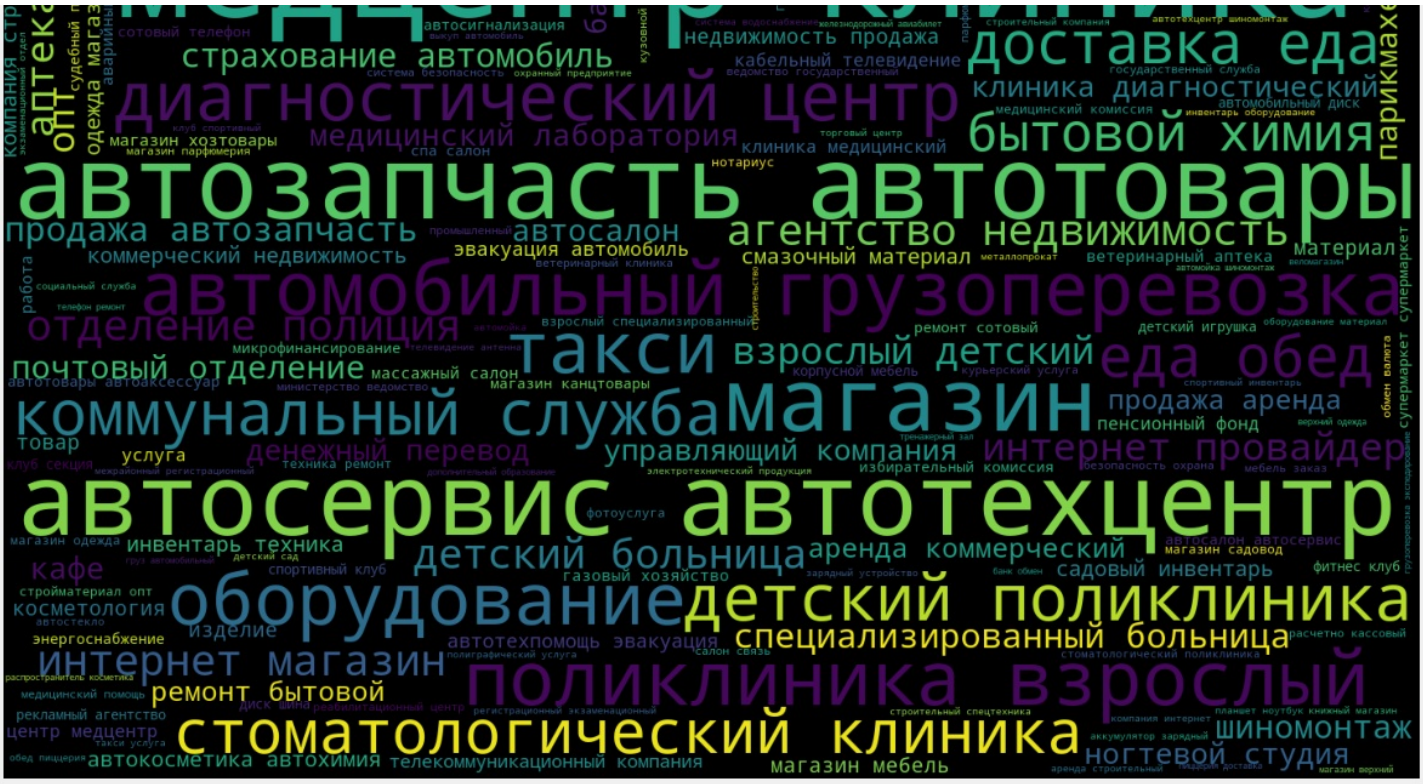
In [27]:

```
# plot tag cloud
cloud = WordCloud(width=1440, height=1080).generate(" ".join(processed_text))
plt.figure(figsize=(20, 15))
plt.imshow(cloud)
plt.axis('off')
```

Out[27]:

```
(-0.5, 1439.5, 1079.5, -0.5)
```





In [41]:

```
# print top 5 categories and their counts
df.groupby('category').count()['phone_number'].reset_index().sort_values(by='phone_numbe
r', ascending=False).head()
```

Out[41]:

	category	phone_number
6	6	114
111	160	98
83	109	90
64	79	58
149	262	51

In [62]:

```
# categories with one value
one_categories = df.groupby('category')
one_categories.filter(lambda x: len(x) == 1).sort_values(by=['category', 'phone_number']
)
```

Out[62]:

	company_categories	phone_number	category
1751	спецтехника и спецавтомобили;грузовые автомоби...	79133372333	24
1860	изготовление протезно-ортопедических изделий;о...	78432220442	26
1317	ремонт измерительных приборов	74742710072	30
1049	медико-социальная экспертиза	74732679426	39
429	ювелирные изделия оптом;металлургическое предп...	73912593333	43
1227	кассовые аппараты и расходные материалы;весы и...	79537065328	49
582	дом инвалидов и престарелых	73517772366	57
1650	справочник	78212400808	58
1135	пищевое сырьё	74742555658	73
150	управляющая компания;согласование перепланировок	78216789888	83

	company_categories	phone_number	category
1754	тепличное оборудование;полимерные материалы	79612201454	213
842	насосы, насосное оборудование;промышленное обо...	73512105633	119
891	автотранспортное предприятие, автобаза	79612612251	123
1110	автосалон;страхование автомобилей;автоломбард	79508111811	124
1276	спортивный комплекс	78314497799	128
1279	магазин электроники;ноутбуки и планшеты;компью...	79087765356	161
499	квесты	73422357820	162
643	цирк	74872311298	164
196	исправительное учреждение	73912214262	168
10	магазин радиодеталей;радиотелефонная и радиоте...	73912702737	174
949	автоматические двери и ворота;окна;алюминий, а...	79036500405	179
704	военная форма, камуфляж	74732608099	189
1371	аэропорт	73842390139	191
1858	курьерские услуги;почтовые услуги	78332714021	215
419	наркологическая клиника;психотерапевтическая п...	78127027072	223
626	техникум	73844530495	224
1397	оценочная компания;экспертиза	79029433408	234
1782	салон оптики;контактные линзы	74732611756	240
1418	дополнительное образование;центр развития ребенка	73517736282	245
411	редакция сми	73912266622	255
1478	дом отдыха;турбаза	79372757040	265
1063	зоопарк	73952664639	268
1236	магазин суши и азиатских продуктов	79509116891	274
1026	религиозное объединение	73843783556	277
1398	магазин обуви;обувная косметика	73512253407	283

In [97]:

```
df.to_excel('result.xlsx', index=False)
df.to_csv('result.csv', index=False)
```