II SHO‘BA. KORPUS LINGVISTIKASI

KORPUS MATNLARINI PYTHON TILI VOSITASIDA TEMATIK MODELLASHTIRISH

Botir Elov Boltayevich,
Texnika fanlari falsafa doktori, dotsent
elov@navoiy-uni.uz
ToshDO‘TAU

Narzillo Aloyev Raxmatilloyevich,
ToshDO‘TAU tayanch doktorant
vip.alayev@gmail.com

Annotatsiya. Tematik modellashtirish – bu hujjatlar to‘plamini tahlil qilish, ulardagi so‘z va so‘z birikmalarni aniqlash hamda hujjatlar to‘plamini eng yaxshi tavsiflovchi so‘z guruhlari va shunga o‘xshash frazalarni avtomatik ravishda klasterlashga asoslangan nazoratsiz mashinali o‘rganish usuli. Har kuni yuzlab, hatto minglab mijozlar bilan muloqot qiladigan kompaniyada, ijtimoiy tarmoqlardagi xabarlar, elektron pochta xabarlari, chatlar, ochiq so‘rovnomalar va boshqa shu turdagi ma'lumotlarini tahlil qilish ancha murakkab jarayon. Sun'iy intellekt vositalari yordamida matn tahlili tilni tabiiy ravishda qayta ishlash uchun turli xil usullar yoki algoritmlardan foydalanadi. Ulardan biri mavzu tahlili - matnlardan mavzularni avtomatik aniqlash uchun ishlatiladi. Mavzuni tahlil qilish modellaridan foydalangan holda, tashkilotdagi katta hajmdagi matnli ma’lumotlarni qayta ishlash vazifalari mashinalarga yuklatilishi mumkin. Agar mashina har kuni mijozlar katta sondagi so‘rovlarini saralab bera olsa, tashkilot xodimlarining qancha vaqt tejashi va muhimroq vazifalarga sarflashi mumkinligini ta`kidlash zarur. Ushbu maqolada mavzuni tahlil qilish usullarining zamonaviy usullari tavsifi, matematik asosi va Python tilidagi tadbig‘i ko‘rib chiqiladi. Tematik modellashtirish – bu “nazoratsiz” mashinali o‘rganish usuli bo‘lib, dataset treningini talab qilmaydi.

Annotation. Topic modeling is an unsupervised machine learning technique based on analyzing a set of documents, identifying words and phrases in them, and automatically clustering groups of words and similar phrases that best describe the set of documents. In a company that communicates with hundreds or even thousands of customers every day, analyzing social media messages, emails, chats, open surveys and other such data is a complex process. Text analysis using artificial intelligence tools uses different techniques or algorithms to naturally process language. One of them is topic analysis, which is used to automatically identify topics from texts. Using topic analysis models, large text data processing tasks in an organization can be outsourced to machines. It's worth noting how much time an organization's employees can save and spend on more important tasks if a machine

can sort through a large number of customer requests every day. This article describes modern methods of topic analysis, mathematical basis and implementation in Python language. Topic modeling is an "unsupervised" machine learning technique that does not require dataset training.

Аннотация. Тематическое моделирование – это метод машинного обучения, основанный на анализе набора документов, выявлении в них слов и фраз и автоматической кластеризации групп слов и похожих фраз, которые лучше всего описывают набор документов. В компании, которая ежедневно общается с сотнями или даже тысячами клиентов, анализ сообщений в социальных сетях, электронных писем, чатов, открытых опросов и других подобных данных является сложным процессом. Анализ текста с использованием инструментов искусственного интеллекта использует различные методы или алгоритмы для естественной обработки языка. Один из них – тематический анализ, который используется для автоматического определения тем из текстов. Используя модели тематического анализа, большие задачи по обработке текстовых данных в организации можно передать машинам. Стоит отметить, сколько времени сотрудники организации могут сэкономить и потратить на более важные задачи, если машина сможет обрабатывать большое количество запросов клиентов каждый день. В этой статье описаны современные методы тематического анализа, их математическая основа и реализация на языке Python. Тематическое моделирование – это метод машинного обучения, который не требует обучения набору данных.

Kalit so‘zlar: *Tematik modellashtirish, tematik modellar, Python, LSA, LSI, LDA, NMF, HDP, nazoratsiz mashinali o‘qitish usullari, til korpusi.*

Kirish

Tematik modellashtirish – bu hujjatlar to‘plami uchun klaster so‘zlarini aniqlash uchun matn ma’lumotlarini avtomatik ravishda tahlil qiladigan mashinani o‘rganish usuli. Bu usul “nazoratsiz” mashinali o‘rganish sifatida tanilgan bo‘lib, avvaldan odamlar tomonidan tasniflangan teglar yoki o‘quv ma'lumotlarining oldindan belgilangan ro‘yxatini talab qilmaydi [1].

Tematik modellashtirish o‘quv ma’lumotlarini talab qilmagani uchun, ushbu usul ma’lumotlarni tahlil qilishni boshlashning tez va oson yo‘lidir. Shu sababli, tashkilotlar o‘z faoliyalari samaradorligini oshirish maqsadida mavzuni tasniflash modelini o‘rganishga vaqt sarflashni afzal ko‘rishadi.

Bugungi kundagi tematik modellashtirishning ommabop algoritmlariga *yashirin semantik tahlil (Latent Semantic Analysis, LSA), yashirin semantik indeksatsiya (Latent Semantic Indexing, LSI), ierarxik Dirixle jarayoni*

(Hierarchical Dirichlet Process, HDP), yashirin Dirixle taqsimoti (Latent Dirichlet Allocation, LDA) va *manfiy bo‘lmagan matritsa faktorizatsiyasi (Non-negative Matrix factorization, NMF)* kabi usullarini misol sifatida keltirish mumkin [2,3,4]. Ular orasida LSA amalda aniqroq va samarali natijalarni ko‘rsatgan va shuning uchun keng miqyosida qo‘llaniladi [5,6]. Ushbu maqolada yuqorida keltirilgan tematik modellashtirish usullarining barchasini birma-bir ko‘rib chiqiladi va Python tili vositalari orqali tadbiq qilinadi.

Yashirin semantik tahlil (Latent Semantic Analysis, LSA) usuli

Yashirin semantik tahlil – bu hujjatlar to‘plamidagi yashirin munosabatlarni kam o‘lchamli fazoda aniqlashga harakat qiladigan matematik usul. LSA ma’nosi yaqin bo‘lgan so‘zlar o‘xshash matn qismlarida *paydo bo‘lishini taxmin (tarqatish gipotezasi)* qiladi.

Har bir paragrafda so‘zlar sonini o‘z ichiga olgan matritsa (satrlar noyob so‘zlarni, ustunlar esa har bir paragrafni ifodalaydi) matnning katta qismidan hosil qilinadi va o‘xshashlik strukturasini saqlab qolgan holda qator(satr)lar sonini kamaytirish uchun *singulyar qiymat dekompozitsiyasi (singular value decomposition, SVD)* deb ataladigan matematik usuldan foydalaniladi [3,7]. LSA usulida har bir hujjatni boshqalardan ajratilgan holda ko‘rib chiqish o‘rniga, munosabatlarni aniqlash uchun barcha hujjatlarni va ulardagi shartlarni ko‘rib chiqiladi.

SVDdan quyi darajali matritsaga yaqinlashish masalasini hal qilish uchun foydalanish mumkin. So‘ngra ushbu matrisadan *termin-hujjat matritsalarini* aniqash mumkin. Buning uchun quyidagi uch bosqichli amallarni bajarish lozim [8,9]:

1. Berilgan **C** uchun **SVD** matrisasini $\boldsymbol{C = U \sum V^T}$ ko‘rinishda hosil qilamiz.
2. $\sum$ diagonalidagi **r−k** eng kichik singulyar qiymatlarni nolga almashtirish natijasida hosil bo‘lgan $\boldsymbol{\sum k}$ matritsasini hosil qilish.
3. $\boldsymbol{C_k = U \sum kV^T}$ni **C** ga **k**-darajada yaqinlashtirish.

Bu yerda, **C** – *termin-hujjat matrisasi,* $\boldsymbol{U}$, $\sum$ va $\boldsymbol{V^T}$ qiymatlar **SVD** matrisalari.

Yuqorida keltirilgan nazariy ma’lumotlar asosida Python tili vositalari yordamida **LSI** uchun **scikit-learning** modulidan foydalanamiz.

Scikit-learning modulida LSA usuli uchun o‘lchamlarni kamaytirish SVD metodi yordamida amalga oshiriladi.

```
from sklearn.decomposition import NMF, LatentDirichletAllocation, TruncatedSVD
from sklearn.feature_extraction.text import CountVectorizer
NUM_TOPICS = 4
```

```
# Hujjatni tokenlar matritsasiga aylantirish
vectorizer = CountVectorizer(min_df=5, max_df=0.9,
                        stop  words='english', lowercase=True,
                        token  pattern='[a-zA-Z\-][a-zA-Z\-]{2,}')
data_vectorized = vectorizer.fit_transform(title)
# SVDdan foydalaniv Latent Semantic Indexing (LSI) modelni shakllantirish
lsi_model = TruncatedSVD(n_components=NUM_TOPICS)
lsi  Z = lsi  model.fit  transform(data  vectorized)
print(lsi_Z.shape)
```

_____

(3000, 4)

```
def print_topics(model, vectorizer, top_n=10):
   for idx, topic in enumerate(model.components_):
      print("Topic %d:" % (idx))
      print([(vectorizer.get  feature  names  out()[i], topic[i])
                  for i in topic.argsort()[:-top  n - 1:-1]])

print("LSI Model:")
print  topics(lsi  model, vectorizer)
print("=" * 20)
```

_____

LSI Model:

Topic 0:

[('new', 0.9423293654141083), ('york', 0.09876242707814353), ('samsung‘, 0.08674228078722532), ('study', 0.0842158513506398), ('says', 0.0774287430910753), ('google', 0.07544052179967121), ('apple', 0.06923485166706239), ('galaxy', 0.06400938270176393), ('report', 0.05169112595076477), ('cancer', 0.04687394655373266)]

Topic 1:

[('google', 0.6199432211963671), ('samsung‘, 0.4647407644932722), ('galaxy', 0.3972519251332346), ('says', 0.17182596426071345), ('apple', 0.14386302171460943), ('android', 0.1330713392779825), ('glass', 0.13117330837697994), ('tab', 0.1099072145269534), ('report', 0.10445918022761816), ('price', 0.08276757970152882)]

Topic 2:

[('samsung‘, 0.5275265496715218), ('galaxy', 0.459192654236729), ('tab', 0.11338869931862162), ('price', 0.08361848415269445), ('apple', 0.051806901002287985), ('note', 0.045800336897351294), ('specs', 0.04483063275553376), ('india', 0.042755633537156765), ('mini', 0.038437773223479306), ('features', 0.03624060134992984)]

Topic 3:

[('says', 0.7334439981745634), ('ebola', 0.2912417585498625), ('study', 0.2819044681057816), ('health', 0.12002575609088922), ('world', 0.11769503771591194), ('west', 0.11704154406432216), ('outbreak', 0.10062689666445736), ('virus', 0.09807160654546011), ('report', 0.09094477004970669), ('million', 0.08399790424182825)]

==========

Yuqoridagi ro‘yxatda LSI modeli bo‘yicha mavzular keltirilgan.

```
from sklearn.manifold import TSNE
# NLTK
from nltk.tokenize import RegexpTokenizer
from nltk.stem.snowball import SnowballStemmer
from nltk.corpus import stopwords
import re

# Visualizatsiya
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import matplotlib
%matplotlib inline
import seaborn as sns

# Bokeh
from bokeh.io import output  notebook
from bokeh.plotting import figure, show
from bokeh.models import HoverTool, CustomJS, ColumnDataSource, Slider
from bokeh.layouts import column
from bokeh.palettes import all  palettes
output_notebook()
```

Keyingi qadamda LSI modelini visualizatsiya qilamiz va model bo‘yicha bir-biriga yaqin so‘zlar va hujjatlarni ko‘rib chiqamiz.

```
import pandas as pd

from bokeh.io import push_notebook, show, output_notebook
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource, LabelSet
output  notebook()

svd = TruncatedSVD(n_components=100)
documents_2d = svd.fit_transform(data_vectorized)

df = pd.DataFrame(columns=['x', 'y', 'document'])
df['x'], df['y'], df['document'] = documents_2d[:,0], documents_2d[:,1], range(len(data))

source = ColumnDataSource(ColumnDataSource.from_df(df))
labels = LabelSet(x="x", y="y", text="document", y  offset=8,
                  text_font_size="8pt", text_color="#555555",
                  source=source, text_align='center')

plot = figure(width=600, height=600)
plot.circle("x", "y", size=12, source=source, line_color="black", fill_alpha=0.8)
plot.add_layout(labels)
show(plot, notebook_handle=True)
```

Xulosa. Ushbu maqolaning asosiy g‘oyasi keng qo‘llaniladigan tematik modellashtirish usullarini amalga oshirish va taqqoslasdan iborat. Olingan

natijalarga ko‘ra NMF usuli eng yuqori muvofiqlik ballini bergan bo‘lsa-da, LDA eng ko‘p qo‘llaniladigan usul hisoblanadi va izchil deb hisoblanadi. Chunki LDA usuli ko‘proq "muvofiq" mavzularni taqdim etishi mumkin. Mavzu ehtimoli har bir hujjatda o‘zgarmas bo‘lishi kerak bo‘lsa, NMF yaxshiroq ishlaydi. Boshqa tomondan, HDP usulidan kamroq foydalaniladi, chunki mavzular soni oldindan aniqlanmagan va shuning uchun ushbu usul kamdan-kam qo‘llaniladi. Albatta, olingan natijalar berilgan namunaviy korpus ma'lumotlariga bog‘liq.

Foydalanilgan adabiyotlar:

1. Elov B., Alayev R., Aloyev N. (2024). Tematik modellashtirishning zamonaviy usullari. Digital transformation and artificial intelligence, *2*(1), 8–16. Retrieved from https://dtai.tsue.uz/index.php/dtai/article/view/v2i12
2. Elov.B., Alayev N. Matnlarini tematik modellashtirish va tasniflash usullari. barqarorlik va yetakchi tadqiqotlar onlayn ilmiy-amaliy jurnali. Vol. 3 No. 12 (2023). 263-276
3. Elov B., Aloyev N., Yuldashev A. (2023). SVD va NMF metodlari orqali tematik modellashtirish. O‘zbekiston: til va madaniyat (Kompyuter lingvistikasi), 2023, 2(6). 55-66
4. Alghamdi, R., & Alfalqi, K. (2015). A Survey of Topic Modeling in Text Mining. International Journal of Advanced Computer Science and Applications, 6(1). https://doi.org/10.14569/ijacsa.2015.060121
5. Tao, R., Wei, Y., & Yang, T. (2021). Metaphor Analysis Method Based on Latent Semantic Analysis. Journal of Donghua University (English Edition), 38(1). https://doi.org/10.19884/j.1672-5220.202010087
6. Darmalaksana, W., Slamet, C., Zulfikar, W. B., Fadillah, I. F., Maylawati, D. S. adillah, & Ali, H. (2020). Latent semantic analysis and cosine similarity for hadith search engine. Telkomnika (Telecommunication Computing Electronics and Control), 18(1). https://doi.org/10.12928/TELKOMNIKA.V18I1.14874
7. Ke, Z. T., & Wang, M. (2022). Using SVD for Topic Modeling. Journal of the American Statistical Association.
https://doi.org/10.1080/01621459.2022.2123813
8. Churchill, R., & Singh, L. (2022). The Evolution of Topic Modeling. ACM Computing Surveys, 54(10). https://doi.org/10.1145/3507900
9. Kherwa, P., & Bansal, P. (2020). Topic Modeling: A Comprehensive Review. EAI Endorsed Transactions on Scalable Information Systems, 7(24). https://doi.org/10.4108/eai.13-7-2018.159623