O‘ZBEK TILI KORPUSI MATNLARI UCHUN TF-IDF STATISTIK KO‘RSATKICHNI HISOBLASH

Botir Elov Boltayevich
Texnika fanlari falsafa doktori, dotsent
Alisher Navoiy nomidagi Toshkent davlat o‘zbek tili va adabiyoti universiteti Kompyuter lingvistikasi va raqamli texnologiyalar kafedrasi mudiri
Zilola Xusainova Yuldashevna
Alisher Navoiy nomidagi Toshkent davlat o‘zbek tili va adabiyoti universiteti stajor-o‘qituvchisi
Nizomaddin Xudayberganov Uktambay o‘g‘li
Alisher Navoiy nomidagi Toshkent davlat o‘zbek tili va adabiyoti universiteti stajor-o‘qituvchisi
https://doi.org/10.5281/zenodo.7440059

Annotatsiya. *Matnli ma’lumotlarni qayta ishlashning eng keng tarqalgan usullaridan biri TF-IDF hisoblanadi. Google qidiruv tizimi ko‘p yillar davomida foydalanuvchi so‘roviga mos kontentni tartiblash uchun TF-IDF usulidan foydalanib kelmoqda. Amalga oshirilgan tadqiqotlar natijasiga ko‘ra Google tizimi kalit so‘zlarni hisoblashdan ko‘ra terminlar chastotasiga ko‘proq e’tibor qaratganligi aniqlangan. TF-IDF usuli orqali aniqlangan qiymat kalit so‘zning til korpusidagi dolzarbligini ifodalaydi. TF-IDF usuli orqali korpus hujjatlariga mos raqamli vektor hosil qilinadi. Ushbu raqamli vektori ma’lumot qidirish (IR) va mashinali o‘rganish (ML) sohalarida qo‘llaniladigan o‘lchov bo‘lib, qator ko‘rinishlarining (so‘zlar, iboralar, lemmalar va boshqalar) hujjat uchun ahamiyatini ifodalaydi. Ushbu maqolada o‘zbek tili korpusidagi hujjatlarni TF-IDF usulidan foydalanib, kalit so‘zga mos tarzda tartiblash jarayonini ko‘rib chiqamiz.*

Kalit so‘zlar: *TF-IDF, BoW, raqamli vektorlar, korpus, so‘zlar chastotasi, hujjatning teskari chastotasi, tokenizatsiya, lemmatizatsiya.*

СТАТИСТИЧЕСКИЙ ИНДЕКС, РАССЧИТАННЫЙ С ИСПОЛЬЗОВАНИЕМ TF-IDF ДЛЯ ТЕКСТОВ НА УЗБЕКСКОМ ЯЗЫКЕ

Аннотация*. Одним из наиболее распространенных методов обработки текстовых данных является TF-IDF. Поисковая система Google уже много лет использует метод TF-IDF для ранжирования контента, соответствующего запросам пользователей. По результатам проведенного исследования было определено, что система Google больше внимания уделяла частоте употребления терминов, чем расчету использования ключевых слов. Значение, определенное методом TF-IDF, представляет собой релевантность ключевого слова в языковом корпусе. С помощью метода TF-IDF генерируется цифровой вектор, соответствующий документам существующих в корпусе.. Этот числовой вектор является мерой, используемой в области поиска информации (IR) и машинного обучения (ML) для представления важности строковых представлений (слов, фраз, лемм и т. д.) в документе. В данной статье мы рассмотрим процесс сортировки документов в узбекскоязычном корпусе методом TF-IDF по ключевому слову.*

Ключевые слова: *TF-IDF, BoW, цифровые векторы, корпус, частота слов, обратная частота документа, токенизация, лемматизация.*

A STATISTICAL INDEX CALCULATED USING THE TF-IDF FOR TEXTS IN THE UZBEK LANGUAGE CORPUS

Abstract. *One of the most common methods of processing textual data is TF-IDF. Google's search engine has been using the TF-IDF method for ranking content relevant to user queries for many years. According to the results of the conducted research, it was determined that the Google system paid more attention to the frequency of terms than to the calculation of keywords. The value determined by the TF-IDF method represents the relevance of the keyword in the language corpus. Using the TF-IDF method, a digital vector corresponding to corpus documents is generated. This numeric vector is a measure used in the fields of information retrieval (IR) and machine learning (ML) to represent the importance of string representations (words, phrases, lemmas, etc.) to a document. In this article, we will consider the process of sorting documents in the Uzbek language corpus using the TF-IDF method according to the keyword.*

Keywords*: TF-IDF, BoW, digital vectors, corpus, word frequency, document inverse frequency, tokenization, lemmatization.*

Kirish

TF-IDF (Term Frequency-Inverse Document Frequency) – kalit so‘zning berilgan hujjatlar to‘plamiga mosligini aniqlash usuli bo‘lib, TF-IDF qiymat statistik ko‘rsatkich (baho) hisoblanadi [Stecanella, 2019]. J.Qin va Z.Zhoular tomonidan TF-IDF usulidan foydalanib, xitoy tilidagi so‘zlarni segmentatsiya algoritmi ishlab chiqilgan [Qin, Zhou, Tan, Xiang, He, 2021]. Ular tomonidan taklif qilingan algoritm yordamida ijtimoiy tarmoqdagi yangiliklar klasterlarga ajratilib, yangiliklardagi aktual so‘zlar aniqlangan va amalga oshirilgan eksperimentlar asosida hozirgi kundagi dolzarb mavzularni samarali topish imkoniyati taqdim etilgan. D. Cahyani va I. Patasik tomonidan berilgan matndagi inson tuyg‘ularni tasniflash uchun TF-IDF va Word2Vec raqamli vektorlar modellaridan foydalanilgan [Cahyani, Patasik, 2021; Pietro, 2020]. Shuningdek, tvit shaklidagi ma’lumotlarini tasniflash ikki bosqichda SVM (support vector machine) va MNB (Multinomial Naïve Bayes) usullaridan foydalanib amalga oshirilgan. Birinchi bosqischda his-tuyg‘ularni o‘z ichiga olgan yoki his-tuyg‘usiz ma’lumotlar aniqlangan. Ikkinchi bosqichda his-tuyg‘ularni o‘z ichiga olgan ma’lumotlar hissiyotlarning besh turiga, ya’ni *baxtli, g‘azablangan, qayg‘uli, qo‘rquv va hayratga* ajratilgan. Ushbu tadqiqotda TF-IDF bilan SVM, Word2Vec bilan SVM va TF-IDF bilan MNB metodlar qo‘llanilgan va natijalar o‘zaro qiyosiy taqqoslangan. R. Qaiser va R. Ali tomonidan korpusdagi hujjatlarga kalit so‘zlarning mosligini tekshirish muhokama qilingan [Qaiser, Ali, 2018]. Ularning tadqiqotlari TF-IDF algoritmini hujjatlar soni bo‘yicha qanday qo‘llash mumkinligiga qaratilgan. Birinchidan, TF-IDFni amalga oshirish uchun amal qilish kerak bo‘lgan bosqichlar ketma-ketligi ishlab chiqilgan. Ikkinchidan, TF-IDF algoritmi natijalari tahlil qilingan va TD-IDF algoritmining kuchli va zaif tomonlari keltirilgan. Ijtimoiy tarmoqlar va Internet global miqyosda foydalanuvchilarning o‘zaro yangiliklar, g‘oyalar va ma’lumotlarni bir zumda almashish imkonini berdi. B. Ahmed va G. Ali tomonidan olib borilgan tadqiqotlarda ijtimoiy tarmoqlar va Internetdagi mish-mishlar yoki soxta xabarlarni aniqlash algoritmlari ishlab chiqilgan va joriy etilgan [Ahmed, Ali, Hussain, Baseer, Ahmed, 2021]. Ular tomonidan taqdim etilgan yondashuvga ko‘ra 3 ta xususiyatli ekstraktorlardan foydalangan holda neyron tarmoqlarga asoslanib soxta yangiliklarni aniqlash modeli ishlab chiqilgan: TD-IDF, Glove va BERT. Baholash uchun har bir xususiyat ekstraktori uchun bir nechta ko‘rsatkichlar, ya’ni aniqlik, eslab qolish, AUC ROC va AUC PR qiymatlar hisoblab chiqilgan va transformatsiya usullari chuqur

o‘rganish modeliga joriy etilgan. TD-IDF usuli orqali olingan natijalar neyron tarmoqlarga asoslangan BERT usuliga yaqinroq qiymatlarni bergan.

Asosiy qism

Bugungi kunda o‘zbek tili korpusi matnlari uchun TF-IDF usulidan foydalanish jarayoni respublikamizda deyarli amalga oshirilmagan. Ushbu maqolada mualliflar guruhi tomonidan o‘zbek tili ta’limiy korpusi [http://uzschoolcorpara.uz] matnlari uchun TF-IDF usulidan foydalanib raqamli vektorlarni hosil qilish va ularni mashinali o‘rganishda qo‘llash usullari keltiriladi.

TF-IDF usuli ikkita statistik ko‘rsatkichni o‘zaro ko‘paytirish orqali aniqlanadi:

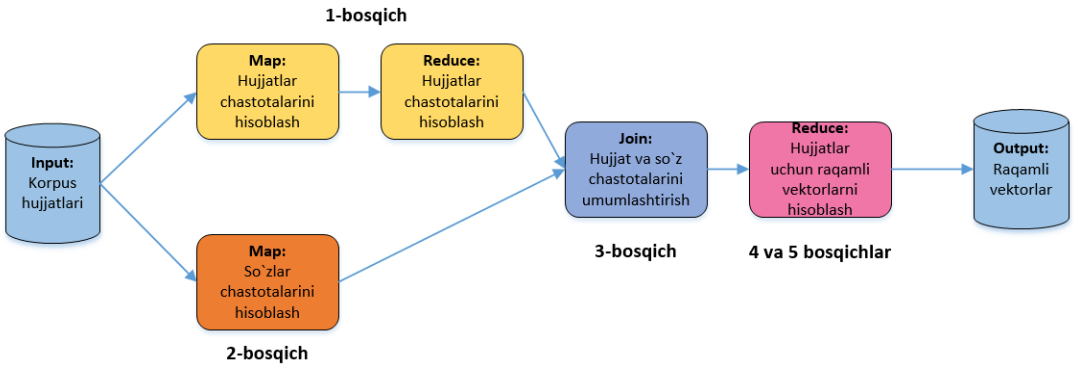
– **so‘zlar chastotasi (Term Frequency, TF):** *so‘zning hujjatda necha marta uchrashi;*

– **hujjatning teskari chastotasi (Inverse Document Frequency, IDF):** *hujjatlar to‘plamidagi so‘zning teskari chastotasi. Ushbu chastota orqali aniqlangan qiymatlarga ko‘ra unikal so‘zlar yuqori ballga, ko‘p qo‘llaniladigan so‘zlar past ballga ega bo‘ladi.*

Hujjatdagi so‘zning chastotasi. Ushbu chastotani hisoblashning bir necha usullari mavjud. Eng oddiy usulda hujjatda so‘z paydo bo‘lgan holatlarni aniqlash hisoblanadi. Shuningdek, hujjat uzunligi yoki hujjatdagi ko‘p uchraydigan so‘zning dastlabki chastotasi bo‘yicha chastotani sozlash usullari ham mavjud [Stecanella, 2019; Cahyani, Patasik, 2021; Pietro, 2020].

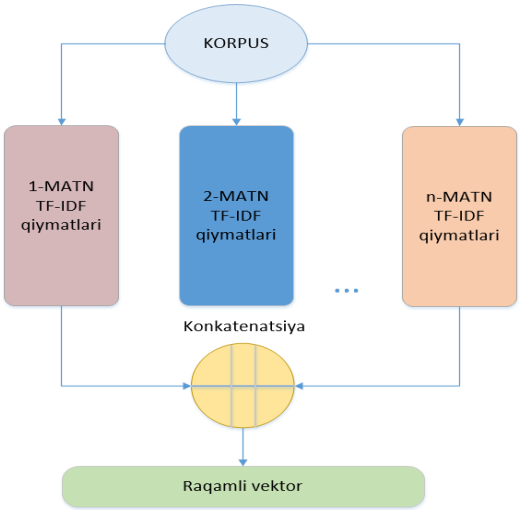
Hujjatning teskari chastotasi. Korpusdagi hujjatlar to‘plamida so‘z qanchalik keng tarqalgan yoki kamdan-kam uchrashini anglatadi [Stecanella, 2019; Qaiser, Ali, 2018; Ahmed, Ali, Hussain, Baseer, Ahmed, 2021].

TF-IDF usulidan *ma’lumot olish (information retrieval)* [Carneiro, Novais, Neves, 2014; Azad, Deepak, 2019], *matn tahlili (text analysis)* [Kharis, Laksono, Suhartono, Ridwan, Mintowati, Yuniseffendri, 2022], *kalit so‘zlarni ajratib olish (keyword extraction)* va *mashinali o‘rganish algoritmlari (machine learning algorithms)* [Razno, 2019] uchun matndan raqamli xususiyatlarni ajratib olish kabi NLP vazifalarida foydalaniladi. TF-IDF usuli orqali korpus matnlarini tahlil qilish bosqichlari quyidagi 1-rasmda keltirilgan:



1-rasm. TF-IDF qiymatni hisoblash bosqichlari

TF-IDF usuli (2-rasm) birinchi bo‘lib hujjatlarni qidirish va ma’lumot olish algortimlarida qo‘llanilgan bo‘lib, foydalanuvchi so‘roviga mos tarzda axborot tizimi ma’lumotlar bazasidan eng kerakli hujjatlarni aniqlagan. Masalan, **“bu xato”** so‘roviga mos amallar quyidagicha bajarilgan. Tizim har bir hujjatga kam uchraydigan **“xato”** so‘ziga mos chastotaga mutanosib ravishda yuqori ball qo‘yadi va **“bu”** kabi umumiy so‘zlarga kichikroq qiymatni mos qo‘yadi.



2-rasm. TF-IDF qiymatni hisoblash

d hujjatidagi **t** termini uchun $W_{t,d}$ vazn qiymati quyidagicha hisoblanadi:

$$W_{t,d} = TF_{t,d} * \log(N/DF_t)$$

bunda,

$TF_{t,d}$ – d hujjatida t ning uchrash soni;

DF_t – t terminimi o‘z ichiga olgan hujjatlar soni;

N – korpusdagi hujjatlarning umumiy soni;

Hujjat chastotani aniqlashning bir nechta o‘lchovlari yoki usullari mavjud:

– *hujjatda so‘z paydo bo‘lish soni (rc);*

– *hujjatning uzunligiga nisbatan chastota (chastota soni hujjatdagi so‘zlar soniga bo‘linadi);*

– *logarifmik o‘lchovli chastota (masalan, log10(1 + rc));*

– *mantiqiy chastota (masalan, hujjatda so‘z paydo bo‘lsa – 1, bo‘lmasa 0).*

Ushbu maqolada TF-IDF usulini korpusdagi matnlarga qo‘llash orqali so‘rovga mos tarzda hujjatlarni tartiblash va korpus matnlari uchun TF-IDF qiymatlarni hisoblash algoritmlarini ko‘rib chiqamiz va o‘zbek tilidagi matnlarga qo‘llaymiz.

I. Foydalanuvchi so‘rovi (kalit so‘zlar) asosida korpusdagi matnlarni TF-IDF usuli vositasida tartiblash. TF-IDF qiymat **[0..1]** oralig‘ida baholanadi. Raqamli vazn qiymati qanchalik katta bo‘lsa, termin korpusda shunchalik kam uchraydi. Og‘irligi qanchalik kichik bo‘lsa, termin korpusda shunchalik keng tarqalgan bo‘ladi.

Mavjud til korpusdagi **D1**, **D2** va **D3** hujjatlariga mos **Q** so‘rovini amalga oshirish lozim bo‘lsin.

Q: Qish ham keldi.

D1: Qor parchalari ham bir-biriga aslo o‘xshamaydi.

D2: Qor yog‘masa, qish faslining qizig‘i ham yo‘qday go‘yo.

D3: O‘lkamizga qish fasli ham kirib keldi.

TF qiymatni hisoblashning bir necha usullari mavjud bo‘lib, ko‘p hollarda hujjatda so‘zning uchrash soni aniqlanadi. So‘rovga mos hujjatdagi uchrashlar sonini hujjat uzuligiga nisbati yordamida TF qiymatni hisoblaymiz:

TF (word, document) = “hujjatdagi so‘zning chastotasi soni” / “hujjatdagi so‘zlar soni” yoki

$$\mathrm{TF}_{t,d} = \frac{n_{t,d}}{\sum_k n_{t,d}}$$

D1, D2 va D3 hujjatlariga nisbatan “qish”, “ham” va “keldi” so‘zlarining TF qiymatlarini hisoblaymiz. Ushbu amalni bajarishdan avval D1, D2, D3 hujjatlar va Q so‘rovida lemmatizatsiya amalini bajarish lozim [x].

TF (“qish”, D1) = 0/7 = 0
TF (“qish”, D2) = 1/8 = 0.125
TF (“qish”, D3) = 1/6 = 0.167
TF (“ham”, D1) = 1/7 = 0.167
TF (“ham”, D2) = 1/8 = 0.125
TF (“ham”, D3) = 1/6 = 0.167
TF (“kelmoq”, D1) = 0/7 = 0
TF (“kelmoq”, D2) = 0/8 = 0
TF (“kelmoq”, D3) = 1/6 = 0.167

	ham	**kelmoq**	**qish**
TF(D1)	0.166667	0.0	0.0
TF(D2)	0.125	0.0	0.125
TF(D3)	0.166667	0.166667	0.166667
IDF	0.0	0.477121	0.176091

IDF qiymatni hujjatlarning umumiy sonini, berilgan so‘zni o‘z ichiga olgan hujjatlar soniga bo‘lish va logarifmni aniqlash orqali hisoblash mumkin. Agar so‘zdan korpusdagi hujjatlarda ko‘p foydalanilgan bo‘lsa IDF qiymat **0** ga, aks holda **1** ga yaqin bo‘ladi.

IDF(word) = log(hujjatlar soni / so‘zni o‘z ichiga olgan hujjatlar soni) yoki

$$\mathrm{IDF(w)} = \log(\frac{N}{\mathrm{DF}_t})$$

Keyingi qadamda “qish”, “ham” va “keldi” so‘zlarining IDF qiymatlarini hisoblaymiz:

IDF(“qish”) = log(3/2) = log(1.5) = 0.176
IDF(“ham”) = log(3/3) = log(1) = 0
IDF(“kelmoq”) = log(3/1) = log(3) = 0.477

TF va IDF qiymatlarni ko‘paytirish orqali so‘zning hujjatga mos TF-IDF qiymati aniqlanadi. Ushbu qiymat qanchalik katta bo‘lsa, joriy hujjatda so‘z shunchalik muhim (dolzarb) hisoblanadi. TF-IDF (word, document) = TF (word, document) * IDF (word). Keyingi qadamda “the” va “cat” so‘zlarining TF-IDF qiymatlarini hisoblaymiz:

TF-IDF(“qish”, D1) = 0 * 0.176 = 0
TF-IDF(“qish, D2) = 0.125 * 0.176 = 0.022
TF-IDF(“qish”, D3) = 0.167 * 0.176 = 0.029
TF-IDF(“ham”, D1) = 0.167 * 0 = 0
TF-IDF(“ham, D2) = 0.125 * 0 = 0
TF-IDF(“ham”, D3) = 0.167 * 0 = 0
TF-IDF(“kelmoq”, D1) = 0 * 0.477 = 0
TF-IDF(“kelmoq, D2) = 0 * 0.477 = 0
TF-IDF(“kelmoq”, D3) = 0.167 * 0.477 = 0.079

Ushbu uchta qiymatlar to‘plamini birlashtirib, korpus hujjatlaridagi so‘z uchun TF-IDF qiymatni (w) olamiz:

$$w_{t,d} = TF_{t,d} * \log(\frac{N}{DF_t})$$

Keyingi qadamda, korpusdagi hujjatlarni berilgan Q so‘rovga mos TF-IDF qiymatlar bo‘yicha tartiblash amalga oshiriladi. Q so‘roviga nisbatan D1, D2 va D3 hujjatlar bo‘yicha o‘rtacha TF-IDF qiymatlardan foydalanish lozim.

Average TF-IDF of D1 = (0 + 0 + 0) / 3 = 0

Average TF-IDF of D2 = (0 + 0.022 + 0) / 3 = 0.0073

Average TF-IDF of D3 = (0.079 + 0.029 + 0) / 3 = 0.036

	ham	kelmoq	qish	AVG
TF-IDF(D1)	0.0	0.0	0.0	0.000000
TF-IDF(D2)	0.0	0.0	0.022011	0.007337
TF-IDF(D3)	0.0	0.07952	0.029349	0.036290

Yuqoridagi natijalarni tahlil qilish natijasida “ham” so‘zining barcha hujjatlarning TF-IDF qiymatiga ta’sir ko‘rsatmasligini aniqlash mumkin. Chunki, “ham” barcha hujjatlarda uchraydi va shu sababli u ushbu korpus uchun **nomuhim so‘z** hisoblanadi. Stiven E. Robertson va Karen Spärk Jons tomonidan ishlab chiqilgan ehtimollik qidiruv tizimiga asoslangan Okapi BM25 [https://en.wikipedia.org] funksiyasi yordamida hujjatlar reytingini aniqlash algoritmlari ham mavjud bo‘lib, ular yordamida tahlil samaradorligini yanada oshirish imkoniyati taqdim etiladi. Xulosa sifatida, D1, D2 va D3 hujjatlar to‘plami bo‘yicha so‘rovni bajarishda reyting natijalari quyidagicha bo‘ladi:

Q: Qish ham keldi.

D3: O‘lkamizga qish fasli ham kirib keldi.

D2: Qor yog‘masa, qish faslining qizig‘i ham yo‘qday go‘yo.

D1: Qor parchalari ham bir biriga aslo o‘xshamaydi.

Yuqorida keltirilgan TF-IDF qiymatlarni hisoblashlarni Python yili vositalari yordamida amalga oshiramiz va olingan natijalar tahlilini taqdim etamiz:

```
# so‘rov
wd="Qish ham keldi".lower()
# Tokenizatsiya + Lemmatizatsiya
LWd=tolist(Lemma(wd))
# Umumiy lug‘atni shakllantirish
wordSetWd = set(LWd)

# DataFrame ni shakllantirish
df3 = pd.DataFrame(columns = wordSetWd,
                   index = ['TF(D1)', 'TF(D2)', 'TF(D3)', 'IDF'])
# TF qiymatlarni hisoblash metodi
def TermFreq(word, document, id):
   N = len(document)
   occurance = len([token for token in document if token == word])
   return occurance/N
# IDF qiymatlarni hisoblash metodi
```

```
def InverseDocFreq(word):
    try:
        word_occurance = sum(df1[word].values.tolist())
    except:
        word_occurance = total_documents
    return np.log10(total_documents/word_occurance)
# TF qiymatlarni hisoblash
for word in wordSetWd:
        df3[word][0]=TermFreq(word, bowA, 0)
        df3[word][1]=TermFreq(word, bowB, 1)
        df3[word][2]=TermFreq(word, bowC, 2)
        df3[word][3]=InverseDocFreq(word)
# TF-IDF qiymatlarni hisoblash
df4 = pd.DataFrame(columns = wordSetWd,
                index = ['TF-IDF(D1)', 'TF-IDF(D2)', 'TF-IDF(D3)'])
for word in wordSetWd:
        df4[word][0]=df3[word][0]*df3[word][3]
        df4[word][1]=df3[word][1]*df3[word][3]
        df4[word][2]=df3[word][2]*df3[word][3]
# O‘rta arifmetik qiymatni aniqlash
df4['AVG‘]=df4.sum(axis=1)/total_documents
# Tartiblash va natijasni chiqarish
df5=df4.sort_values(by=['AVG‘], ascending=False)
print(df5)
```

O‘zbek tili korpusidagi matnlar va berilgan so‘rovga mos TF-IDF qiymatlarni hisoblashda tokenlarga ajratish yoki lemmatizatsiya asosida tahlilni amalga oshirish mumkin [http://uzschoolcorpara.uz]. Faqat tokenlar asosida tahlil qilinganda turlicha TF-IDF qiymatlari hosil qilinadi [http://uznatcorpara.uz]:

3-rasm. Tokenizatsiya va lemmatizatsiya jarayonining farqi

II. Korpus matnlari uchun TF-IDF qiymatlarni hisoblash

Yuqorida keltirilgan fikr-mulohazalar asosida berilgan D1, D2 va D3 ta hujjatlar uchun TF-IDF qiymatlarni hisoblash ketma-ketligini ko‘rib chiqamiz.

TF-IDF qiymatlarni Pythonda hisoblash. TF-IDF qiymatlarni Python vositalari yordamida hisoblash ketma-ketligini shakllantiramiz. So‘ngra, biz jarayonni avtomatlashtirish uchun **sklearn paketi**dan foydalanishni ko‘rib chiqamiz. **computeTF** funksiyasi orqali hujjat bo‘yicha korpusdagi har bir so‘z uchun TF qiymat hisoblanadi.

```
def computeTF(wordDict, bow):
    tfDict={}
    bowcount=len(bow)
    for word,count in wordDict.Items():
        tfDict[word]=count/float(bowcount)
    return tfDict
```

computeIDF funksiyasi korpusdagi har bir so‘zning IDF qiymatini hisoblaydi:

```
def computeIDF(docList):
    import math
    idfDict={}
    N=len(docList)
    idfDict=dict.fromkeys(docList[0].keys(),0)
    for doc in docList:
        for word, val in doc.Items():
            if val>0:
                idfDict[word]+=1
    for word, val in idfDict.Items():
        idfDict[word]=math.log10(N/float(val))
        return idfDict
```

Quyidagi **computeTF-IDF** funksiyasi TF va IDF qiymatlarni ko‘paytirish orqali har bir so‘z uchun TF-IDF qiymatni hisoblaydi:

```
def computeTFIDF(tfBow, idfs):
    tfidf={}
    for word,val in tfBow.items():
        tfidf[word]=val*idfs[word]
    return tfidf
```

TF-IDF qiymatlarni Pythonda hisoblashning yakuniy qadamlari quyida keltirilgan:

```
# Korpus hujjatlari
docA = "Qor parchalari ham bir biriga aslo o‘xshamaydi"
docB = "Qor yog‘masa, qish faslining qizig‘i ham yo‘qday go‘yo"
docC = "O‘lkamizga qish fasli ham kirib keldi"
# Tokenizatsiya
bowA = docA.split(" ")
bowB = docB.split(" ")
bowC = docC.split(" ")
# Umumiy lug‘atni shakllantirish
wordSet = set(bowA).union(set(bowB)).union(set(bowC))
```

```
wordDictA = dict.fromkeys(wordSet, 0)
wordDictB = dict.fromkeys(wordSet, 0)
wordDictC = dict.fromkeys(wordSet, 0)
# Chastotani aniqlash
for word in bowA:
    wordDictA[word]+=1
for word in bowB:
    wordDictB[word]+=1
for word in bowC:
    wordDictC[word]+=1
# BoW raqamli vektorini chop qilish
import pandas as pd
print(pd.DataFrame([wordDictA, wordDictB, wordDictC]))
```

1-jadval. Berilgan matnlarga mos raqamli vektorlar

	Qor	aslo	bir	parchalari	o‘xshamaydi	yo‘qday	Qish	biriga
1	1	1	1	1	1	0	0	1
2	1	0	0	0	0	1	1	0
3	0	0	0	0	0	0	1	0

	kirib	faslining	qizig‘i	fasli	keldi	yog‘masa	O‘lkamizga	ham	go‘yo
1	0	0	0	0	0	0	0	1	0
2	0	1	1	0	0	1	0	1	1
3	1	0	0	1	1	0	1	1	0

```
# TF qiymatlarni hisoblash
tfBowA = computeTF(wordDictA, bowA)
tfBowB = computeTF(wordDictB, bowB)
tfBowC = computeTF(wordDictC, bowC)
# BoW raqamli vektorini aniqlash
idfs = computeIDF([wordDictA, wordDictB, wordDictC])
# TF-IDF qiymatlarni hisoblash
tfidfBowA = computeTFIDF(tfBowA, idfs)
tfidfBowB = computeTFIDF(tfBowB, idfs)
tfidfBowC = computeTFIDF(tfBowC, idfs)
# TF-IDF qiymatlarni chop qilish
import pandas as pd
print(pd.DataFrame([tfidfBowA, tfidfBowB, tfidfBowC]))
```

2-jadval. Berilgan matnlarga mos TF-IDF qiymatlar

	Qor	Also	bir	parchalari	o‘xshamaydi	yo‘qday	qish	biriga
1	0.025	0.068	0.068	0.068	0.068	0	0	0.068
2	0.022	0	0	0	0	0.06	0.022	0
3	0	0	0	0	0	0	0.029	0

	kirib	faslining	qizig‘i	fasli	keldi	yog‘masa	O‘lkamizga	ham	go‘yo
1	0	0	0	0	0	0	0	0	0
2	0	0.06	0.06	0	0	0.06	0	0	0.06
3	0.079	0	0	0.079	0.079	0	0.079	0	0

1 va 2-jadvallardagi olingan natijalarda korpusdagi matnlarga mos TF-IDF qiymatlarni hisoblashda tokenlarga ajratishdan foydalanildi. Endi dasturni biroz optimallashtirib, lemmalar asosida TF-IDF qiymatlarni hisoblashni amalga oshiramiz va natijalarni keltiramiz:

3-jadval. Berilgan matnlarga mos raqamli vektorlar (+lemmatizatsiya)

	bir	fasl	aslo	kir	ham	qiziq	o‘lka	yog‘moq
1	2	0	1	0	1	0	0	0
2	0	1	0	0	1	1	0	1
3	0	1	0	1	1	0	1	0

	parchalamoq	kelmoq	o‘xshamoq	go‘yo	qish	yo‘q	qor
1	1	0	1	0	0	0	1
2	0	0	0	1	1	1	1
3	0	1	0	0	1	0	0

4-jadval. Berilgan matnlarga mos TF-IDF qiymatlar (+lemmatizatsiya)

	bir	fasl	aslo	kir	ham	qiziq	o‘lka	yog‘moq
1	0.159	0	0.079	0	0	0	0	0
2	0	0.022	0	0	0	0.06	0	0.06
3	0	0.029	0	0.079	0	0	0.079	0

	parchalamoq	kelmoq	o‘xshamoq	go‘yo	qish	yo‘q	qor
1	0.079	0	0.079	0	0	0	0.029
2	0	0	0	0.06	0.022	0.06	0.022
3	0	0.079	0	0	0.029	0	0

Demak, korpusidagi matnlarga mos TF-IDF qiymatlarni hisoblashda lemmatizatsiya jarayonini amalga oshirish orqali tahlil samaradorligini oshishini qayt etish mumkin.

Mashinali o‘rganishda TF-IDFdan foydalanish

TF-IDF usulidan ko‘p hollarda berilgan matnni *raqamli vektoriga aylantirish* jarayonida foydalaniladi. TF-IDF usuli hujjatdagi har bir so‘zni ushbu hujjat uchun qanchalik dolzarbligini ko‘rsatadigan **qiymat** bilan bog‘lash imkoniyatini taqdim etadi. Bunday qiymatlardan keyinchalik mashinali o‘rganish modellarining xususiyatlari sifatida foydalanish mumkin.

Tabiiy til masalalari bilan ishlashda mashinali o‘rganish bitta asosiy to‘siqqa duch keladi - uning algoritmlari odatda raqamlar ustida amallarni bajaradi. Tabiiy tilda esa matnlar qayta ishlanadi. Shunday qilib, tabiiy tildagi matnni raqamlarga aylantirishga to‘g‘ri keladi. Ushbu jarayon NLPda **matn vektorizatsiyasi** deb nomlanadi. Ushbu bosqich matnli ma’lumotlarni tahlil qilish uchun mashinali o‘rganish jarayonidagi asosiy qadamdir va turli vektorlashtirish algoritmlari yakuniy natijalarga keskin ta’sir qiladi.

Matndagi so‘zlar raqamli shaklga o‘tkazilganidan so‘ng, mashinali o‘rganish algoritmlari tushunadigan tarzda, TF-IDF usuli vositasida hisoblangan statistik qiymatlar natijalarini sezilarli darajada yaxshilaydigan **Naive Bayes** va **Support Vector Machines** kabi algoritmlarga berilishi mumkin.

Mashinali o‘rganish bilan matnni tahlil qilishda TF-IDF algoritmlari ma’lumotlarni toifalarga ajratishga, shuningdek, kalit so‘zlarni aniqlashga yordam beradi. Axborot tizimlardagi arizalarni online qabul qilish va tahlil qilish tizimlaridagi (tagging support tickets) ko‘p vaqtni talab qiluvchi NLP vazifalarida TF-IDF algoritmidan foydalanish orqali masalani bir necha soniya ichida hal qilish mumkin. Ushbu maqolada Google kabi qidiruv tizimlarining foydalanuvchi so‘roviga mos hujjatlarni aniqlash va tartiblash metodi haqida ma’lumot berildi.

Xulosa

Ushbu maqolada amalga oshirilgan hisoblashlar va tahlillar kontent muallifi yoki SEO mutaxassisiga juda foydali. Google ushbu algoritmdan qanday foydalanishini tushunib, kontent mualliflari TF-IDFni foydalanuvchilar va qidiruv tizimlari uchun optimallashtirishlari mumkin. Shunigdek NLP mutaxassislari katta hajmdagi til korpuslarida amalga oshiriladigan so‘rovlarga mos hujjatlarni qaytarish vositasi sifatida foydalanishlari mumkin. TF-IDF usuli orqali kontentni optimallashtirishni amalga oshirish imkoniyati taqdim etiladi. TF-IDF usuli orqali quyidagi natijalarga ega bo‘lish mumkin:

– **So‘zlarni yig‘ish**. O‘z kontent (matn)ingizga berilgan so‘rovlarga mos TF-IDF qiymatlarni aniqlash va ularning vazniga ega bo‘lsih;

– TF-IDF usuli orqali aniqlagan og‘irligi yuqori bo‘lgan barcha terminlarni Google qidiruv tizimi **natijalari bilan solishtirish**.

– Kalit so‘z bo‘yicha yuqori vaznga ega kontentga qidiruv tizimlari yuqori vazn beradi.

REFERENCES

1. Stecanella, B. (2019). What is TF IDF? *MonkeyLearn*
2. Qin, J., Zhou, Z., Tan, Y., Xiang, X., & He, Z. (2021). A big data text coverless information hiding based on topic distribution and tf-idf. *International Journal of Digital Crime and Forensics, 13*(4). https://doi.org/10.4018/IJDCF.20210701.oa4
3. Cahyani, D. E., & Patasik, I. (2021). Performance comparison of tf-idf and word2vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics, 10*(5). https://doi.org/10.11591/eei.v10i5.3157
4. Pietro, M. di. (2020). Text Classification with NLP: Tf-Idf vs Word2Vec vs BERT. *Medium*.
5. Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications, 181*(1). https://doi.org/10.5120/ijca2018917395
6. Ahmed, B., Ali, G., Hussain, A., Baseer, A., & Ahmed, J. (2021). Analysis of Text Feature Extractors using Deep Learning on Fake News. *Engineering, Technology & Applied Science Research, 11*(2). https://doi.org/10.48084/etasr.4069
7. O‘zbek tili ta’limiy korpusi - http://uzschoolcorpara.uz/
8. Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021). Semantic Sensitive TF-IDF to Determine Word Relevance in Documents. *Lecture Notes in Electrical Engineering, 736 LNEE*. https://doi.org/10.1007/978-981-33-6987-0_27

9. Carneiro, D., Novais, P., & Neves, J. (2014). Information Retrieval. In *Law, Governance and Technology Series* (Vol. 18). https://doi.org/10.1007/978-3-319-06239-6_7
10. Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. *Information Processing and Management*, *56*(5). https://doi.org/10.1016/j.ipm.2019.05.009
11. Kharis, M., Laksono, K., Suhartono, Ridwan, A., Mintowati, & Yuniseffendri. (2022). Tokenization and Lemmatization on German Learning Textbook Level A1 of CEFR Standard. *Journal of Higher Education Theory and Practice*, *22*(1). https://doi.org/10.33423/jhetp.v22i1.4971
12. Razno, M. (2019). Machine learning text classification model with NLP approach. *Computational Linguistics and IntelligeRazno, M. (2019). Machine Learning Text Classification Model with NLP Approach. Computational Linguistics and Intelligent Systems, 2(18-Apr-2019), 71–73.*
13. *Http://Ena.Lp.Edu.Ua:8080/Handle/Ntb/45487nt Systems*, *2* (18-Apr-2019).
14. O‘zbek tili morfologik analizatori - http://uznatcorpara.uz/