ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

ՆԵՐԱԾՈՒԹՅՈՒՆ

Տեղեկատվության արագ աճը և դրա մեծ ծավալները ստեղծել են նոր խնդիրներ տվյալների կառավարման ու վերլուծության մեջ։ Անհրաժեշտություն է առաջացել արդյունավետ ու ավտոմատացված մեթոդների կիրառման, որոնց միջոցով կարելի է հեշտությամբ գտնել օգտակար և համապատասխան ինֆորմացիա մեծ տվյալների բազաներում։ Առանցքային բառերի հայտնաբերման համակարգը կարևոր գործիք է այս խնդրի լուծման համար։ Այն օգնում է ի հայտ բերել տեքստերից կարևորագույն հասկացությունները, որոնք ավելի ուշ կարող են օգտագործվել տեղեկատվության դասակարգման, որոնման և կառավարման համար։ Այս ոլորտում մշակվող տեխնոլոգիաներն ունեն կիրառություն տարբեր ոլորտներում, այդ թվում՝ մարքեթինգ, գիտական հետազոտություններ, իրավաբանական վերլուծություններ, և այլն։

ԽՆԴՐԻ ԴՐՎԱԾՔ

Այս ուսումնասիրության նպատակն է մշակել արդյունավետ համակարգ առանցքային բառերի հայտնաբերման համար, որը կարող է ավտոմատ կերպով վերլուծել տեքստեր և գտնել դրանց մեջ առկա առանցքային բառերը։ Համակարգը պետք է կարողանա մշակել տարբեր տեքստային աղբյուրներ՝ հաշվի առնելով լեզվաբառական առանձնահատկությունները և ընթանալով տվյալների նախադրյալների միջոցով։ Այս խնդրի լուծումը պահանջում է հարուստ մեթոդաբանություն՝ լեզվաբառագիտության, մոդելավորման և մեքենայական ուսուցման տեխնիկաների օգտագործմամբ։

ԳԼՈՒԽ 1. ՏԵՍԱԿԱՆ ԱՌԸՆՉՈՒԹՅՈՒՆՆԵՐ

1.1 TF-IDF ԱԼԳՈՐԻԹՄԻ ՆԿԱՐԱԳՐՈՒԹՅՈՒՆ

TF-IDF-ը (Term Frequency-Inverse Document Frequency) տեքստային տվյալների վեկտորացման ալգորիթմ է, որն օգտագործվում է տեքստի կարևորագույն բառերը հայտնաբերելու համար: TF-IDF-ի հիմնական սկզբունքն այն է, որ բառի կարևորությունը տեքստում կախված է տվյալ բառի հաճախությունից և նրա գոյությունից մյուս տեքստերում:

TF-IDF-ի հաշվարկը կատարվում է հետևյալ քայլերով՝

1. **Term Frequency (TF)**: Յուրաքանչյուր բառի հաճախությունը տեքստում՝ հարաբերակցված տվյալ տեքստի բառերի ընդհանուր քանակին:

$$\mathrm{TF}(t,d) = \frac{\text{Number of occurrences of term } t \text{ in document } d}{\text{Total number of terms in document } d}$$

2. **Inverse Document Frequency (IDF)**: Բառի հայտնվելու հազվադեպության չափը՝ հաշվի առնելով նրա կրկնությունը ամբողջ կորպուսում:

$$\mathrm{IDF}(t,D) = \log\left(\frac{N}{1 + \text{Number of documents containing term } t}\right)$$

   - N - փաստաթղթերի ընդհանուր քանակը հավաքածուում:
   - Լոգարիթմը կարող է լինել բնական (ln) կամ 10-հիմքով (log10):

3. **Արդյունք (TF - IDF)**: Այսպիսով ստանում ենք վերջնական բանաձևը:

$$\mathrm{TF\text{-}IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$$

1.2 RANDOM FOREST-Ի ՄԱՍԻՆ

Random Forest-ը դասակարգման և ռեգրեսիայի խնդիրների լուծման համար օգտագործվող ալգորիթմ է: Այն կառուցում է բազմաթիվ որոշումների ծառեր՝ ուսուցման ժամանակ և օգտագործում է դրանց միջին արդյունքը: Այս մոտեցումը նվազեցնում է գերհարմարվողականությունը և բարելավում մոդելի ընդհանուր արդյունավետությունը:

Առավելությունները՝

- Հատուկ տվյալների համար հարմարեցման բարձր ճկունություն:
- Արդյունավետություն մեծածավալ տվյալների վրա:

ԳԼՈՒԽ 2. ԾՐԱԳՐԻ ԻՐԱԿԱՆԱՑՈՒՄ

2.1 ԱԼԳՈՐԻԹՄԻ ԻՐԱԿԱՆԱՑՄԱՆ ՔԱՅԼԵՐ

Ալգորիթմի իրականացումը բաղկացած է հետևյալ քայլերից.

1. **Տվյալների նախամշակում**:
    - Տեքստերի մաքրում, ձևափոխում և վեկտորացում re և TfidfVectorizer գործիքների միջոցով:
    - Պիտակների կոդավորում label_to_id և id_to_label բառարանների օգնությամբ:
2. **Մոդելի ուսուցում**:
    - RandomForestClassifier ալգորիթմի կիրառություն՝ ուսուցման և թեստավորման տվյալների վրա:
    - Հավասարակշռված դասային քաշերի սահմանում՝ դասակարգման արդարության համար:
3. **Արդյունքների գնահատում**:
    - Մետրիկաների, ներառյալ ճշգրտությունը, հիշողությունը և F1 միավորը, հաշվարկ classification_report գործիքի միջոցով:
4. **Կիրառման փուլ**:
    - Անհայտ տեքստերում սուբյեկտների հայտնաբերում և հարաբերությունների կանխատեսում:
    - predict_relationship ֆունկցիայի միջոցով արդյունքների մեկնաբանում:

2.2 ՏՎՅԱԼՆԵՐԻ ՆԱԽԱՄՇԱԿՈՒՄ

Տվյալների նախամշակումը համակարգի արդյունավետության համար կարևոր քայլ է, որը ներառում է հետևյալ գործընթացները՝

1. **Տեքստի մաքրում և ձևափոխում**:
    - Կիրառվել են կանոնավոր արտահայտություններ (re գրադարանի միջոցով), չպետքական նշանների հեռացման, ինչպես նաև տեքստի ձևավորման համար (օրինակ՝ հատուկ բառերը փոխարինելով [E1], [E2] նշաններով):
2. **Տեքստի չափանիշների վեկտորացում**:
    - Օգտագործվել է TfidfVectorizer գործիքը՝ տեքստերը թվային հատկանիշների վերածելու համար: Վեկտորիզացիան ապահովում է մուտքային տվյալների օպտիմալ ներկայացում՝ համակարգի համար:
3. **Պիտակների ձևավորում**:
    - Պիտակների եզակի արժեքները դասակարգվել են՝ կոդավորված թվային արժեքներով label_to_id և id_to_label բառարանների միջոցով:

2.3 ՄՈԴԵԼԻ ՈՒՍՈՒՑՈՒՄ

Սովորեցման փուլում կիրառվել է RandomForestClassifier ալգորիթմը, որը հանրաճանաչ է դասակարգման խնդիրներում՝ շնորհիվ իր ճկունության և բարձր արդյունավետության:

- **Տվյալների բաժանում**:
    - train_test_split գործառույթի միջոցով տվյալները բաժանվել են ուսուցման և թեստավորման խմբերի (80% և 20% հարաբերակցությամբ):
- **Հավասարակշռված դասային քաշեր**:
    - Քանի որ պիտակները կարող են ունենալ անհավասար հաճախականություններ, կիրառվել է class_weight='balanced' պարամետրը՝ հավասարակշռված մոդելավորման ապահովման համար:

2.4 ԴԱՍԱԿԱՐԳՄԱՆ ԳՆԱՀԱՏՈՒՄ

Մոդելի որակը գնահատվել է classification_report գործիքի միջոցով, որը տրամադրում է հետևյալ մետրիկաները՝

- Ճշգրտություն (Precision),
- Հիշողություն (Recall),
- F1 միավոր։

F1 միավոր

F1 միավորը մեթրիկա է, որը համատեղում է ճշգրտությունն (Precision) և հիշողությունը (Recall)՝ դասակարգիչի ընդհանուր արդյունավետությունը գնահատելու համար։ Այն հատկապես կարևոր է, երբ տվյալների հավասարակշռությունը խախտված է, և մեկ մետրիկայի վրա հիմնվելը կարող է լինել միակողմանի։

F1-ի հաշվարկը իրականացվում է հետևյալ բանաձևով՝

Այս ցուցանիշը տալիս է դասակարգման ճշգրտության և հիշողության ներդաշնակ միջինը, որտեղ 1-ը ներկայացնում է լավագույն արդյունքը, իսկ 0-ը՝ ամենացածրը։

2.5 ԿԻՐԱՌՄԱՆ ՓՈՒԼ

Մոդելն օգտագործվել է անհայտ տեքստերի վերլուծության համար, ինչի նպատակն էր՝

1. **Սուբյեկտների հայտնաբերում**:
    - Անալիզը ներառում է e1 և e2 սուբյեկտների տեղորոշում և առանձնացում տեքստից։
2. **Դասակարգման արդյունքների մեկնաբանում**:
    - Կիրառվել է predict_relationship ֆունկցիան՝ ամենաբարձր հավանականությամբ դասերի վերլուծության համար, որը հնարավորություն է տալիս հասկանալ սուբյեկտների միջև հարաբերության բնույթը։

2.6 ԳՐԱԴԱՐԱՆՆԵՐԻ ԸՆՏՐՈՒԹՅԱՆ ՊԱՏՃԱՌԱԲԱՆՈՒԹՅՈՒՆ

Օգտագործվել են re, numpy, sklearn գրադարանները։

- re գրադարանը թույլ է տալիս իրականացնել արդյունավետ նախամշակում տեքստերի վրա՝ հաշվի առնելով տվյալների առանձնահատկությունները։
- numpy-ն կիրառվել է բազմաբնույթ մաթեմատիկական գործողությունների պարզեցման համար։
- sklearn-ի գործիքները ապահովել են վեկտորիզացիա, մոդելավորում և գնահատում՝ բարձրացնելով համակարգի արտադրողականությունը։

ԱՐԴՅՈՒՆՔՆԵՐ

Ծրագրային կոդի աշխատացման արդյունում ունենում ենք նկ.1-ի պատկերը:

```
                         precision    recall  f1-score   support

      Cause-Effect(e1,e2)       0.91      0.64      0.75        61
      Cause-Effect(e2,e1)       0.72      0.51      0.60       138
   Component-Whole(e1,e2)       0.37      0.21      0.27       100
   Component-Whole(e2,e1)       0.56      0.09      0.16        97
 Content-Container(e1,e2)       0.70      0.62      0.66        88
 Content-Container(e2,e1)       0.71      0.52      0.60        42
Entity-Destination(e1,e2)       0.67      0.87      0.75       158
Entity-Destination(e2,e1)       0.00      0.00      0.00         0
     Entity-Origin(e1,e2)       0.34      0.72      0.46       107
     Entity-Origin(e2,e1)       0.83      0.17      0.29        29
 Instrument-Agency(e1,e2)       0.33      0.07      0.12        14
 Instrument-Agency(e2,e1)       0.67      0.26      0.37        85
 Member-Collection(e1,e2)       1.00      0.12      0.21        17
 Member-Collection(e2,e1)       0.34      0.45      0.39       119
     Message-Topic(e1,e2)       0.55      0.06      0.11       103
     Message-Topic(e2,e1)       0.50      0.10      0.17        29
                    Other       0.22      0.51      0.31       260
  Product-Producer(e1,e2)       0.57      0.18      0.28        66
  Product-Producer(e2,e1)       1.00      0.02      0.04        87

                 accuracy                           0.42      1600
                macro avg       0.58      0.32      0.34      1600
             weighted avg       0.55      0.42      0.39      1600

Example sentence:
Cats don't taste sweetness

Extracted Entities:
e1: Cats
e2: taste

Predicted Relationships:
Other: 0.51
Entity-Destination(e1,e2): 0.23
Instrument-Agency(e2,e1): 0.09
```

Ն.կ 1. Ծրագրային կոդի աշխատանքի արդյունք

ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Առանցքային բառերի հայտնաբերման առաջարկվող համակարգը ցուցադրել է հաջող արդյունքներ՝ ապահովելով տեքստերի ավտոմատ վերլուծության բարձր արդյունավետություն, ճշգրտություն և հուսալիություն: Համակարգի ճկունությունը հնարավորություն է տալիս այն հեշտությամբ ընդլայնել տարբեր ոլորտներում՝ ինչպես գիտական ուսումնասիրություններում, այնպես էլ բիզնեսի ոլորտում: Հավելյալ հարմարեցման և ֆունկցիոնալության բարելավման շնորհիվ այն կարող է մշակել մեծածավալ տվյալներ և ապահովել ավելի լայն կիրառում՝ դարձնելով այն բազմակողմանի գործիք տեքստային տվյալների մշակման համար:

ՕԳՏԱԳՈՐԾՎԱԾ ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ

1. https://colab.google/
2. https://paperswithcode.com/
3. Դասախոսությունների նյութեր
4. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32
5. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830