

희
기
한

문
제
풀
이

김슬기, 양한솔, 이찬희



목차

1. 목표

2. 과정

2.1 자연어 기반 모델

2.2 유형분석

2.3 규칙 기반 모델

3. 결과

4. 개선 방안

1 프로젝트 목표

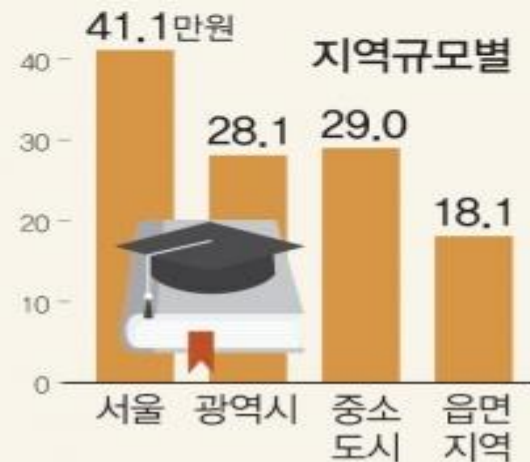
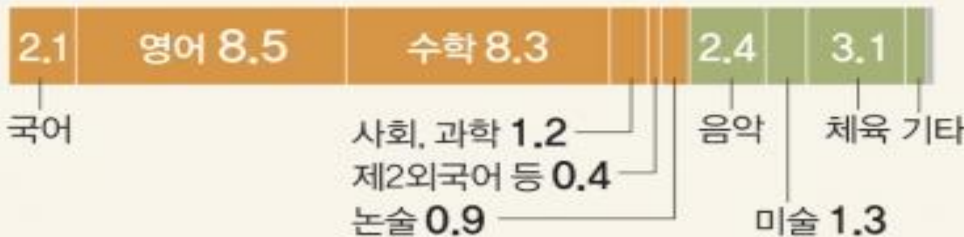
초·중·고 사교육비 현황 ※2018년 1인당 월 평균 기준



과목별 단위: 만원

일반교과 **21.3**만원

예체능 및 기타 **7.6**만원



1 프로젝트 목표

1. 문제

준기는 동화책을 어제는 58쪽, 오늘은 54쪽 읽었습니다. 준기는 어제와 오늘 동화책을 모두 몇 쪽 읽었을까요?

2. 수식화

$$58 + 54 =$$

3. 정답

[112]

2.1 자연어 기반 모델

형태소 분석

● 형태소 분리

<kkma>

'주', '니', '길', '는',
'동화책', '을', '어제',
'는', '58', '쪽', ' ',
'오늘', '은', '54', ' ',
'쪽', '읽', '었', '습니
다'

<Okt>

'준', '기는', '동화책',
'을', '어제', '는', ' ',
'58', '쪽', ' ', '오늘',
'은', '54', '쪽', '읽',
었습니다'

워드임베딩

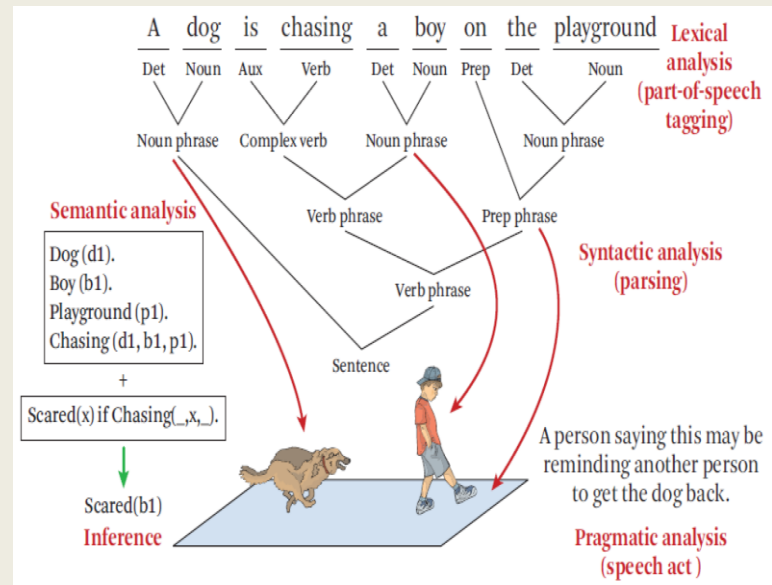
● 분리된 형태소 벡터화

<Word2Vec>

{'아야': array([5.
3870882e-04, -8.
1316539e-04, 1.
6465907e-03, 1.
2780178e-03,
-1.6143252
e-03, 5.9329579
e-04, -1.2845687
e-03, -9.7159127
e-04,

구문분석

수식화



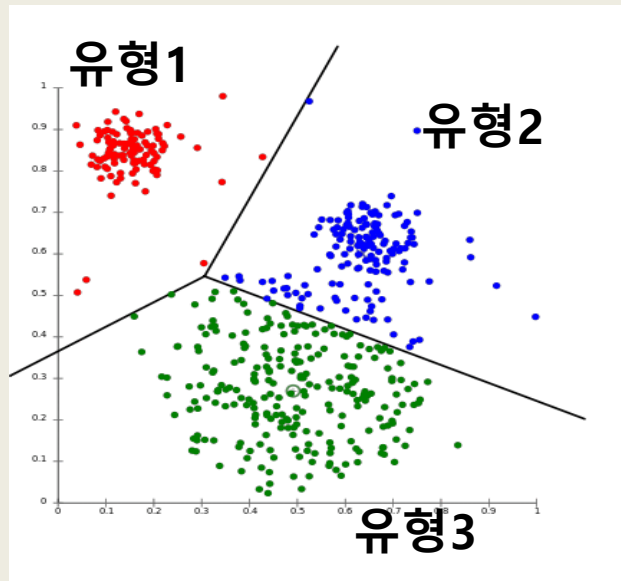
2.2 유형 분석 : k-means

형태소 분석

임베딩벡터

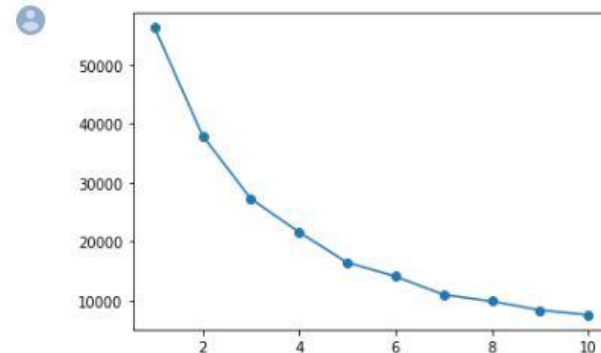
k-means

<하고자 한 목표>



<elbow curve>

```
[ ] elbow(data_points1)
```



- elbow curve가 완만
- 뚜렷한 k 값을 정할 수 없음
- 임의의 수 5로 선정

2.2 유형 분석 : k-means

< K-means 분석의 결과 >

245를 넣으면 127이 나오는 상자가 있습니다. 이 상자에 396을 넣으면 얼마가 나오니까?	0
483를 넣으면 386이 나오는 상자가 있습니다. 이 상자에 572를 넣으면 얼마가 나오니까?	1
194를 넣으면 36이 나오는 상자가 있습니다. 이 상자에 342를 넣으면 얼마가 나오니까?	0
75를 넣으면 38이 나오는 상자가 있습니다. 이 상자에 56을 넣으면 얼마가 나오니까?	1
324를 넣으면 112가 나오는 상자가 있습니다. 이 상자에 632를 넣으면 얼마가 나오니까?	1
560을 넣으면 367이 나오는 상자가 있습니다. 이 상자에 407을 넣으면 얼마가 나오니까?	0
어떤 상자에 24를 넣었더니 17이 나왔습니다. 51을 넣으면 얼마가 나오겠습니까?	0
어떤 상자에 78을 넣었더니 49가 나왔습니다. 95를 넣으면 얼마가 나오겠습니까?	4
어떤 상자에 56을 넣었더니 38이 나왔습니다. 32를 넣으면 얼마가 나오겠습니까?	0

● 문제점

모두 같은 유형임에도 다른 유형으로 분류된 것을 볼 수 있음

2.3 규칙 기반 모델

- (1) 수작업으로 유형 분류
- (2) 정규식을 활용해서 문제 추출, 답 계산
- (3) 유형으로 분류되지 않는 문제는 예외처리

< 유형 1 >

```
1 #매일 똑같은 수
2 for i in range(0, len(df)):
3     if (re.compile('[\d]{1,2}[\#\w]{1,2}가 하루에 [\#\w]{1,2}["을,를"] [\d]{1,2}[\#\w]{1,2}').findall(df["질문"][i])!=[]):
4         a=re.compile('[\d]{1,2}[\#\w]{1,2}가 하루에 [\#\w]{1,2}["을,를"] [\d]{1,2}[\#\w]{1,2}').findall(df["질문"][i])
5         aa=re.compile('[\d]{1,2}').findall(a[0])
6         b=df["질문"][i].split("다면")
7         bb=re.compile('[\d]{1,2}').findall(b[1])
8         df["model_answer"][i]=int(bb[1])//((int(aa[1])//int(aa[0]))*int(bb[0]))
9
10    print(i, df["질문"][i], df["model_answer"][i])
```

6 고양이 8마리가 하루에 쥐를 24마리씩 잡는다고 합니다. 모든 고양이가 매일 똑같은 수의 쥐를 잡는다면, 고양이 3마리가 쥐를 81마리를 잡는데 며칠이 걸리는지 구하십시오. 9

31 토끼 6마리가 하루에 당근을 18개 먹습니다. 모든 토끼가 매일 똑같은 개수의 당근을 먹는다면 토끼 8마리가 당근 72개를 먹는 데에는 며칠이 걸리겠습니까? 3

90 말 4마리가 하루에 당근을 8개 먹습니다. 모든 말이 매일 똑같은 개수의 당근을 먹는다면 말 6마리가 당근 48개를 먹는 데에는 며칠이 걸리겠습니까? 4

2.3 규칙 기반 모델

< 유형 2 >

```
1 #동전 총 금액
2
3 for i in range(0, len(df)):
4     if (re.compile("동전이 모두 [\\d]{1,4}개").findall(df["질문"][i]) != []) and (re.compile("[\\d]{1,4}원짜리").findall(df["질문"][i])):
5         a1=re.compile("동전이 모두 [\\d]{1,4}개").findall(df["질문"][i])
6         a11=re.compile("[\\d]{1,4}").findall(a1[0])
7         a2=re.compile("[\\d]{1,4}원짜리").findall(df["질문"][i])
8         a22=[]
9         for jj in range(0, len(a2)):
10             a22.append(int((re.compile("[\\d]{1,4}").findall(a2[jj]))[0]))
11         a3=re.compile("[\\d]{1,4}종류").findall(df["질문"][i])
12         a33=re.compile("[\\d]{1,4}").findall(a3[0])
13         k=int(a11[0])//int(a33[0])
14
15         df["model_answer"][i]=0
16         for j in range(0, len(a22)):
17             df["model_answer"][i]=df["model_answer"][i]+(int(a22[j])*k)
18
19     print(i, df["질문"][i], df["model_answer"][i])
```

85 동전이 모두 12개 있습니다. 동전은 10원짜리, 50원짜리, 100원짜리, 500원짜리로 4종류이고, 종류별 동전의 개수는 모두 같다고 합니다. 동전의 금액을 모두 더하면 얼마입니까? 1980

86 동전이 모두 27개 있습니다. 동전은 10원짜리, 50원짜리, 100원짜리로 3종류이고, 종류별 동전의 개수는 모두 같다고 합니다. 동전의 금액을 모두 더하면 얼마입니까? 1440

95 동전이 모두 18개 있습니다. 동전은 10원짜리, 50원짜리, 100원짜리로 3종류이고, 종류별 동전의 개수는 모두 같다고 합니다. 동전의 금액을 모두 더하면 얼마입니까? 960

2.3 규칙 기반 모델

< 유형 3 >

```

1 for i in range(0, len(df)):
2     if (re.compile("다리를 세었더니 모두 [wd]{1,4}개").findall(df["질문"][i])) or (re.compile("다리를 세어 보니 모두 [wd]{1,4}개").
3         a1=(re.compile("다리를 세었더니 모두 [wd]{1,4}개").findall(df["질문"][i])) or (re.compile("다리를 세어 보니 모두 [wd]{1,4}개").
4         a=int(re.compile("[wd]{1,4}").findall(a1[0])[0]) #총 다리수
5         r1=['문어가', '강아지가', '오리가', '염소가', '닭이', '토끼가', '문어는', '강아지는', '오리는', '염소는', '닭은', '토끼는', '문어', '강아
6         r2=[8,4,2,4,2,4,8,4,2,4,8,4,2,4,8,4,2,4,8,4,2,4,4]
7         b=[]
8         if re.compile("[ww]{1,4} [wd]{1,3}마리").findall(df["질문"][i])!=[]:
9             b=re.compile("[ww]{1,4} [wd]{1,3}마리").findall(df["질문"][i])
10        if re.compile("[ww]{1,4} [ww]{1,4} 모두 [wd]{1,3}마리라면").findall(df["질문"][i])!=[]:
11            b=re.compile("[ww]{1,4} 모두 [wd]{1,3}마리라면").findall(df["질문"][i])
12        c=[] #기존동물 다리수
13        for j in range(0, len(b)):
14            if re.compile("[ww]{1,4}").findall(b[j])[0] in r1:
15                c.append(r2[r1.index(re.compile("[ww]{1,4}").findall(b[j])[0])])
16            b[j]=int(re.compile("[wd]{1,3}").findall(b[j])[0]) # 기존 동물수
17
18
19        if re.compile("[ww]{1,4} 몇 마리").findall(df["질문"][i])!=[]:
20            d=re.compile("[ww]{1,4} 몇 마리").findall(df["질문"][i])
21            d=[r2[r1.index(re.compile("[ww]{1,4}").findall(d[0])[0])]] #구하고 싶은 동물 다리수
22
23        if b==[]: df["model_answer"][i]=a//d[0]
24        else:
25            for k in range(0, len(b)):
26                a=a-b[k]*c[k]
27            df["model_answer"][i]=a//d[0]
28        print(i, df["질문"][i], df["model_answer"][i])

```

11 시영이네 집에 있는 강아지의 다리를 세어 보니 모두 20개였습니다. 시영이네 집에 있는 강아지는 몇 마리입니까? 5
 17 오리 7마리와 염소 몇 마리가 한 울타리에 있습니다. 오리와 염소의 다리 수를 세어 보니 모두 50개입니다. 염소는 몇 마리 있습니까? 9
 19 마당에 강아지, 닭, 염소가 있습니다. 동물들의 다리 수를 세어 보니 모두 70개였습니다. 강아지는 7마리, 닭은 9마리라면, 염소는 몇 마리입니까? 6
 20 진호네 농장에서 염소와 닭을 기르고 있습니다. 염소와 닭의 다리를 세었더니 모두 50개였습니다. 닭이 9마리라면 염소는 몇 마리입니까? 8
 48 아름이네 집에서 기르고 있는 닭, 토끼, 염소의 다리 수를 세어 보니 모두 170개였습니다. 토끼와 염소가 모두 27마리라면, 아름이네 집에서 기르고 있는 닭은 몇 마리입니까? 31

2.3 규칙 기반 모델

< 모델 구조 >

for문 : 전체문장

For 문 : 유형1

For 문 : 유형2

For 문 : 유형3

.

For 문 : 유형n

```
1 #매일 똑같은 수
2 for i in range(0, len(df)):
3     if (re.compile('[Wd]{1,2}[Hw]{1,2}')가 하루에 [Hw]{1,2}["을,를"] [Wd]{1,2}[Hw]{1,2}').findall(df["질문"][i])!=[]):
4         a=re.compile('[Wd]{1,2}[Hw]{1,2}')가 하루에 [Hw]{1,2}["을,를"] [Wd]{1,2}[Hw]{1,2}').findall(df["질문"][i])
5         aa=re.compile('[Wd]{1,2}').findall(a[0])
6         b=df["질문"][i].split("다만")
7         bb=re.compile('[Wd]{1,2}').findall(b[1])
8         df["model_answer"][i]=int(bb[1])//((int(aa[0])*(int(bb[0]))+int(bb[0]))
9
10        print(i, df["질문"][i], df["model_answer"][i])
```

유형 1

6 고양이 8마리가 하루에 쥐를 24마리씩 잡는다고 합니다. 모든 고양이와 매일 똑같은 수의 쥐를 잡는다면, 고양이 3마리가 쥐를 81마리를 잡는 데 며칠이 걸리는지 구하시오. 9
31 토끼 6마리가 하루에 당근을 18개 먹습니다. 모든 토끼가 매일 똑같은 개수의 당근을 먹는다면 토끼 8마리가 당근 72개를 먹는 데에는 며칠이 걸리겠습니까? 3
90 말 4마리가 하루에 당근을 8개 먹습니다. 모든 말이 매일 똑같은 개수의 당근을 먹는다면 말 8마리가 당근 48개를 먹는 데에는 며칠이 걸리겠습니까? 4

```
1 #동전 총 금액
2
3 for i in range(0, len(df)):
4     if (re.compile("동전이 모두 [Wd]{1,4}개").findall(df["질문"][i])!=[]) and (re.compile("[Wd]{1,4}원짜리").findall(df["질문"][i])!=[]):
5         a1=re.compile("동전이 모두 [Wd]{1,4}개").findall(df["질문"][i])
6         a11=re.compile("[Wd]{1,4}").findall(a1[0])
7         a2=re.compile("[Wd]{1,4}원짜리").findall(df["질문"][i])
8         a22=[]
9         for j in range(0, len(a2)):
10            a22.append(int((re.compile("[Wd]{1,4}").findall(a2[j])[0]))
11            a3=re.compile("[Wd]{1,4}종류").findall(df["질문"][i])
12            a33=re.compile("[Wd]{1,4}").findall(a3[0])
13            k=int(a11[0])//int(a33[0])
14
15            df["model_answer"][i]=0
16            for j in range(0, len(a22)):
17                df["model_answer"][i]=df["model_answer"][i]+(int(a22[j])*k)
18
19        print(i, df["질문"][i], df["model_answer"][i])
```

유형 2

85 동전이 모두 12개 있습니다. 동전은 10원짜리, 50원짜리, 100원짜리, 500원짜리로 4종류이고, 종류별 동전의 개수는 모두 같다고 합니다. 동전의 금액을 모두 더하면 얼마입니까? 1980
86 동전이 모두 27개 있습니다. 동전은 10원짜리, 50원짜리, 100원짜리로 3종류이고, 종류별 동전의 개수는 모두 같다고 합니다. 동전의 금액을 모두 더하면 얼마입니까? 1440
85 동전이 모두 18개 있습니다. 동전은 10원짜리, 50원짜리, 100원짜리로 3종류이고, 종류별 동전의 개수는 모두 같다고 합니다. 동전의 금액을 모두 더하면 얼마입니까? 960

3. 결과 - 정답률 측정

덧셈

약 **92%**
(110/120)

뺄셈

약 **97%**
(115/119)

곱셈

약 **90%**
(107/119)

나눗셈

약 **73%**
(87/120)

덧셈 + 뺄셈

약 **87%**
(208/239)

곱셈 + 나눗셈

약 **74%**
(178/239)

< 한계 >

1. 모델을 합쳤을 때 속도가 저하되고 정답률이 떨어지는 것을 개선해야 함
2. 사칙연산에 모두 적용되는 모델을 구현해내야 함

4. 개선방안

< 현재 모델 >

for문 : 전체문장

For 문 : 유형1

For 문 : 유형2

For 문 : 유형3

⋮

For 문 : 유형n

< 개선 모델 >

for문 : 전체문장

덧셈

뺄셈

곱셈

나눗셈

유형1

유형1

유형1

유형1

유형2

유형2

유형2

유형2

⋮

⋮

⋮


⋮

유형k1

유형k2

유형k3

유형k4



감사합니다.