

# ■ 신문사에 따른 기사 분류

- 통계학과 2014150345 이강휘
- 통계학과 2014150365 김경민
- 통계학과 2015150022 서지예

## 개요

---

1. 목적
2. 데이터 설명 및 전처리
3. 1차 분석
4. 2차 분석
5. 결과 해석

## 목적

---

- 사회적 계층에 따라 구독하는 신문이 다른 영국의 경우, 각 종류의 신문사마다 계층의 성향이 묻어난다고 함
- 정치 성향에 따라 신문사가 나뉘는 한국, 정치 외 다른 종류의 기사(사회, 기술)에도 차이가 있는지
- 정치, 사회, 기술 기사에 대해 신문사 분류

# 빅데이터 분석 발표회

## 데이터 설명

```
32 list_baseurl = 'https://news.naver.com/main/list.nhn?mode=LPOD&sid1=' + sid1+ '&mid=shm&oid=' + oid + '&listType=title&date='
33 list_hrefList = []
34
35 directory = os.getcwd()
36 driver = webdriver.Chrome('chromedriver',options=options)
37 driver.implicitly_wait(1)
38
39 print('opening selenium...')
40 for single_date in daterange(start_date, end_date):
41     search_date = single_date.strftime('%Y%m%d')
42     driver.get(list_baseurl+search_date)
43     elements = driver.find_elements_by_xpath('//*[@id="main_content"]/div[2]/ul[*]/li[*]/a')
44     for i, element in enumerate(elements):
45         if i > 5 : break # max 4 articles per day
46         list_hrefList.append(element.get_attribute('href'))
47
48
49 hrefList = np.unique(list_hrefList)
```

```
[14] 1 data_train
```

```
array([[ '유시민의 알릴레오' 에 첫 출연WnWn "국회의원, 수사대상 포함되게 해달라" WnWn "야당 탄압 주장은 황당, 여불
[ '중앙선관위-한겨레 신문 공동기획' WnWn 대선 오명WnWn거리유세 · 토론회 · 정책발표회 금지WnWn후보자, 조함원 물
[ '여의도공원에서 임시정부 수립 100주년 기념식WnWn임정 요인들 고국으로 돌아오는 장면 재현도WnWn임시헌장 낭독
[ '한국 의회 대표단-벨로시 美 하원 의장, 12일 워싱턴DC 국회의사당에서 면담WnWn비핵화 회담 전망 놓고 치열한 공
[ '이해찬 대표 '조월회' 회동서 전해WnWnWnWn여야 5당 대표들과 문희상 국회의장이 4일 낮 국회 사랑재에서 오찬
[ '한국당, " '손혜원 랜드 게이트' , 필요 시 고발도 생각" WnWn손 의원 "검찰 수사 요청할 것. 에스비에스도 수사
[ '연합뉴스TV 캡처 화면WnWn연합뉴스TV가 문재인 대통령의 사진 밑에 북한 인공기를 배치해 논란이 일었다.WnWn연합
[ '[토요판] 뉴스분석 왜WnWn한미훈련 65년...방어전으로 시작 '유사시 정권교체' 성격으로 변모WnWn상황 따라 성격
[ '참석자들 "내년 최저임금 동결해야" "일자리아정자금 신청요건 부담"WnWn文대통령 "최저임금 소상공인 입장 말할
[ '정치BAR_노지원의 진토닉_조선노동당 ABCWnWn김정은 정권서 전권회의 4차례 열려WnWn2013년 경제·핵 무력 병진노
[ '정세현<사진> 전 통일부 장관은 1일 반(反) 김정은 정권 단체인 W'자유조선W'이 스페인 주재 북한대사관을 습격해
[ '지난해 12월 더불어민주당 박영선 의원이 국회의원회관 의원실에서 연합뉴스와 인터뷰를 하고 있다/연합뉴스WnWn
[ '아세안 3개국 순방 중인 문재인 대통령과 부인 김정숙 여사가 14일 오후 마지막 순방지인 캄보디아 프놈펜 국제공
[ '정의용 청와대 국가안보실장./연합뉴스WnWn청와대는 7일 정의용 국가안보실장 주재로 국가안전보장회의(NSC) 상임
[ '시민사회단체 대표 청와대 간담회WnWn' 일자리 문화와 비근로자 가구소득 저하 일리있어.WnWn소득양극화 해소되는
[ '비서실장·정무수석·소통수석 교체WnWn문 대통령은 집권 3년차 '성과 드라이브' WnWn여당 "당정청·야당과 대화중
[ '국회 사무처가 6일 강원도 고성지역 산불 이재민들을 위해 국회 고성연수원 시설을 제공할 계획이라고 밝혔다. 국
[ '당내 일부 의원 "결국 선거제 개편 패스트트랙 처리하겠다는 것"WnWn바른미래당 김관영 원내대표가 20일 오후 국회
[ '문재인 대통령이 18일 태안 화력발전소 컨베이어벨트 사고로 숨진 비정규직 노동자 고(故) 김용균씨의 어머니 김D
[ '세 자녀 유학하는 몬트리올로 공무연수WnWn몰의 일자 '사죄서W' 내고 비용 반납WnWn경기 과천시의회 박상진(더불어
[ '당원자격심사위원회 회의 통해 결론WnWn "민주당 정강정책 맞지 않는 활동 많이 해" WnWn "지난 선거에서 민주당
[ '금강산에서 남북 민간교류 행사WnWn유엔사 '협의 부족' 문제제기에WnWn통일부, 소극적 대응... "노트북 금지" Wn
```

- Selenium 사용
- 네이버 뉴스 크롤링

- 경향, 한겨레, 동아, 조선
- 기술, 사회, 정치, 500개씩

총 6000개의 기사

# 빅데이터 분석 발표회

## 데이터 전처리

[14] 1 data\_train

```
array([[ '유시민의 알릴레오' 에 첫 출연WnWn "국회의원, 수사대상 포함되게 해달라" WnWn "야당 탄압 주장은 황당, 어불  
[ '중앙선관위-한겨레 신문 공동기획] WnWn 돈선거 오명 WnWn 거리유세 · 토론회 · 정책발표회 금지 WnWn 후보자, 조합원 몰  
[ '여의도공원에서 임시정부 수립 100주년 기념식 WnWn 임정 요인들 고국으로 돌아오는 장면 재현도 WnWn 임시헌장 낭독  
[ '한국 의회 대표단-펠로시 美 하원 의장, 12일 워싱턴DC 국회의사당에서 연담 WnWn 비핵화 회담 전망 놓고 치열한 공  
[ '이해찬 대표 '초월회' 회동서 전해 WnWn WnWn 여야 5당 대표들과 문희상 국회의장이 4일 낮 국회 사랑채에서 오찬  
[ '한국당, " "손혜원 랜드 게이트", 필요 시 고발도 생각" WnWn 손 의원 "검찰 수사 요청할 것. 에스비에스도 수사  
[ '연합뉴스TV 캡처 화면 WnWn 연합뉴스TV가 문재인 대통령의 사진 밑에 북한 인공기를 배치해 논란이 일었다. WnWn 연합  
[ '토요판] 뉴스분석 왜 WnWn 한미훈련 65년...방어전으로 시작 '유사시 정권교체' 성격으로 변모 WnWn 상황 따라 성격  
[ '참석자들 "내년 최저임금 동결해야" "일자리안정자금 신청요건 부담" WnWn 文대통령 "최저임금 소상공인 입장 말할  
[ '정치BAR_노지원의 진토닉_조선노동당 ABC WnWn 김정은 정권서 전원회의 4차례 열려 WnWn 2013년 경제 · 핵 무력 병진해  
[ '정세현<사진> 전 통일부 장관은 1일 반(反) 김정은 정권 단체인 W' 자유조선W'이 스페인 주재 북한대사관을 습격해  
[ '지난해 12월 더불어민주당 박영선 의원이 국회의원회관 의원실에서 연합뉴스와 인터뷰를 하고 있다/연합뉴스 WnWn  
[ '아세안 3개국 순방 중인 문재인 대통령과 부인 김정숙 여사가 14일 오후 마지막 순방지인 캄보디아 프놈펜 국제공  
[ '정의용 청와대 국가안보실장 /연합뉴스 WnWn 청와대는 7일 정의용 국가안보실장 주재로 국가안전보장회의(NSC) 상임  
[ '시민사회단체 대표 청와대 간담회 WnWn' 일자리 문화와 비교로자 가구소득 저하 알리었어, WnWn 소득 양극화 해소되는
```

[16] 1 x\_in\_train

```
array([[ '유시민 레오 첫 출연 국회의원 수사 대상 포함 야당 탄압 주장 황당 불성 설 처 임명 야당 개입 수 조국 청와대 민정 수  
' 중앙 선관위 한겨레 신문 공동 기획 돈 선거 오명 거리 유세 토론회 정책 발표회 금지 후보자 조합원 표 매수 성행 선거  
' 여의도 공원 임시정부 수립 100주년 기념 임정 요 고국 장면 재현 임시 헌장 낭독 축하공연 총리 대한민국 임시정부 뿌리 위  
' 한국 의회 대표 단 펠 로시 하원 의장 워싱턴 국회의사당 연담 비핵화 회담 전망 공방전 예정 훌쩍 면담 종료 펠 로시 전  
' 이해찬 대표 초월 회 회 동서 전해 여야 당 대표 문희상 국회의장 낮 국회 사랑 채 오찬 간담 회 전 기념 사진 왼쪽 정미  
' 한국 손 혜원 랜드 게이트 필요 시 고발 생각 손 의원 검찰 수사 요청 에스비에스 수사 손 혜원 민주당 의원 유튜브 채널  
' 연합뉴스 캡처 화면 연합뉴스 문재인 대통령 사진 밑 북한 인공 배치 논란 연합뉴스 오후 뉴스 한미정상회담 참석 차 방  
' 토요판 뉴스 분석 한미 훈련 방어전 시작 유사시 정권 교체 성격 변모 상황 성격 규모 초기 방어 성격 점차 공격 세계 최  
' 참석자 내년 최저임금 동결 일자리 안정 자금 신청 요건 부담 대통령 최저임금 소상공인 입장 말 수 최저임금 인상 필요  
' 정치 노지원 진토닉 조선노동당 김정은 정권 전원 회의 차례 경제 핵 무력 병진 노선 채택 경제 건설 집중 새 전략 노선  
' 정세현 사진 전 통일부 장관 반 김정은 정권 단체 자유 조선 스페인 주재 북한 대사관 습격 정보 연방 사국 관련 미국 지  
' 지난해 민주당 박영선 의원 국회의원 회관 의원 실 연합뉴스 인터뷰 연합뉴스 박영선 민주당 의원 공직 선거 여성 공천  
' 아세안 개국 순방 중인 문재인 대통령 부인 김정숙 여사 오후 마지막 순방 지인 캄보디아 프놈펜 국제공항 도착 순 인사  
' 정의용 청와대 국가안보실 연합뉴스 청와대 정의용 국가안보실 주재 국가 안전보장 회의 상임 위원회 미국 북한 간 비핵  
' 시민사회단체 대표 청와대 간담 회 일자리 문화 비 근로자 가구 소득 저하 알리 소득 양극화 해소 회안 전망 구축 더 노력
```

- Okt 사용
- 명사 토큰화
- 일반적인 불용어

(<https://www.ranks.nl/stopwords/Korean>) + 따로 선정한 불용어(뉴스 한정, 특수문자)

# 한겨레

# 빅데이터 분석 발표회

## 사회

한국, 미국, 세계, 자동차, 공장, 현대차, 모델, 시작, 사회, 정도, 최근, 개, 안, 일본, 중국, 국내, 서울, 대전, 자동차, 가장, 위해, 사진, 시작, 판매, 생산, 이야기, 자동차, 가장, 위해, 사진, 시작, 판매

한국, 미국, 세계, 자동차, 공장, 현대차, 모델, 시작, 사회, 정도, 최근, 개, 안, 일본, 중국, 국내, 서울, 대전, 자동차, 가장, 위해, 사진, 시작, 판매, 생산, 이야기, 자동차, 가장, 위해, 사진, 시작, 판매

조선

동아

한국, 미국, 세계, 자동차, 공장, 현대차, 모델, 시작, 사회, 정도, 최근, 개, 안, 일본, 중국, 국내, 서울, 대전, 자동차, 가장, 위해, 사진, 시작, 판매, 생산, 이야기, 자동차, 가장, 위해, 사진, 시작, 판매

한국, 미국, 세계, 자동차, 공장, 현대차, 모델, 시작, 사회, 정도, 최근, 개, 안, 일본, 중국, 국내, 서울, 대전, 자동차, 가장, 위해, 사진, 시작, 판매, 생산, 이야기, 자동차, 가장, 위해, 사진, 시작, 판매

경향

한겨레

# 빅데이터 분석 발표회

## 기술

스마트폰 게임 기업 지난해 작년  
세계 약 국내 현재 개 시장  
기술 한국 미 한국 서비스  
삼성 업체 중국 특허 제품 신청  
데이터 관 돈 중국 반도체 제품 사업

때문 제공 진행 기존 데이터  
기업 인공 사업 지원 시작 화면 중  
클라우드 국내 전문 사용자 동해 로영  
서비스 정도 시장 경우 제품  
출시 주요 가장 위해 공 스마트폰 한국 콘텐츠

조선

동아

출시 문제 지난해 공인 게임 스마트폰 기업 테레콤  
서비스 과학 전문 우주 이용 중  
때문 달 개발 연구 제품 기존 지구로 중  
만화 시장 한국 유독스 플랫폼  
택시 업체 세계 화면 한국 점 플랫폼

개발 설명 에스케이텔레콤 스마트폰 시작 통신 엘지  
지능 동해 사회 결과  
인공 지능 달 사람 사업 공공 연구 위해  
세계 테 한국 이동통신 공 제품 정부 가장  
문제 지난해 한국 제품

경향

한겨레



## 1차 분석

---

- LDA
- Doc2Vec을 이용한 문서분류

# 빅데이터 분석 발표회

## LDA

```
[ ] 1 lda.show_topics()
```

```
↳ [(0,  
    '0.010*"의원" + 0.008*"한국" + 0.006*"정부" + 0.005*"김" + 0.005*"민주당" + 0.005*"대표" + 0.005*"국회" + 0.005*"후보" + 0.004*"선거" + 0.004*"대해"',  
    1,  
    '0.014*"대통령" + 0.008*"북한" + 0.008*"전" + 0.007*"미국" + 0.006*"한국" + 0.006*"북" + 0.005*"트럼프" + 0.005*"청와대" + 0.005*"한겨레" + 0.005*"국회"',  
    2,  
    '0.007*"의원" + 0.007*"한국" + 0.007*"북한" + 0.006*"전" + 0.006*"미국" + 0.006*"정부" + 0.005*"금지" + 0.005*"날" + 0.004*"대표" + 0.004*"대통령"',  
    3,  
    '0.014*"의원" + 0.014*"한국" + 0.011*"전" + 0.007*"당" + 0.007*"자유" + 0.007*"대표" + 0.006*"대통령" + 0.006*"날" + 0.005*"장관" + 0.005*"김"',  
    4,  
    '0.018*"의원" + 0.012*"한국" + 0.008*"대표" + 0.007*"대통령" + 0.007*"자유" + 0.006*"당" + 0.006*"전" + 0.005*"한겨레" + 0.005*"선거" + 0.005*"민주당"',  
    5,  
    '0.011*"대표" + 0.010*"의원" + 0.010*"한국" + 0.006*"대해" + 0.006*"전" + 0.006*"북한" + 0.006*"대통령" + 0.005*"국회" + 0.005*"민주당" + 0.004*"북"',  
    6,  
    '0.010*"북한" + 0.009*"대통령" + 0.008*"의원" + 0.006*"후보자" + 0.006*"정상회담" + 0.006*"북" + 0.006*"위" + 0.005*"청와대" + 0.005*"한국" + 0.005*"회의"',  
    7,  
    '0.009*"대통령" + 0.007*"전" + 0.007*"한국" + 0.006*"청와대" + 0.006*"의원" + 0.005*"북한" + 0.005*"국회" + 0.004*"금지" + 0.004*"민주당" + 0.004*"당"',  
    8,  
    '0.011*"의원" + 0.009*"한국" + 0.007*"후보" + 0.007*"대표" + 0.006*"영상" + 0.005*"위" + 0.005*"당" + 0.004*"청와대" + 0.004*"한겨레" + 0.004*"날"',  
    9,  
    '0.017*"대통령" + 0.009*"전" + 0.007*"한국" + 0.007*"민주당" + 0.006*"의원" + 0.006*"북한" + 0.006*"문" + 0.006*"김" + 0.006*"날" + 0.005*"대표"')]
```

- Topic이 잘 나뉘지 않음

## Document embedding



- Doc2Vec 임베딩 기법을 사용
- 토큰화된 각각의 신문기사들을 길이 50의 수치형 벡터로 변환
- Logistic Regression, SVM, Naïve Bayes

# 빅데이터 분석 발표회

## Document embedding

### 정치(1.0)

```
#see how much documents are the most similar to themselves, according to our doc2vec model
#(Testing the performance of our doc2vec model)

correct_list = []
wrong_list = []
wrong_id_list = []
for doc_id in range(len(train_corpus)):
    v = model.infer_vector(train_corpus[doc_id].words)
    sims = model.docvecs.most_similar([v])
    dif = int(sims[0][0]) - doc_id
    if dif==0:
        correct_list.append(sims)
    else:
        wrong_list.append(sims)
        wrong_id_list.append(doc_id)

print(len(correct_list) / len(train_corpus), ' correct')
```

1.0 correct

### 사회(0.95)

```
#see how much documents are the most similar to themselves, according to our doc2vec model
#(Testing the performance of our doc2vec model)

correct_list = []
wrong_list = []
wrong_id_list = []
for doc_id in range(len(train_corpus)):
    v = model.infer_vector(train_corpus[doc_id].words)
    sims = model.docvecs.most_similar([v])
    dif = int(sims[0][0]) - doc_id
    if dif==0:
        correct_list.append(sims)
    else:
        wrong_list.append(sims)
        wrong_id_list.append(doc_id)

print(len(correct_list) / len(train_corpus), ' correct')
```

0.9571428571428572 correct

### 기술(0.98)

```
#see how much documents are the most similar to themselves, according to our doc2vec model
#(Testing the performance of our doc2vec model)

correct_list = []
wrong_list = []
wrong_id_list = []
for doc_id in range(len(train_corpus)):
    v = model.infer_vector(train_corpus[doc_id].words)
    sims = model.docvecs.most_similar([v])
    dif = int(sims[0][0]) - doc_id
    if dif==0:
        correct_list.append(sims)
    else:
        wrong_list.append(sims)
        wrong_id_list.append(doc_id)

print(len(correct_list) / len(train_corpus), ' correct')
```

0.9985714285714286 correct

# 빅데이터 분석 발표회

## Test set performance (Logistic Regression, SVM)

정치(0.30, 0.35)

```
#with test set
X_test = getDocVec(model2)
Y_test = label_test[:,0]

X_test = getDocVec(model2)
y_test_lm = lmfit.predict(X_test)
y_test_svm = svmfit.predict(X_test)

col_lm = collections.Counter(y_test_lm - Y_test)
col_svm = collections.Counter(y_test_svm - Y_test)
print('Logistic Regression Performance with Test set : ', col_lm[0]/len(y_test_lm))
print('Support Vector Machine Performance with Test set : ', col_svm[0]/len(y_test_svm))
```

Logistic Regression Performance with Test set : 0.30666666666666664  
Support Vector Machine Performance with Test set : 0.35166666666666667

사회(0.34, 0.38)

```
[45] #with test set
X_test = getDocVec(model2)
Y_test = label_test[:,0]

X_test = getDocVec(model2)
y_test_lm = lmfit.predict(X_test)
y_test_svm = svmfit.predict(X_test)

col_lm = collections.Counter(y_test_lm - Y_test)
col_svm = collections.Counter(y_test_svm - Y_test)
print('Logistic Regression Performance with Test set : ', col_lm[0]/len(y_test_lm))
print('Support Vector Machine Performance with Test set : ', col_svm[0]/len(y_test_svm))
```

Logistic Regression Performance with Test set : 0.34  
Support Vector Machine Performance with Test set : 0.37833333333333335

기술(0.37, 0.39)

```
[65] #with test set
X_test = getDocVec(model2)
Y_test = label_test[:,0]

X_test = getDocVec(model2)
y_test_lm = lmfit.predict(X_test)
y_test_svm = svmfit.predict(X_test)

col_lm = collections.Counter(y_test_lm - Y_test)
col_svm = collections.Counter(y_test_svm - Y_test)
print('Logistic Regression Performance with Test set : ', col_lm[0]/len(y_test_lm))
print('Support Vector Machine Performance with Test set : ', col_svm[0]/len(y_test_svm))
```

Logistic Regression Performance with Test set : 0.37166666666666665  
Support Vector Machine Performance with Test set : 0.39166666666666666

## Test set performance(Naïve Bayes)

### 정치(0.34)

```
#NB with test set politics
X_test = getDocVec(model2)
Y_test = label_test[:,0]
y_test_gnb = gnbfit.predict(X_test)

col_gnb = collections.Counter(y_test_gnb - Y_test)
print('NB Performance with Test set : ', col_gnb[0]/len(y_test_gnb))
```

NB Performance with Test set : 0.34

### 사회(0.35)

```
#NB society with test set
X_test = getDocVec(model2)
Y_test = label_test[:,0]
y_test_gnb = gnbfit.predict(X_test)

col_gnb = collections.Counter(y_test_gnb - Y_test)
print('NB Performance with Test set : ', col_gnb[0]/len(y_test_gnb))
```

NB Performance with Test set : 0.3516666666666667

### 기술(0.41)

```
#NB with test set tech
X_test = getDocVec(model2)
Y_test = label_test[:,0]
y_test_gnb = gnbfit.predict(X_test)

col_gnb = collections.Counter(y_test_gnb - Y_test)
print('NB Performance with Test set : ', col_gnb[0]/len(y_test_gnb))
```

NB Performance with Test set : 0.4133333333333333

## 2차 분석

### Text CNN

```
[8] 1 ## Customize FLAGS, because I don't find the tf flags easy to use :-(
    2 class Flags():
    3     def __init__(self):
    4         self.training_sample_percentage = float(0.7)
    5         self.max_doc_length = int(350)
    6
    7         self.embedding_dim = int(64)
    8         self.filter_sizes = str('3,4,5')
    9         self.num_filters = int(128)
   10         self.dropout_keep_prob = float(0.5)
   11         self.l2_reg_lambda = float(0.0)
   12         self.learning_rate = float(1e-3)
   13
   14         self.batch_size = int(64)
   15         self.num_epochs = int(50)
   16
   17 FLAGS = Flags()
```

## 2차 분석

### RNN

```
[26] 1 ## Customize FLAGS, because I don't find the tf flags easy to use :-(
      2 class Flags():
      3     def __init__(self):
      4         self.training_sample_percentage = float(0.7)
      5
      6         self.n_step = int(350) # how many words there will be per document
      7         self.n_hidden = int(32) # number of recurrent steps
      8
      9         self.embedding_dim = int(64)
     10         self.learning_rate = float(1e-3)
     11         self.batch_size = int(64)
     12         self.num_epochs = int(50)
     13         self.cell_type = int(1)
     14
     15
     16 FLAGS = Flags()
```




# 빅데이터 분석 발표회

## Text CNN

### 정치(0.8933)


Epoch: 0047 Avg. cost = 0.157 Avg. accuracy = 0.971  
Epoch: 0048 Avg. cost = 0.145 Avg. accuracy = 0.970  
Epoch: 0049 Avg. cost = 0.134 Avg. accuracy = 0.964  
Epoch: 0050 Avg. cost = 0.114 Avg. accuracy = 0.976  
Training Complete!

 1 test(x\_test, y\_test, session, cnnModel)

 Test set Accuracy : 0.8933333

### 기술(0.8933)


Epoch: 0047 Avg. cost = 0.070 Avg. accuracy = 0.981  
Epoch: 0048 Avg. cost = 0.092 Avg. accuracy = 0.977  
Epoch: 0049 Avg. cost = 0.093 Avg. accuracy = 0.978  
Epoch: 0050 Avg. cost = 0.080 Avg. accuracy = 0.981  
Training Complete!

 1 test(x\_test, y\_test, session, cnnModel)

 Test set Accuracy : 0.8933333

### 사회(0.8417)

Epoch: 0047 Avg. cost = 0.108 Avg. accuracy = 0.971  
Epoch: 0048 Avg. cost = 0.074 Avg. accuracy = 0.979  
Epoch: 0049 Avg. cost = 0.099 Avg. accuracy = 0.974  
Epoch: 0050 Avg. cost = 0.107 Avg. accuracy = 0.973  
Training Complete!

 1 test(x\_test, y\_test, session, cnnModel)


 Test set Accuracy : 0.8416664

# 빅데이터 분석 발표회

## RNN

### 정치(0.6383)


Epoch: 0047 Avg. cost = 0.009 Avg. accuracy = 1.000  
Epoch: 0048 Avg. cost = 0.008 Avg. accuracy = 1.000  
Epoch: 0049 Avg. cost = 0.008 Avg. accuracy = 1.000  
Epoch: 0050 Avg. cost = 0.007 Avg. accuracy = 1.000  
Training Complete!

 1 test(x\_test, y\_test, session, rnnModel)

 Test set Accuracy : 0.6383333

### 사회(0.47)


Epoch: 0047 Avg. cost = 0.009 Avg. accuracy = 1.000  
Epoch: 0048 Avg. cost = 0.008 Avg. accuracy = 1.000  
Epoch: 0049 Avg. cost = 0.008 Avg. accuracy = 1.000  
Epoch: 0050 Avg. cost = 0.007 Avg. accuracy = 1.000  
Training Complete!

 1 test(x\_test, y\_test, session, rnnModel)

 Test set Accuracy : 0.47

### 기술(0.5967)

Epoch: 0047 Avg. cost = 0.009 Avg. accuracy = 1.000  
Epoch: 0048 Avg. cost = 0.008 Avg. accuracy = 1.000  
Epoch: 0049 Avg. cost = 0.008 Avg. accuracy = 1.000  
Epoch: 0050 Avg. cost = 0.007 Avg. accuracy = 1.000  
Training Complete!

 1 test(x\_test, y\_test, session, rnnModel)

 Test set Accuracy : 0.5966667

## 결과

---

- RNN은 overfitting되는 것으로 보임
- CNN의 성능이 가장 좋음
- 정치 뿐만 아니라 다른 종류의 기사도 신문사에 따른 분류가 가능