

# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	길 JOB 이
팀 명	4726
문서 제목	중간보고서

Version	2.0
Date	2019-04-18

팀원	고현경 (조장)
	이선홍
	이수민
	김혜인
	김희주
지도교수	윤종영 교수

### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 "길 JOB 이"를 수행하는 팀 "4726"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "4726"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.



국민대학교  
컴퓨터공학부  
캡스톤 디자인 I

## 중간보고서

프로젝트 명

길 JOB 이

팀 명

4726

Confidential Restricted


Version 2.0

2019-APR-18

## 문서 정보 / 수정 내역


Filename	중간보고서-길 JOB 이.doc
원안작성자	고현경, 김희주, 김혜인, 이선희, 이수민
수정작업자	고현경, 김희주, 김혜인, 이선희, 이수민

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-11	고현경	1.0	최초 작성	
2019-04-12	김희주 김혜인 이선희	1.1	내용 추가	수정된 연구내용 추가
2019-04-13	고현경 이수민	1.2	내용 수정	향후 추진 계획 수정
2019-04-15	김희주 김혜인 이선희	1.3	내용 수정	수행 내용 수정
2019-04-17	김희주	1.4	내용 수정	수행 내용 추가 수정
2019-04-18	고현경	2.0	내용 수정	전체적인 수정 및 마무리

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

## 목 차

1	프로젝트 목표	4
2	수행내용 및 중간 결과	4
	2.1 계획서 상의 연구 내용	4
	2.1.1 서버	4
	2.1.2 자연어 처리	7
	2.2 수행내용	9
	2.2.1 서버	9
	2.2.2 자연어처리	15
	2.2.3 결과물 목록 및 진행사항	19
3	수정된 연구내용 및 추진방향	21
	3.1 수정사항	21
	3.1.1 서버	21
	3.1.2 자연어처리	22
	3.1.3 개발일정	24
4	향후 추진계획	25
	4.1 향후 계획의 세부 내용	25
	4.1.1 서버	25
	4.1.2 자연어처리	26
5	고충 및 건의사항	28

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

## 1 프로젝트 목표

본 프로젝트는 텍스트 마이닝을 통해 자기소개서를 작성하는 취업 준비생들에게 객관적인 자기소개서 분석 서비스를 제공하는 것을 목표로 한다. 단순히 합격과 불합격을 판가름하는 것이 아니라 사용자의 강점과 약점을 분석하여 취업준생들이 자기소개서를 작성하는데 도움이 되는 것이 목적이다.

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 연구내용

Web Application 에서 사용자는 로그인 한 뒤 자신의 자기소개서를 웹페이지에 입력한다. 입력된 자기소개서는 자연어 처리 Application 으로 전송되어 자연어처리를 통해 피쳐벡터가 생성된다. 이 피쳐벡터는 자연어처리 Application 에서 미리 저장된 기업별 피쳐벡터, 직무별 피쳐벡터와 유사도를 비교하여 수치화 한다.

이 수치화 된 데이터를 웹페이지 에서 차트와 텍스트로 사용자에게 제공한다.

자세한 개발 내용은 다음과 같다.

#### 2.1.1 서버

- 1) 회원가입/로그인
  - 길 JOB 이 서비스에서는 Springboot, Spring security 및 OAuth 를 통해 구축한 SSO(Single Sign On)을 통해 회원 시스템을 관리한다. 회원 시스템 SSO 를 통해 클라이언트가 회원가입 시 자신의 구글 계정을 인증하는 것만으로 가입이 완료되고 회원 정보가 DB 에 저장되어 로그인 가능하게 된다. 인증된 계정으로 로그인 하면 access token 을 발급 받아 자기소개서 분석 서비스를 이용할 수 있는 권한을 가지게 된다.



## 2) 자기소개서 입력

- 사용자는 자기소개서를 분석하기 위해 웹 페이지에 사용자의 자기소개서를 제출하게 된다. 제출된 자기소개서는 `JSONParser parser = new JSONParser(); Object obj = parser.parse(jsonStr)` 메소드를 통해 Json 타입으로 변환되어 AWS DynamoDB 에 저장된다. 자연어처리 Application Server 는 자연어처리 request 를 받게 되면 DynamoDB 에 저장 되어 있는 자기소개서를 다운 받아 자연어 처리를 진행하게 된다.

## 3) 분석 / 결과 페이지

- 사용자가 입력한 자기소개서에 대한 분석 결과를 자연어 처리 Application 에서 Web Application 로 전달하게 된다. 자연어처리 Application 에서 Web Server 로 전달하게 될 데이터는

- (1) 사용자의 자소서에 적합한 기업순위,
- (2) 사용자가 선택한 **직무**의 핵심역량별 적합도,
- (3) 사용자가 선택한 **기업**의 핵심역량별 적합도,
- (4) (2), (3)과 비교될 사용자 자기소개서의 핵심역량별 적합도이다.

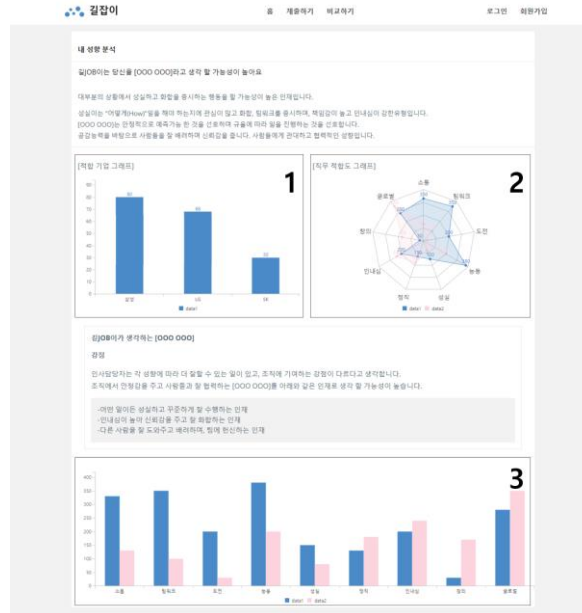
기존에 기업별, 직무별 벡터가 저장되어있지만 (2)과 (3)을 각각 전달하는 이유는 직무와 기업 벡터와 비교할 때 각각 다른 보정치를 사용하기 때문이다. 이 때, 10 가지 핵심역량은 '글로벌, 소통, 팀워크, 도전, 능동, 성실, 정직, 인내심, 창의, 글로벌, 주인 의식'으로 구성된다.

기존에 분석한 기업별, 직무별 유사도 벡터의 데이터는 JPA DB 에 저장되어 있고 사용자의 자기소개서를 분석한 벡터 데이터는 자연어처리 AWS Dynamo DB 에 저장되어 결과페이지 chart 에 나타나게 된다. 저장된 DB 데이터들은 Ajax 를 통해 View 와 연결된다.


사용자에게 제공 될 View 는 아래 첨부 한 그림과 동일하다.



중간보고서		
프로젝트 명	길 JOB 이	
팀 명	4726	
Confidential Restricted	Version 2.0	2019-APR-18



- [1] 적합 기업 그래프 : 사용자가 입력한 자기소개서가 어떤 기업에 적합한지 상위 3 개 순으로 기업을 보여준다.
- [2] 직무 적합도 그래프 : 사용자의 자기소개서와 선택한 직무와의 적합도를 10 가지 핵심역량별 수치로 나타낸다.
- [3] 사용자 선택 기업 적합도 : 사용자의 자기소개서와 선택한 기업과의 적합도를 핵심역량별 수치로 나타낸다.
- 위에 나타난 차트들은 모두 billboard.js 를 사용하여 view 를 작성하였다.
- 자연어 처리 서버에서 자기소개서를 분석한 결과데이터를 DB 에 저장하여 기록을 조회하거나 비교할 수 있게 만든다. DB 에서 결과 데이터를 웹 서버로 전송하면 controller 가 받아 분석 결과에 대한 정보를 view 로 넘긴다. view 는 받은 데이터를 이용하여 웹페이지에 분석 결과를 시각적으로 보여준다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

#### 4) 비교 페이지

- 사용자가 자기소개서 분석 서비스를 이용하면 제출한 자기소개서에 대한 분석 결과(직무 적합도, 기업 적합도, 추천 기업 순위)가 DB 에 저장된다. 사용자는 자신의 분석결과 기록 2 개를 선택하여 비교 결과를 확인 할 수 있다.

### 2.1.2 자연어 처리

합격한 자기소개서의 특징을 분석해내려면 자연어처리가 반드시 필요하다. 자연어처리의 과정은 크게 데이터 수집단계, 데이터 전처리 단계, 데이터 벡터화 단계 3 단계로 나뉜다.

#### 1) 데이터 수집 단계

자기소개서 분석을 위한 데이터 수집을 진행한다. 데이터 수집은 필요한 데이터를 웹사이트 등에서 추출하는 단계이다. 수집할 데이터의 종류는 합격 자기소개서와 자기소개서 성향 분석을 위해 정한 10 개 핵심역량에 대한 문장이다. 자연어처리에 사용될 학습데이터는 정확성과 신뢰성이 중요하므로 정보의 출처와 수집할 데이터의 종류를 신중하게 선택해야 한다.

데이터 수집은 웹 크롤링을 통해 이루어지는데 python 라이브러리인 beautiful soup 이나 selenium 을 사용한다. beautiful soup 은 사용자의 행동을 특정해서 데이터를 가져올 때 빠르게 처리할 수 있다는 장점이 있고 selenium 은 브라우저를 직접 동작시키기 때문에 그보다는 느리지만 사용자의 행동을 동적으로 처리할 수 있다.

#### 2) 데이터 전처리 단계

전처리는 수집된 방대한 데이터들을 분석할 수 있도록 정제하는 단계이다. 수집된 데이터들은 불필요한 어미, 조사 등을 포함하고 있어 분석에 영향을 미치게 된다. 따라서 데이터를 분석하기 전 정제를 해야하고 정제 단계는 크게 4 단계로 구분된다. 각 단계는 1) 데이터 토큰화, 2) 데이터 정규화, 3) 데이터 어근화, 4) 불용어 제거 순으로 진행된다.

수집한 원문 데이터는 첫 번째로 토큰화(tokenization)를 진행하여 데이터들을 품사별로 분리한다. 한글 토큰화는 일반적으로 Konlpy 라이브러리를 많이 활용한다. 이러한 라이브러리를 활용하게 되면 "안녕하세요, 반갑습니다" 같은 문장은 안녕(일반명사 ,NNG) 하(동사 파생 접미사, XSV) 세요(종결어미, EF) ,(심표, SP) 반갑(형용사, VA) 습니다(종결어미, EF) ,(마침표, SP)와 같이 분석된다. 토큰화된 데이터에서 분석에 필요한 품사(단어(명사), 동사)




위주로 추출하여 표현방법이 다른 단어들을 통합시키는 정규화, 의미를 담고 있는 부분을 원형으로 바꿔주는 어근화 단계를 진행한다.

데이터를 정제하는데 적용되는 또다른 과정중에 하나는 n-gram 언어모델이다. n-gram 언어모델은 문장에서 어휘의 위치를 고려하여 의미를 파악하는 언어모델링 기법이다. 어휘의 위치는 문장에서 의미를 바꿀 수 있다. 예를 들어 '나는 수학은 싫고 국어는 좋다'와 '나는 국어는 싫고 수학이 좋다'라는 두 문장은 의미가 정반대인 문장이다. 하지만 n-gram 언어 모델이 아닌 보통의 'Bag of words'의 기법으로 두 문장을 분석하게 되면 쓰인 어휘가 완전히 같기 때문에 두 문장은 동일한 문장이 된다. 하지만 n-gram 이라는 연속된 n 개의 단어를 하나의 분석 단위로 두고 분석하는 개념을 적용하게 되면 더 많은 맥락적인 정보를 얻을 수 있게 된다. 예를 들어 '나는 수학은 싫고 국어는 좋다'에 2-gram 언어 모델을 적용하게 되면 [['나는', '수학은'], ['수학은', '싫고'], ['싫고', '국어는'], ['국어는', '좋다']]와 같은 결과를 확인할 수 있다. 분석하는 데이터양이 많아질수록 더욱 정확한 정보를 얻을 수 있으며 이 언어모델은 데이터의 맥락적인 정보를 파악하는데 도움이 된다. 마지막으로 데이터를 정제하고 최종적으로 분석에 큰 의미가 없는 불용어를 제거하고 나면 데이터를 분석하기 위한 전처리 단계가 완료된다.

### 3) 데이터 벡터화 단계

데이터 벡터화 단계란 데이터를 분석한 결과를 수치로 나타내어 비교 분석이 가능하도록 하는 단계이다. 중요하게 연관된 정도는 단순히 빈도수로 정의하긴 어려우므로 텍스트마이닝에서 사용되는 TF-IDF 를 사용하여 나열하게 된다. TF-IDF 는 출현빈도를 사용하여 어떤 단어가 문서 내에서 얼마나 중요한지 나타내는 수치이다. 이 수치가 높을수록 단어는 문서를 대표하는 성격을 띠게 된다고 볼 수 있다. TF-IDF 의 TF 는 Term Frequency 로 단어의 문서 내에 출현한 횟수를 의미한다. 그리고 IDF 는 Inverse Document Frequency 로 그 단어가 출현한 문서의 숫자의 역수(inverse)를 의미한다. TF 는 단어가 문서 내에서 출현한 횟수이다. 따라서 그 숫자가 클수록 문서 내에서 중요한 단어일 확률이 높다. 하지만, 'the'와 같은 단어도 TF 값이 매우 클 것이다. 하지만 'the'가 중요한 경우는 거의 없으므로 이때 IDF 가 필요하다. DF 는 그 단어가 출현한 문서의 숫자를 의미 하므로, 그 값이 클수록 'the'와 같이 일반적으로 많이 쓰이는 단어일 가능성이 높다. 따라서 IDF 를 구해 TF 에 곱해줌으로써, 'the'와 같은 단어들에 대한 패널티를 준다. 최종적으로 우리가 얻는 숫자는, 다른 문서들에서는 잘 나타나지 않지만, 특정 문서에서만 잘 나타난 경우에 횟수가 높아지기 때문에, 특정 문서에서 얼마나 중요한 역할을 차지하는지 나타내는 수치가 될 수 있다.



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

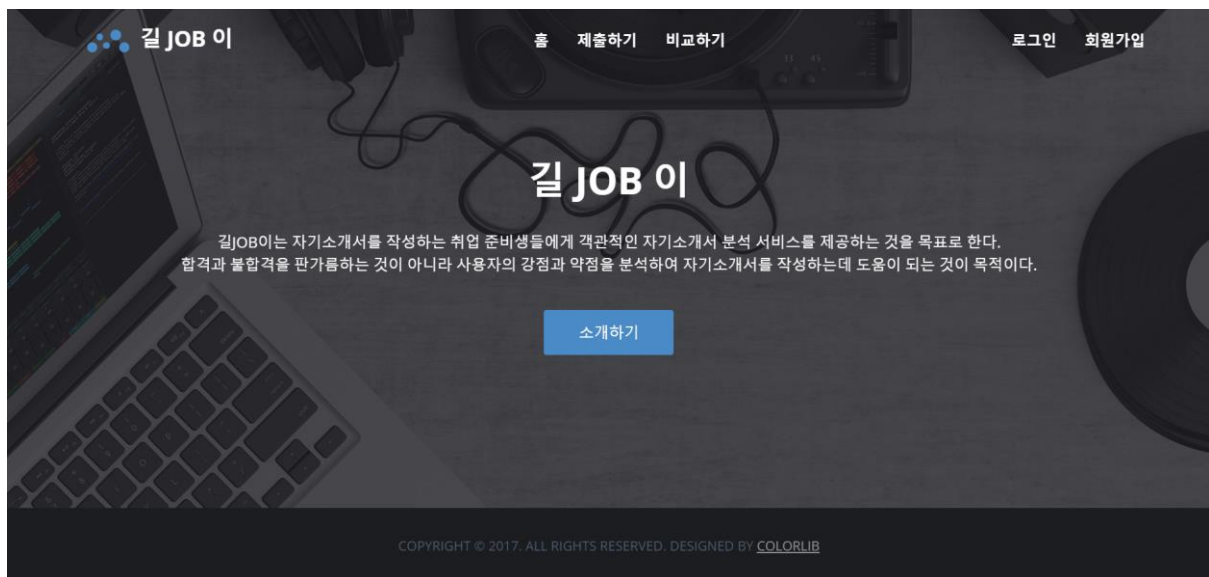
학습시키는 문서별로 tf-idf 값을 행렬의 형태로 나타내어 하나의 벡터를 이루게 된다.

## 2.2 수행내용

### 2.2.1 서버


#### 1) view

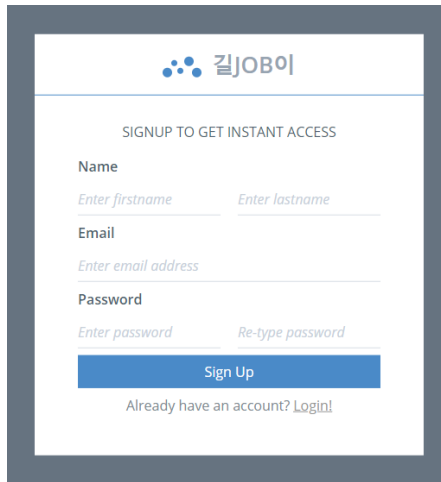
웹 페이지는 반응형 웹 프론트엔드 프레임워크인 부트스트랩을 이용하여 모바일, 웹 및 다양한 브라우저를 사용하는 클라이언트들이 사용할 수 있도록 디자인하였다. 현재까지 [메인 페이지, 자기소개서 제출 페이지, 분석 로딩 페이지, 분석 결과 페이지, 분석 결과 비교 선택 페이지, 분석 결과 비교 결과 페이지, 회원가입 페이지, 로그인 페이지] 가 완성되었다.



[메인 페이지]

헤더 부분에는 navigation bar 로 페이지간 이동이 편리하도록 구성하였고 화면 중앙 부분의 소개하기 버튼을 클릭하면 자기소개서 분석 서비스를 설명하는 소개 페이지가 보여지게 된다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18



**길JOB이**

SIGNUP TO GET INSTANT ACCESS

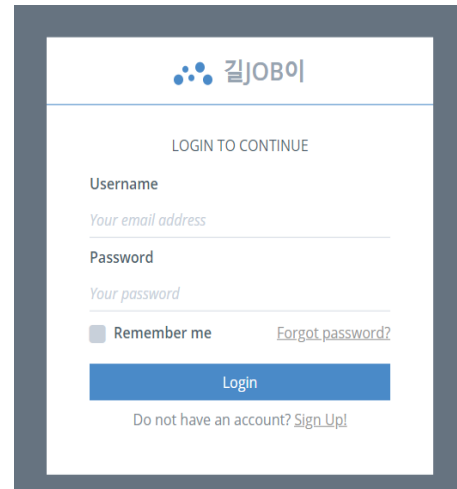
**Name**  
 Enter firstname      Enter lastname

**Email**  
 Enter email address

**Password**  
 Enter password      Re-type password

**Sign Up**

Already have an account? [Login!](#)



**길JOB이**

LOGIN TO CONTINUE

**Username**  
 Your email address

**Password**  
 Your password

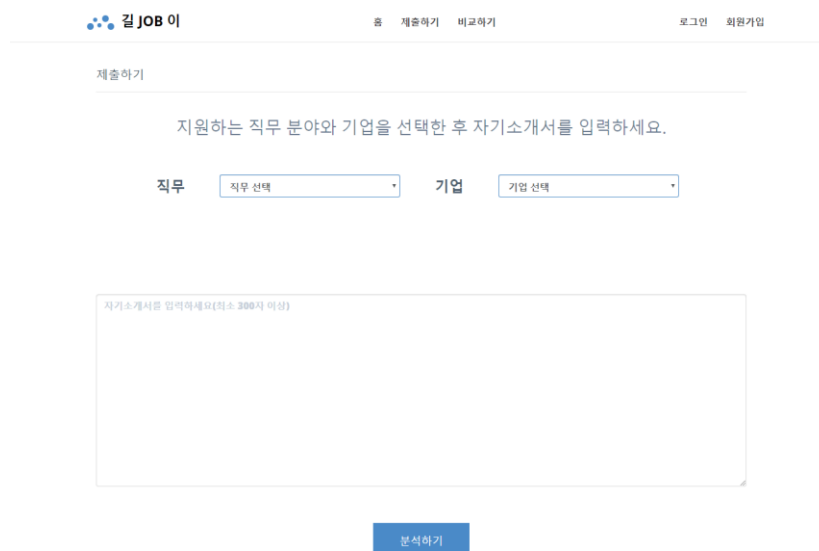
☐ Remember me      [Forgot password?](#)

**Login**

Do not have an account? [Sign Up!](#)

### [회원가입, 로그인 페이지]

회원가입 페이지에서는 이름과 이메일 패스워드를 입력하여 간단하게 회원가입이 가능하도록 구성하였고 로그인 페이지는 자신의 Username 과 Password 를 입력하여 로그인 하도록 구성하였다.



**길 JOB 이**      [홈](#)   [제출하기](#)   [비교하기](#)   [로그인](#)   [회원가입](#)

**제출하기**

지원하는 직무 분야와 기업을 선택한 후 자기소개서를 입력하세요.

**직무**         **기업**  

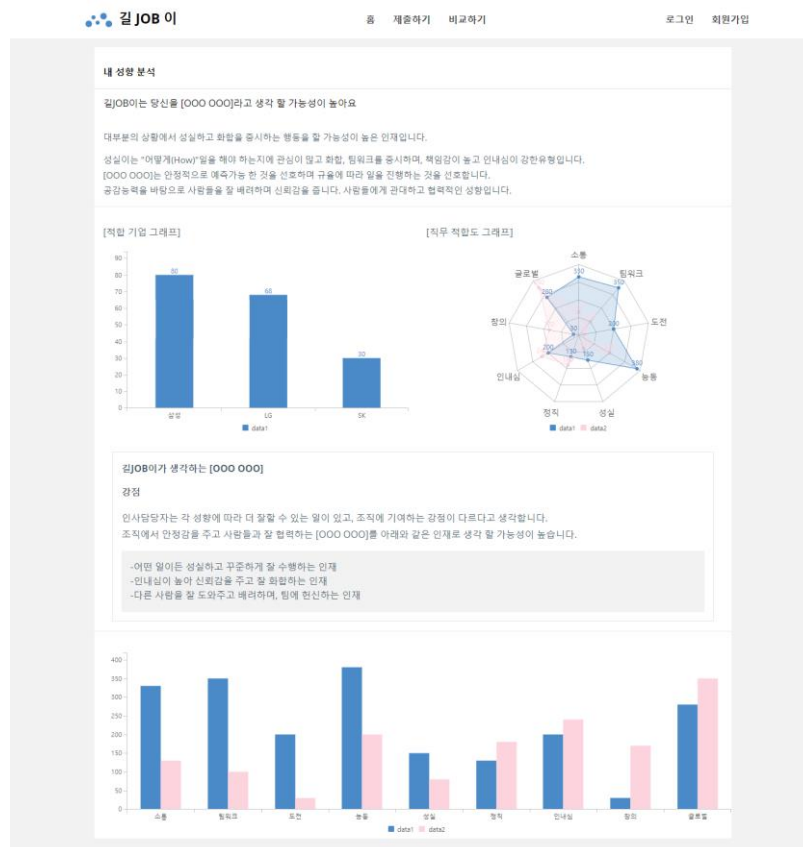
자기소개서를 입력하세요(최소 300자 이상)

**분석하기**

### [자기소개서 제출 페이지]




제출 페이지는 직무와 기업을 선택하고 자기소개서를 입력하여 제출하도록 구성하였다.  
직무, 기업 선택은 selectbox 를 이용하여 선택 정보가 서버로 전송될 수 있게 하였다.



[분석 결과 페이지]

자기소개서를 분석한 결과를 보여주는 결과페이지는

- 1) 자기소개서의 성향에 대한 설명 텍스트,
- 2) 자기소개서의 성향과 일치하는 기업을 추천해주는 적합 기업 그래프,
- 3) 10 가지 역량으로 나누어진 직무 적합도에 대한 점수를 방사형 차트로 보여주는 직무 적합도 그래프
- 4) 제출 시 선택했던 기업의 성향에 맞는 적합도를 보여주는 기업 적합도 그래프로 되어있다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

길 JOB 이

홈 제출하기 비교하기

로그인 회원가입

### 분석 결과 비교하기

이전에 제출했던 자기소개서의 분석 결과와 비교하여 확인 할 수 있습니다.

분석 직무 선택

<input type="checkbox"/>	삼성 영업 자소서 최종 수정본	2019-03-10 19:07
<input type="checkbox"/>	삼성 영업 자소서 1차 수정	2019-03-10 14:07
<input type="checkbox"/>	삼성 영업 자소서 1	2019-03-08 11:47
<input type="checkbox"/>	롯데 영업마케팅 자소서 1	2019-03-10 19:07

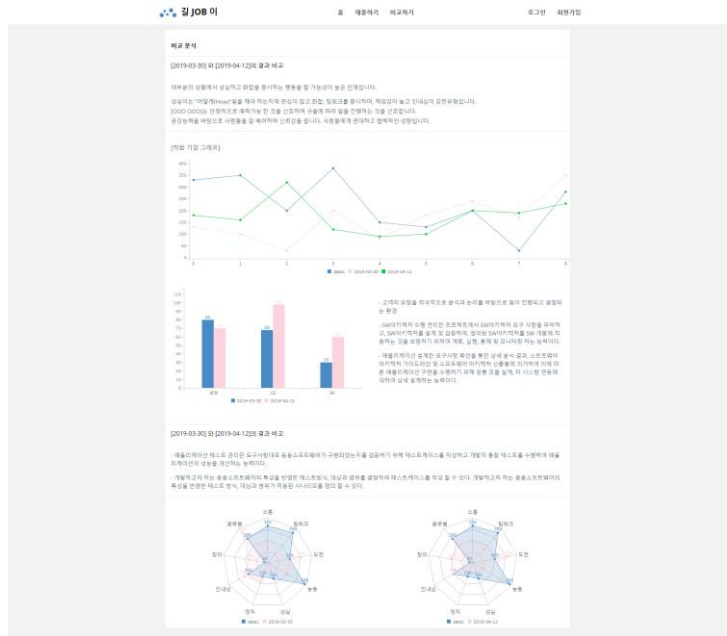
비교하기

[분석 결과 비교 선택 페이지]

비교페이지에 들어가게 되며 처음 접속 시 사용자가 분석했던 모든 결과가 조회되며 위의 select box 에서 분석 직무를 선택하면 해당 분석 직무로 제출했던 자기소개서 분석 결과만 조회하도록 작성하였다. 기록 2 개를 선택하여 비교하기 버튼을 누르면 비교 결과 페이지로 진행된다.



중간보고서		
프로젝트 명	길 JOB 이	
팀 명	4726	
Confidential Restricted	Version 2.0	2019-APR-18



### [분석 결과 비교 결과 페이지]

사용자의 두 개의 기록을 선택하여 비교한 결과를 보여주는 페이지이다. 각 분석 항목에 대하여 두 기록의 비교 결과를 보여준다.

#### 2) 로그인/회원가입

사용자 인증 시스템은 Spring boot, Spring security, OAuth 2.0 을 사용하여 구현하였다. 사용자는 구글 계정을 인증해 회원으로 등록할 수 있다.

로그인 전의 사용자는 메인 페이지, 회원가입, 로그인 기능에만 권한이 있다. 로그인 후 Access token 을 받으면 사용자는 자기소개서 분석 서비스와 분석결과 비교 기능을 이용할 수 있게 된다.

#### 3) DB

Spring boot 에서 제공하는 JPA 를 사용하여 DB 를 설계하였다. JPA 는 DB 테이블을 직접 생성하지 않고 Entity 클래스만 생성하면 어노테이션을 통해 DB 와 자동 매핑되게 된다. 또한



SQL 문을 따로 작성 할 필요 없이 Java 코드에서 처리가 가능하기 때문에 생산성 또한 증대 된다.

JPA 로 작성된 Entity 클래스는 4 개이며, 회원 정보를 구성할 속성을 가지고 있는 Member, 사용자의 정보를 저장하는 User, 자연어처리 Application 에서 생성된 기업별 직무별 특징 벡터의 수치화 된 값을 저장하는 Company, Duty 로 나뉘게 된다.

회원 정보를 구성할 속성을 가지고 있는 Member 클래스는 {id, uid, upw, uemail, regdate, updatedate, roles} attribute 를 가지고 있다.

사용자의 정보를 저장하는 User 클래스는 {id, name, email, time, state, roles, company, job\_duty} attribute 를 가지고 있다.

기업별, 직무 특징 벡터의 수치화 된 값을 저장하는 company, duty 클래스는 {company/job\_duty, communication, teamwork, challenge, active, faithful, honest, patience, creative, global, ownership} attribute 를 가지고 있다.



## 2.2.2 자연어처리

### 1) 데이터 수집

#### 1.1) 합격 자기소개서 수집

취업 정보 제공 포털사이트인 사람인, 인쿠르트, 자소설 닷컴에서 제공하는 합격 자기소개서를 수집하였다. 자기소개서를 수집하기 위한 웹 크롤러를 만들기 위해 python 라이브러리인 beautiful soup 을 사용하였다. 수집된 자기소개서는 딕셔너리 형태의 json 으로 저장한다. key 는 직무, 회사, 자기소개서 내용이다. 약 만개의 합격 자기소개서를 수집하였고 사이트 별로 자기소개서에 기입된 직무와 기업의 이름 형태가 제각각이므로 통일성 있게 정리한다.

#### 1.2) 역량 별 데이터 수집

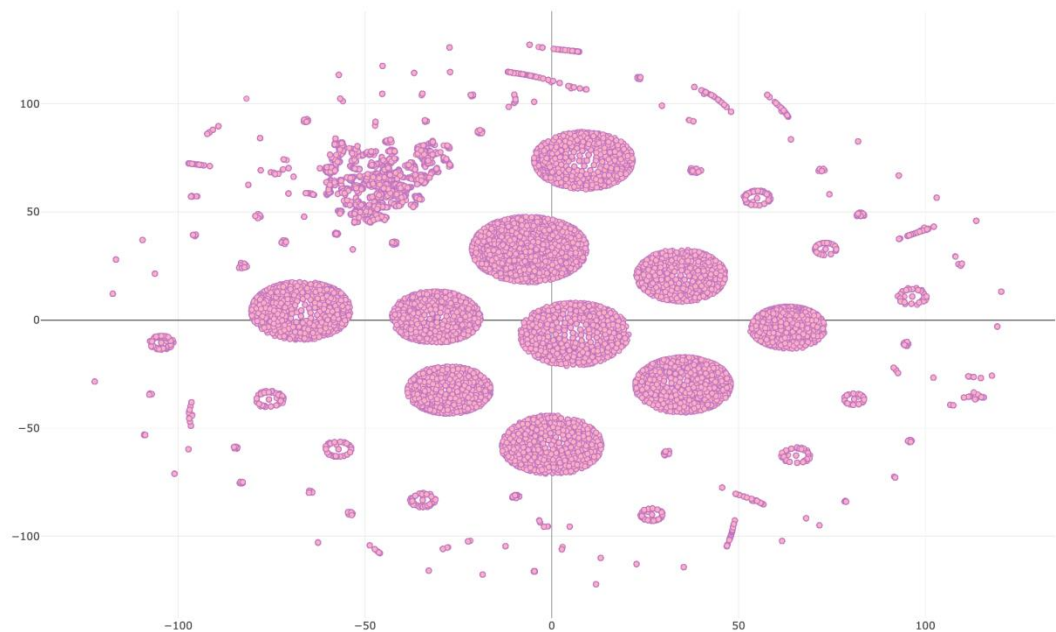
자기소개서 분석은 능동, 창의, 인내심과 같이 사용자의 핵심 역량을 나타내는 키워드 10 개를 바탕으로 분석하게 된다. 이 역량 키워드는 구인 구직 사이트에서 기업의 인사 담당자들에게 자기소개서 핵심 키워드를 조사한 것을 바탕으로 결정하였다. 따라서 사용자가 자기소개서를 입력하게되면 해당 자기소개서가 능동, 창의, 인내심과 같은 핵심 역량에 얼마나 부합하는지를 분석하여 수치로 보여준다. 이를 위해 능동, 창의, 인내심과 같은 용어가 일반적으로 어떤 문장에서, 어떤 어휘들과 사용되는지에 대한 텍스트 데이터의 확보가 필요하다. 그 결과로 입력된 자기소개서의 역량을 분석할 수 있게 된다. 따라서 핵심역량 키워드에 대한 데이터를 뉴스, 블로그를 통하여 수집하였다. 데이터를 수집하기 위한 웹 크롤러는 python 라이브러리인 beautiful soup 과 selenium 을 사용해 만들었다. 네이버와 다음에서 각 역량에 해당하는 단어를 검색하여 검색 결과로 나온 글에서 검색 단어가 포함되어 있는 문장만 추출한 뒤 텍스트 파일로 저장했다.



## 2) 데이터 전처리

전처리를 진행하는 tokenizer 함수를 만들어 인자로 분석할 데이터, 추출할 품사, 불용어 목록 등을 추가했다. 함수가 리턴하는 값은 konlpy 의 pos 함수를 이용해 토큰화, 정규화, 어근화, 불용어 제거를 거친 정제된 데이터가 된다.

### 2.1) 역량 데이터 벡터화



[역량 데이터를 벡터화 한 결과]

벡터화 과정은 scikit-learn 라이브러리를 활용하여 진행하였다. 자기소개서를 분석한 결과는 "글로벌역량", "능동", "도전", "성실", "소통", "인내심", "정직", "주인의식", "창의", "팀워크"와 같은 사용자 역량 대한 수치가 된다. 그리고 뉴스와 블로그에서 수집한 텍스트 데이터의 어휘들로 각 핵심역량별 문서 내 중요도를 나타내기 위하여 TF-IDF 가중치를 값으로 하는 벡터를 만든다. scikit-learn 의 TfidfVectorizer 함수를 분석에 사용하는데, 분석에 유의미한 어휘를 추려내기 위하여 max\_df, min\_df 를 추가하여 일정 기준보다 많이 등장한 어휘와 적게 등장한 어휘를 추렸다. 또한 텍스트의 연속된 샘플을 의미하는 n-gram 언어모델을

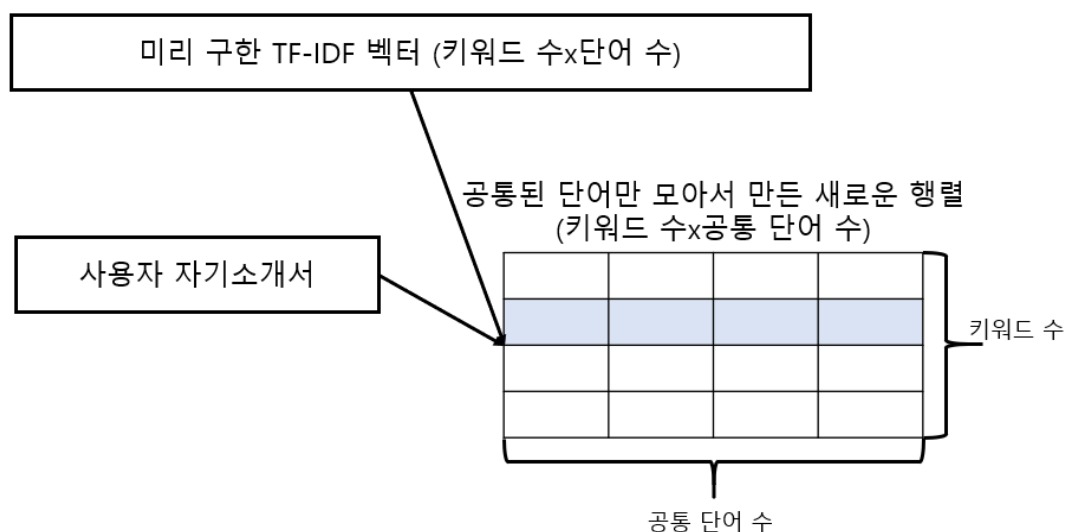




추가하여 어휘의 주변 쓰임새를 통해 보다 분석에 정확성을 높이도록 했다. n-gram 에서 n 을 너무 작게 선택하면 훈련 코퍼스(말뭉치-위의 그래프에서 각각의 군집)에서 카운트는 잘 되지만 근사의 정확도는 실제의 확률분포와 점점 멀어지므로 최대 5 를 넘지 않도록 했다. 벡터화된 결과는 행렬 형태로 저장되며 행렬의 각 행은 역량이 되고 열은 데이터에서 추출한 총 어휘가 된다.

예시로, 뉴스와 블로그등으로 부터 역량 키워드(창의, 능동, 소통..)데이터를 수집한 결과를 벡터 2 차원으로 표현하면 위의 그래프처럼 나타난다. 10 개의 역량(창의, 능동 등)이 있기 때문에 크게 10 개의 군집으로 나누어져 나타나있는 것을 볼 수 있다. 이 군집들 각각에는 예를 들어서 책임, 의무와 같이 비슷한 의미를 가진 단어들이 모여있기 때문에 꼭 자기소개서에 '책임' 이라는 단어가 들어가지 않아도 책임과 유사한 의미로 사용된 단어가 있다면 그 자기소개서는 책임의 성향을 띄고있는 자기소개서라고 판단하게 된다.

## 2.2) 사용자 자기소개서 데이터 벡터화 테스트



수집한 합격 자기소개서 중 하나를 골라 테스트 해 본 내용은 다음과 같다. 미리 만들어놓은 핵심역량 벡터와 일치하는 단어만을 뽑아서 행렬을 만든다. 이 행렬의 수치는 미리 구한

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

핵심역량 벡터와 동일하다. 다만 단어가  $n$  번 중복되면  $n$  개의 같은 열이 중복해서 생기므로 문서 내에서 자주 등장한 단어일수록 유사도가 높아진다. 행렬의 각 행을 더해서 핵심역량과 얼마나 일치하는지 유사도를 알아내고 (1x 키워드 수) 벡터에 저장한다. 이와 같은 입력 텍스트 분류 방식은 Rocchio 알고리즘과 유사하다.



## 2.2.3 결과물 목록 및 진행사항

대분류	소분류	기능	진행 현황
서버	메인 페이지	길 JOB 이 서비스의 메인 페이지	완료
	회원가입 페이지	name, email, pw 를 입력하여 회원 가입을 진행하는 페이지	완료
	로그인 페이지	가입 완료 후 인증된 계정을 통해 로그인을 진행하는 페이지	완료
	제출 페이지	클라이언트가 지원하는 직무, 기업을 선택하고 자기소개서를 작성하여 제출하는 페이지	완료
	로딩 페이지	입력한 자기소개서의 분석 진행 정도를 보여주는 페이지	진행중
	결과 페이지	클라이언트의 자기소개서가 선택한 직무, 기업에 얼마나 적합한지를 보여주는 적합도 그래프와 성향이 맞는 기업을 추천해 보여주는 페이지	진행중
	비교 페이지	자기소개서 분석한 결과 중 같은 직무, 기업을 선택한 기록 2 개를 선택하는 페이지	진행중
	비교 결과 페이지	비교 페이지에서 선택한 기록간의 비교 결과를 비교 차트를 통해 보여준다.	진행중
	웹 서버 구축	AWS 를 통해 웹 서버 구축	진행 중
	회원가입 기능	Spring boot, Spring security, OAuth 를 통해 회원 시스템을 구축	진행 중
	페이지 이동 기능	페이지간 버튼을 통해 이동하는 기능	완료
	로그인 기능	계정정보를 입력하여 로그인 하거나 인증된 구글 계정을 통해 로그인 가능	진행 중



	자기소개서 제출기능	제출 페이지에서 선택한 직무, 기업 정보와 입력한 자기소개서 텍스트 데이터를 제출하면 자연어처리 서버로 전송	진행 중
자연어처리	데이터 수집	웹크롤러를 이용한 데이터 수집	1 차 진행완료
	데이터 전처리	원문 데이터로부터 분석을 위한 형태로 전처리	1 차 진행완료
	데이터 벡터화	사이킷런 라이브러리를 이용한 데이터 벡터화	진행중
	사용자 데이터 분석	학습된 데이터와 입력된 사용자 데이터를 분석	진행예정
	프로그램 신뢰성 테스트	테스트 데이터셋을 통하여 분석 결과의 신뢰성 체크	진행예정



## 3 수정된 연구내용 및 추진 방향

### 3.1 수정사항

#### 3.1.1 서버

##### 1) Nginx (web server)

Spring boot 에서는 ServletWebServerFactoryAutoConfiguration 클래스에서 자동적으로 내장 Tomcat 설정을 자동적으로 처리한다. 따로 설정을 하지 않으면 Springboot 프로젝트를 생성 할 때 내장 Tomcat 으로 웹서버를 관리하게 된다. 하지만 추후 프로젝트의 규모가 커지고, 속도 저하를 대비해 nginx 로 웹서버를 구현하여 ec2 와 연동하여 무중단 배포를 진행 할 예정이다.

##### 2) DB

기존 계획은 NoSQL 기반의 MongoDB 를 이용하여 사용자의 정보 및 자기소개서 분석 결과를 저장하려고 했다. 하지만 auth 의 문제와 효율적인 쿼리문을 사용하지 못하는 문제로 DB 를 Spring boot 에서 제공하는 JPA 로 변경하였다. JPA 는 SQL 문을 사용 할 필요없이 Java 코드로 DB 에 접근 및 쿼리 작성이 가능 하기 때문에 생산성의 증대를 가져올 수 있다.

##### 2) AWS

최초 계획은 python 어플리케이션을 사용하기위해 Django 를 기반으로 따로 서버를 구축한 뒤 웹 소켓을 통해 Java 기반 웹 서버와 입력 자기소개서를 제출하고 분석결과를 받는 통신으로 구축하는 것이었다.

하지만 Django 기반의 파이썬 서버를 이용하려면 서버 구축 및 통신 개발 및 연구에 많은 시간이 든다. AWS lambda 서비스를 이용하면 간단하게 어플리케이션만을 활용할 수 있어 개발시간도 단축되고 Dynamo DB 를 사용하기 때문에 DB 와의 연동도 간편하여 AWS lambda 로 수정하였다.



입력된 사용자의 자기소개를 분석하여 적합도에 대한 결과 값을 반환해주는 python nlp(natural language processing) model 어플리케이션을 AWS lambda 와 API Gateway 를 활용해 서버리스로 구축한 후 API 를 작성한다.

### 3.1.2 자연어처리

#### 1) 딥러닝 클라우드 사용


데이터를 벡터화 하는 과정에서 구글 클라우드 플랫폼을 사용할 예정이었지만 국민대학교에서 클라우드 서비스를 제공한다는 것을 확인하고 더 편리하게 사용할 수 있는 국민대 딥러닝 클라우드를 사용하기로 하였다.

#### 2) 사용자 자기소개서 벡터화 과정 변경

초기 계획은 핵심역량별(능동, 도전, 창의 등), 회사별, 직무별에 대한 벡터화를 마친 후, 사용자 자기소개서를 벡터화 시킨 결과를 가지고 각 핵심역량 벡터와 코사인 유사도를 분석하여 각 역량의 해당되는 정도를 분석할 예정이었다. 하지만 사용자 자기소개서의 특징벡터를 만들때 기존에 수집한 데이터들과 함께 벡터화 하는 것이 아니기 때문에 각각의 벡터 공간이 다르기 때문에 코사인 유사도를 분석하는데 유의미한 결과를 얻을 수 없다는 문제점을 발견하였다. 따라서, 사용자 자기소개서로부터 핵심역량 벡터에서 학습된 단어와 동일한 단어의 tf-idf 값을 추출해내어 사용자 자기소개서가 각 역량의 유사도를 분석하는 방식으로 변경하였다.

#### 3) 사용자의 자기소개서와 미리 저장된 벡터들과의 비교 방식 변경

사용자의 자기소개서 분석 결과를 시각화해서 보여주기 위한 방식을 변경했다. 2 번에서 기술한 것처럼 특징벡터를 만드는 방식이 달라졌기 때문에 코사인 유사도는 사용자와 적합한 기업의 상위 3 순위를 알려줄 때만 사용한다. 사용자가 선택한 기업과 직무를 현 자기소개서와 비교 분석한 결과는 미리 학습시켜 저장한 벡터들(기업별, 직무별)과 별도로 유사도 계산을 하지 않고 사용자 자기소개서의 역량별 유사도 수치 그대로 그래프에

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

표현한다. 이 때, 학습시켜 저장한 벡터들의 표현은 보정을 거쳐 적절하게 비교를 할 수 있도록 한다.

#### 4) 특징 벡터에 새로운 내용 반영

특징 벡터를 만들 때 합격 자기소개서만을 이용하여 각 직무와 기업별 벡터를 만들려고 하였으나 더 정교한 벡터를 만들기 위하여 기업이 추구하는 인재상과 같은 정보를 추가로 반영하고 반영한 정보에 대해 가중치를 줄 예정이다.



중간보고서	
프로젝트 명	길 JOB 이
팀 명	4726
Confidential Restricted	Version 2.0 2019-APR-18

### 3.1.3 개발 일정

항목	세부내용	1 월	2 월	3 월 1 주	3 월 2 주	3 월 3 주	3 월 4 주	4 월 1 주	4 월 2 주	4 월 3 주	4 월 4 주	5 월 1 주	5 월 2 주	5 월 3 주	5 월 4 주
요구사항 분석	요구 분석														
	아이디어 구상														
관련분야 연구	자연어처리 및 텍스트마이닝 연구														
	관련 시스템 분석														
설계	시스템 설계														
구현	데이터 크롤링														
	웹 UI 구현														
	DB 구축 및 설계														
	회원가입 /로그인														
	결과 페이지 API 구축														
	비교분석 페이지 API 구축														
	데이터 전처리														
	데이터 벡터화														
	사용자 데이터 분석														
	프로토 타입 구현														
	자연어처리 어플리케이션과 웹 서비스 간 연동														
테스트	자연어처리 어플리케이션 신뢰성 테스트														
	자연어처리 어플리케이션과 서버간 통신 테스트														
	서버 유지보수														
최종발표	발표준비 및 발표														





## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### 4.1.1 Server

##### 1) Web Server

제출 페이지에서 입력한 자기소개서를 AWS DynamoDB 에 전송하고 자연어처리 어플리케이션에서 분석한 결과 데이터를 받아 결과 페이지에 보여주고, 분석 결과를 데이터베이스에 저장하는 기능을 구현할 예정이다.

spring boot 로 구현한 controller 와 bootstrap 으로 구현된 view 를 연결하기 위해서 ajax 를 통해 통신한다. ajax 는 비동기식으로 XML 을 이용하여 서버와 통신하는 방식으로 서버에서 Data 만 전송하면 되므로 전체적인 코딩의 line 이 줄어 들게 된다.

##### 2) 자연어 처리 Application Server

자연어처리 Application 의 서버를 Aws 에서 제공 하는 Lambda 를 사용하여 구현 예정이다. Lambda 는 서버를 프로비저닝 하거나 관리 할 필요 없이 코드 실행이 가능하게 해주는 서비스이다.

AWS lambda 에 함수를 작성하고 Dynamo DB 에 연동하여 자기소개서 분석 결과가 데이터베이스에 저장되고 조회할 때에도 lambda 를 호출하도록 한다. lambda 의 호출은 API Gateway 를 통해 설정한다.

##### 3) 데이터베이스

기존에 저장된 기업, 직무별 데이터는 Spring boot 에서 제공하는 JPA 를 사용하여 데이터를 제공한다. 자연어처리 Application 에서 제공하는 데이터들 또한 user 의 정보에 저장해야 된다. 실시간 데이터를 JPA 에 직접 저장하는 것은 비효율적인 방법이기 때문에 AWS 에서 제공하는 Dynamo DB 를 이용하여 데이터를 저장한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

#### 4) view 와 controller 연동

웹 페이지와 controller 의 통신은 ajax 를 통해 구현할 예정이다.

### 4.1.2 자연어처리

#### 1) 합격 자기소개서 추가 수집


현재 보유하고 있는 합격 자기소개서는 약 1 만개이다. 합격 자기소개서 자체에 대한 신뢰성을 높이기 위해 국민대학교 경력개발 지원단 홈페이지(에듀스) 에서 제공하고 있는 합격 자기소개서 약 1 만 5 천개를 추가 수집하여 벡터를 만드는 과정에 반영할 예정이다. 합격 자기소개서를 추가 수집하게 되면 분석해주는 직무와 기업의 종류를 늘릴 수 있다

#### 2) 불용어 사전을 이용한 데이터 전처리

학습 데이터 총량이 방대하므로 데이터를 분석하는 용도에 맞게 정제하는 과정에서 시간이 매우 오래걸리는 것을 확인할 수 있었다. 따라서 문장들로 이루어진 데이터에서 큰 의미를 가지고 있지 않은 단어들을 최대한 제거하는 것이 실행 과정에 있어서 효율적일 것 이다. 현재, '하다', '되다' 두 개의 불용어만 제거하여 전처리 과정을 진행했는데 한글에는 다양한 불용어들이 있으므로 다양한 불용어 정보를 수집하여 불용어 리스트를 만든 뒤 제거하는 방식으로 전처리를 수행할 예정이다.


#### 3) 사용자 자기소개서와 비교할 수치 보정

직무별, 기업별 합격 자기소개서의 특징 벡터와 사용자로부터 자기소개서를 입력 받아 만드는 특징 벡터는 TF-IDF 값의 합으로 만들게 된다. 이 수치는 단어의 빈도수에 따라 값이 커지기 때문에 많은 양의 데이터로 만드는 합격 자기소개서의 특징 벡터의 값이 압도적으로 높을 수 밖에 없다. 그렇기 때문에 사용자의 자기소개서와 유의미한 비교를 위해서 두 벡터의 값을 합리적인 수치로 보정해야 한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

#### 4) 텍스트 분류 단계의 다양한 방법 적용으로 효율성 비교 및 신뢰성 테스트

지금 현재 텍스트 분류를 위한 작업은 Rocchio 알고리즘과 유사한 방식으로 진행된 결과이다. 이는 프로토타입 결과를 내는 가장 효율적인 방법을 채택한 것으로, 최종 결과물을 내는 데에는 조금 더 다양한 알고리즘 방식을 살펴볼 필요가 있다고 결론을 내렸다. 다양한 알고리즘을 살펴보고 분석된 결과의 정확성을 비교하여 알고리즘을 개선한다면 신뢰성 향상에 도움이 될 것이다. Rocchio 알고리즘 외에도 SVM, Naive bayes, KNN 이 존재한다. 어떤 알고리즘이 본 프로젝트에 가장 적합하고 정확성이 높을지에 대해 비교 분석할 예정이다. 또한 시스템 개발이 완료되면 테스트 데이터 셋(합격 자기소개서 등)을 이용하여 구현한 알고리즘이 얼마나 정확히 분석하는지 테스트한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	길 JOB 이	
	<b>팀 명</b>	4726	
	Confidential Restricted	Version 2.0	2019-APR-18

## 5 고충 및 건의사항

1. 고정적으로 회의하거나 개발할 공간이 부족하다.