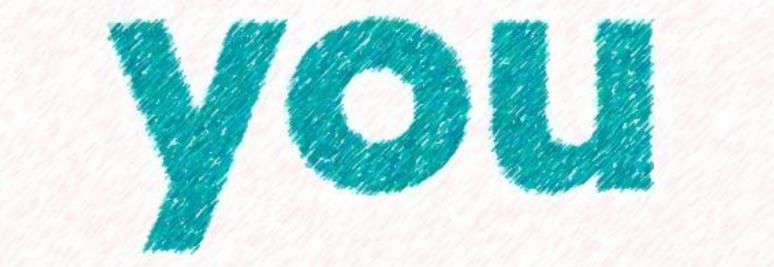


#WITH YOU PROJECT

#그대 곁에

당신은 더 이상 혼자가 아닙니다. 위드유와 함께 소중한 당신을 도와드리겠습니다





목차

#with youu

The most difficult thing in the world is to win human hearts. Various different facets of each face.

01

프로필 & 개발 배경

The most difficult thing in the world is to win human hearts

02

기능 설명

텍스트 검출 방식/ 나이브 베이즈/ 형태소 분석기

03

1차 사용성 평가

디자인 평가 및 사용방식 평가.

04

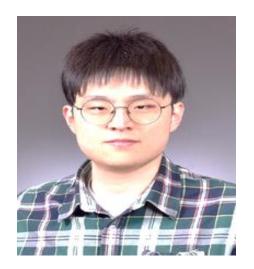
추후 개발 일정

조원 소개

The project Member's profile

조원 소개

The project Member's profile



박민규

Q 010-4150-5696

mingyu0212@gmail.com

 \mathcal{O}



김상현

010-6850-4050

hanseo39@naver.com

⑤ 어플 개발



공민석

Q 010-8020-1064

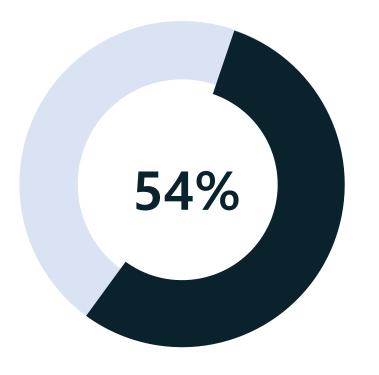
martinJohnsonKong@naver.com

UX/UI 디자인 및 사용성 평가AI를 이용한 텍스트 검출 기능 구현

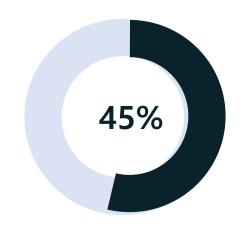
개발 배경 Development background

성범죄 상황 시 왜 적극적으로 대응하지 못했나요?(복수응답)

출처: 한국 성폭력 상담소 부설연구소 '울림'

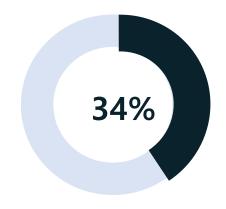


당황하여 아무런 생각이 나지않아서

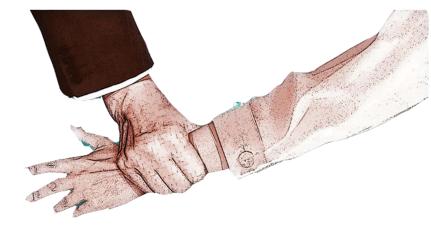


당시에는 성폭력인지 몰라서



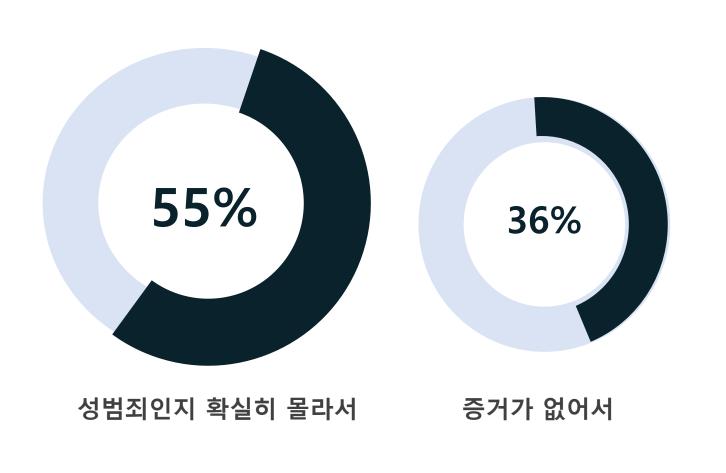


공포심에 몸이 굳어서



왜 주변에게 적극적으로 도움을 요청하지 못했나요?(복수응답)

출처: 한국 성폭력 상담소 부설연구소 '울림'



웹페이지: 2017-09-20 여성신문

http://www.womennews.co.kr/news



개발 목표

: 긴급한 상황, 핸드폰을 꺼내기 어려운 상황에서 손쉽게 증거를 수집하고, 증거력 있는 단어를 검출



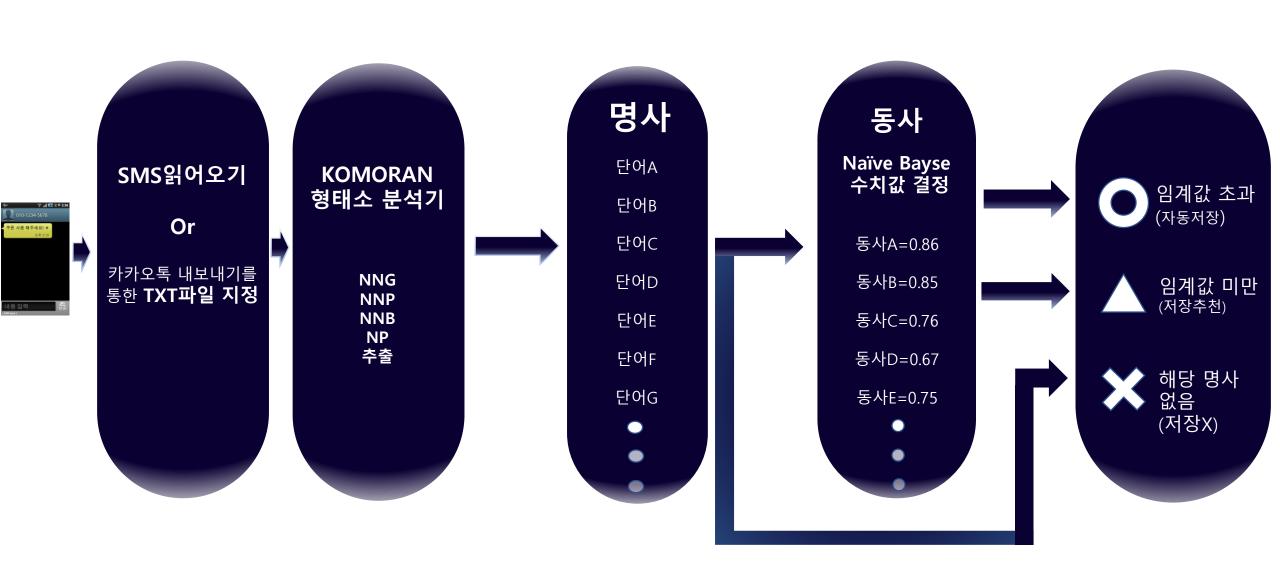
기술 상세 설명

Detailed Description of Technology



텍스트 검출 과정

Process of Text Classification or detection.

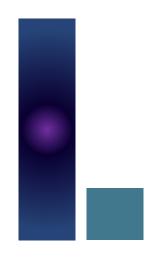




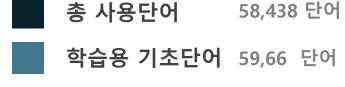
텍스트 검출 과정

Process of Text Classification or detection.

출처 : 국립 국어원.



- 용량이 너무 무거움
- 단어의 확률 값의 객관성 저하(naïve bayse 사용시)



출처: 국립국어원



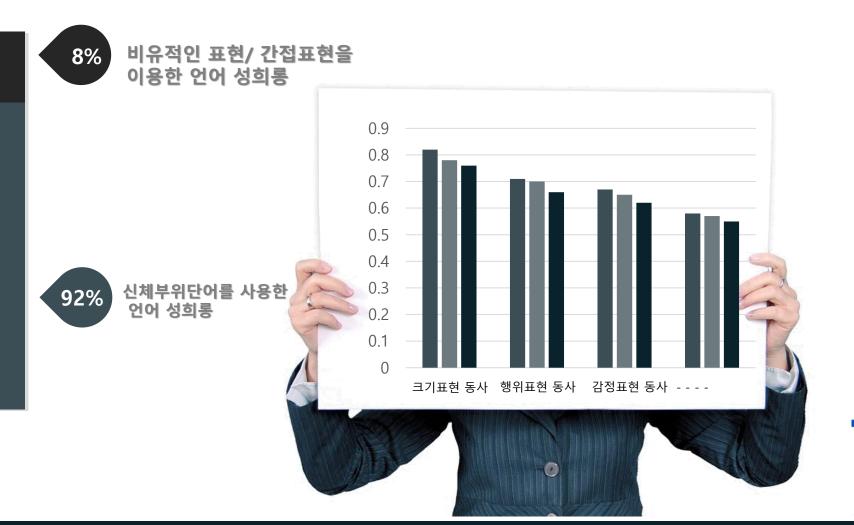




조사+관형형

텍스트 검출(단어 선별 과정)

Process of Text Classification or detection.



명사 선별 (121개)

언어 성희롱의 경우 유죄로 인정된 판례의 90% 이상은 신체부위를 직접적으로 언급하여 수치심을 준 경우,

신체부위 이외에 성적 수치심을 줄수 있었던 (성역할을 부여하는 단어 등) 직접적 단어들 추가

사용빈도 순 동사 800개 +naïve-bayse: 확률 값 지정

국립국어원의 사용 빈도순으로 정리된 단어 중 동사 550개를 선별하여 명사 와 매칭해보고 성희롱 적인 표현이 될 수 있는 명사-동사 조합을 매칭시켜 확률 값을 정합니다



텍스트 검출(명사+동사)

Node.js와 JSON을 이용해서 웹 크롤링 및 파싱을 통해 성범죄 판례를 수집

```
://www.google.com/webhg?hl=ko&sa=Y&ved=OahUKEwif6NTYx)bbAh)EOJOKHW7FC6UQPAgD
         g=19134481 NETARDA/DEBNA 11814ECRAZA/DEBNA 11800source=1msktbn=1schliss=1&ned=16h1MEviff0TTV/bb4VEDJOHNFF030, AUTCigB
ts=19134481 NEDARDA/DEBNA 11814ECRAZA/DEBNA 11800source=1msktbn=nesiss=1&ned=16h1Mevif6NTV/bb4VEDJOHNFF030, AUTCjgC
ts=19134181 NEDARDA/DEBNA 11814ECRAZA/DEBNA 11800source=1msktbn=vidiss=1&ned=16h1Mevif6NTV/bb4VEDJOHNFF030, AUTCjgC
        /maps.google.com/maps?la=MECN84MB1NEDN9DN4CNEBN41NB1+MECN82N4CNEBN41N80Naws_rd=ssl&um=1&ie=UTF-8&sa=X&ved=OarUnEwif6NTYxYbb4nVEO.UXHWFCGUQ.AUIDSdE
          =XECN84XB1 NEDN9DN4CNEBX41 NB1+NECN82N4CNEBX41 N80&source=1 mms&tbm=bks&sa=X&ved=CahUKEvi f6NTYxYbb4hXECUUKHN7FC6UQ_AUIDyoP
         u=XECX84XB1XEDX9DXACXEBXA1XB1+XECX82XACXEBXA1X80&source=1rms&tbm=fin&sa=X&ved=OshUKEvif6NTY×YbbAhXEOUXHW7FC6UQ AU1ECaB
      rences?h|=ko&prev=https://www.google.com/search?qX30K25ECK2584X25B1K25EDK259DX254CK25EBX2541K25B1X28ECK25B2X254CX25EBX2541X250X26gws_rdX3Dss|
     anced_search?g=%ECX84%B1%ED%9D%AC%EB%A1%B1+%EC%82%AC%EB%A1%80&h1=ko_
    i Nese
Nese Bai = Chosswij 1991 Yvito Antial Offici i Dos VRBAGoute Aloi ig=A004.20 Lv20e botogu i fytzgriffentrukto=kved=Chotlevi f90 TYvito Antiel Offici Octobri i Wadur i =
Krse=Bai = Chosswij 1991 Yvito Antial Offici i Dos VRBAGoute Aloi ig=A004.20 Lv20e botogu i fytzgrifentrukto-kvedur i = Ace
Krse=Bai = Chosswij 1991 Yvito Antial Offici i Dos VRBAGoute Aloi ig=A004.20 Kkp Anykontol Wir Plack CAUTIN Agotos I stope = Office d=Chotlevi i f60 TV xito Anticol Offici I Dos VRBAGoute Aloi ig=A004.20 Kkp Anykontol Wir Plack CAUTIN Agotos I = Ace
Krse=Ikai = Chosswij 1991 Yvito Anticol Offici I Dos VRBAGoute Aloi ig=A004.20 Kkp Anykontol Wir Plack CAUTIN Agotos I = Ace
x² ∧ 10 €
```

```
import urllib.request as request
import json

for item in output:
    print(item["name"])
    print(item["full_name"])
    print(item["owner"]["login"])
    print()
```

```
## Soup = BeautifulSoup(xml, "html.parser")

## Soup = BeautifulSoup(xml, "html.parser")

## Soup = Beautiful_Soup(xml, "html.parser")
```

KOMORAN 2.4 형태소 분석기

1차필터링을 위해 명사를 분류해주는 응용 프로그램 입니다. 리눅스, 자바뿐 아니라 파이썬, C++에도 다른 버전의 형태소 분석기가 존재 합니다,

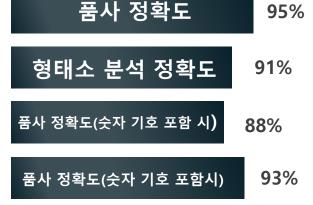


형태소 분석기 (KOMORAN 2.4)

추출된 명사단어와 검출할 문장에서의 명사 관계를 보기 위해 검출할 문장의 형태소 분석에 필요한 JAVA기반의 형태소 분석기 입니다.

형태소 분석기 정확도

(모의 테스트에서는 Mecab-Ko 0.9.2 이용 (가상머신을 이용한 Linux) 어플에서는 자바기반 언어인 KOMORAN으로 적용



- String in = "바람과 함께 사라지다를 봤어.";
 int nbest = 2;
 List<List<Pair<String,String>>>> analyzeNbestResult = komoran.analyze(in,nbest);
- 2150 2150 2150 1 till String, String . . . till y 250 to 550 to 5
- 분석 결과
- best 1
- 바람과 함께 사라지다/NNP+를/JKO
 보/VV+았/EP+어/EF+./SF
- best 2
- 바람과 함께 사라지다/NNP+를/JKO 보/VX+았/EP+어/EF+./SF

※Mecab-ko형태소 분석기(모의테스트)



형태소 분석기 (Mecab-ko)

추출된 명사단어와 검출할 문장에서의 명사 관계를 보기 위해 검출할 문장의 형태소 분석에 필요한 Linux기반의 형태소 분석기 입니다.



```
"token" : "먹/VV",
      "start offset" : 61,
      "end offset" : 62,
      "type" : "VV",
      "position" : 20
      <u>"tok</u>en" : "법",
      "start offset": 63,
      "end offset" : 64,
      "type" : "NNB",
      "position" : 21
      "token" : "좀",
      "start_offset" : 65,
      "end_offset" : 66,
      "type" : "MAG",
      "position" : 22
      "token" : "알려",
      "start offset" : 67,
      "end_offset" : 69,
      "type" : "INFLECT",
      "position" : 23
      "token" : "알리/VV",
      "start_offset" : 67,
      "end_offset" : 69,
      "type" : "VV",
      "position" : 23
      "token" : "줘라",
      "start_offset" : 69,
      "end_offset" : 71,
      "type" : "INFLECT",
      "position" : 24
delete index
  "acknowledged" : true
lks21c@lks21cubuntu:elasticsearch-5.2.0 ()$
```

Naïve Bayse Function

인공지능 분야에서 사용하는 기초적인 함수이며 베이즈의 정리를 이용해 간단하게 연산한다고 해서 Naïve(순진한) Bayse라 이름이 붙여짐.

간단한 함수구조 🔀

간단하면서도 특징벡터를 모두 함수 에 넣을 수 있는 강점이 있다..

// Create a new bayes classifier with string categories and string features. Classifier<String, String> bayes = new BayesClassifier<String, String>(); // Two examples to learn from. String[] positiveText = "I love sunny days".split("\\s"); String[] negativeText = "I hate rain".split("\\s"); // Learn by classifying examples. // New categories can be added on the flv, when they are first used. // A classification consists of a category and a list of features // that resulted in the classification in that category. bayes.learn("positive", Arrays.asList(positiveText)); bayes.learn("negative", Arrays.asList(negativeText)); // Here are two unknown sentences to classify. String[] unknownText1 = "today is a sunny day".split("\\s"); String[] unknownText2 = "there will be rain".split("\\s"); System.out.println(// will output "positive" bayes.classify(Arrays.asList(unknownText1)).getCategory()); System.out.println(// will output "negative" bayes.classify(Arrays.asList(unknownText2)).getCategory()); // Get more detailed classification result. ((BayesClassifier<String, String>) bayes).classifyDetailed(Arrays.asList(unknownText1)); // Change the memory capacity. New learned classifications (using // the learn method) are stored in a queue with the size given // here and used to classify unknown sentences. bayes.setMemoryCapacity(500);

단순 필터링 보다 높은 정확도



단순 필터링은 해당되지않는 문장도 검출하는 경우가 많음.

ex) 맥주가 차갑다 피자가 차갑다 과자가 너무 달다

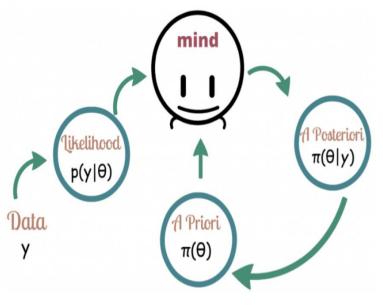
P(피자|P)*P(차갑다|P) p(피자|N)*P(차갑다|N)..... 비교 후 결정

(모르는 값이 있을 경우 확률 값이 '0' 처리 되므로,이를 방지하기 위해 Log 계산)

텍스트 검출 방식에 이용가능



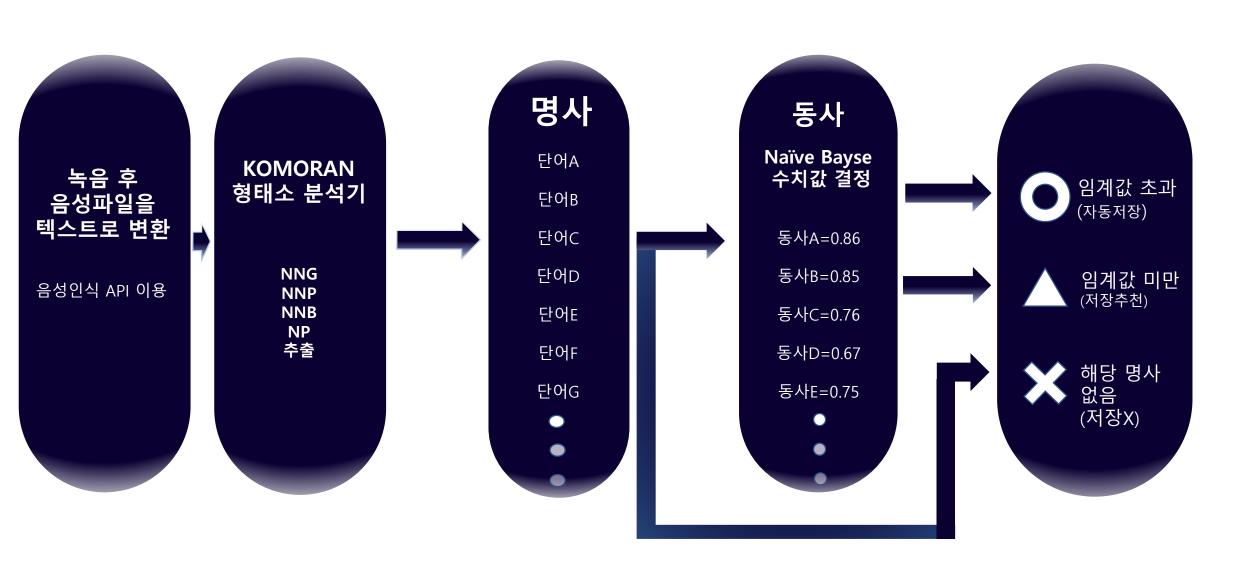
스팸메일을 분류하는 인공지능 함수 를 만드는 방식에도 응용가능





음성 검출 과정

Process of Voice Classification or detection.



1차 사용성 테스트

DIY 사용성 테스트로서, 정성적인 방법으로 명백한 문제점을 잡기위해 테스트를 진행하였습니다.

테스트 참가인원: 9명 (여성7명,남성2명, 한성대 재학생)

테스트 장소: 스크린 공유 가능한 강의실

테스트 방법: adobe XD로 어플 사용 후 <u>심각한 문제 위주로 리스트 작성</u>

테스트 시간: 1시간 내외 (5.3 12:00~1:30)



로고와 화면의 은폐성 강화 녹음 방식 수정 81% 정보의 손쉬운 공유기능 추가 50%









개발일정 및 수정사항 Schedule board & Progress

추후 개발 일정(5.15~5.31)

자바에 KOMORAN기능과 기타기능을 수정하면 대략적 프로젝트 완료, 사용성을 증대 시키고 데이터의 객관성을 높이는데 주력 할 예정입니다.



뉴스나 판례를 분석하여 비유적인 표현을 이용한 성희롱의 판례를 추가하고 비유적인 표현이 가능한 동사의 확률을 수정/보완

명사선별에 대해 객관성 증대

현재 신체부위 위주로 선별되어 있는 단어에 대해서 실제 사용되는 명사에 대해서도 단어를 다 선별/수정 할 예정.

2차 사용성 평가 및 어플 디자인 수정

1차 사용성 평가는 디자인 평가에 치중한 반면 2차 평가 때는 객관성 점검위주로 계획 테스트 참가자 수를 늘리기 위해 원격 사용성 평가 방식도 도입 정량적인 방식으로 데이터 검증

기타 기능상세

specific function



성범죄성 단어 검출 시 대상자 폴더를 자동 생성하고 텍스트 검출 시 자동저장 하여 추후 정보관리에 용이함



낯선 곳 에서 와아파이가 잡힌 경우 자동재생, 및 설정가능



긴급하거나 돌발상황시, 간단한 터치로 실행하여 증거를 수집 할 수 있음 (현재 어플은 흔들어서 키는 정도)



이미지를 대상자에게 지정 및 관리

어플리케이션 UI

Application User Interface

일정표 및 진행상황

Schedule board & Progress

	1							2	2						3	3													
	3월	3월						4월							5월	5월													
	일	목	화 10	일 15	금 20	수 25	윌 30	일	목 5	화 10	일 15	금 20	수 25	월 30	일	목	화 10	일 15	금 20	수 25	월 30	일 1	목	화 10	일 15	금 20	수 25	월 30	
			10	15	20	23	30		,	10	13	20	23	50			10	15	20	23	50							30	상태:
공민석		거플 구	조 설계	계																									완료
02.					기 프로!	토타입	디자인																						완료
										0	l EJ Till (nI A I	LQM	ᇛᄼ	DI 1111														 진행중
		인터페이스 사용성 평가 및 피드백 이를 기계															진행 중												
		서울시 앱 공모전 공공부문:"여성"															시작 안 함												
기사성				N	: 6:	4: C	2 010	ᇵᆒ		시지! 7	71 년										^	^{스시}	H 0 J	E.G. 6	OOTI	E. M	O		완료
김상현		Notification을 이용한 텍스트 수집 기능																											
		자동 프로필 작성기능 구현 기능 기계															진행중												
		기타 기능 구현 및 기능 오류 수정															진행중												
		간편/자동 녹음 기능 구현 기능 구현 기능 구현 기능															진행중												
박민규	머신러닝 API를 이용한 텍스트 검출기능 구현 기가 기계															진행중													
		검출 기능 어플리케이션에 구현															시작안함												
ALL																						2	최종 별	生					진행중
																							보	고서	작성				시작안함
											J	나용 E	당성	검증															
	인터뷰 및 사용성 테스트/설문조사																												

