ISAC

INTEGRATED SYSTEM FOR ANSWERING CUSTOMER JUSTICE USING NLP

프로젝트 배경
PROJECT OUTLINE

소비자 민원 서비스

소비자 민원의 증가

시장 경제는 사람들의 소비에 의해 돌아간다. 하지만 사람들은 소비를 하는 과정 속에서 피해가 발생하게 되었고, 이 피해액은 연간 5조원 이상으로 추정된다. 이에 따라 대한민국 정부의 공정거래 위원회 에서는 그러한 소비자들을 보호하기 위해 소비자 보호원을 설립하고, 2010년 1372 소비자 상담센터를 개설하여 운영하고 있다.

그럼에도 불구하고 연간 약 700만건의 소비자 피해 사례가 발생하고 있고, 이 중에서 약 11%만이 민원을 신청하지만 이러한 민원만 해도 연평균 80만건이 넘고, 현재까지 인터넷으로만 집계된 민원이 200만건이 넘어서고 있는 상황에서 소비자 상담에 대한 필요성과 중요성이 꾸준히 증가되고 있다. 또한 늘어나는 소비자 민원으로 인해서 소비자들에겐 늦은 답변시간, 근로자에게는 과도한 업무와 같이 모두에게 불편함이 증대되어 가고 있는 현실이다.

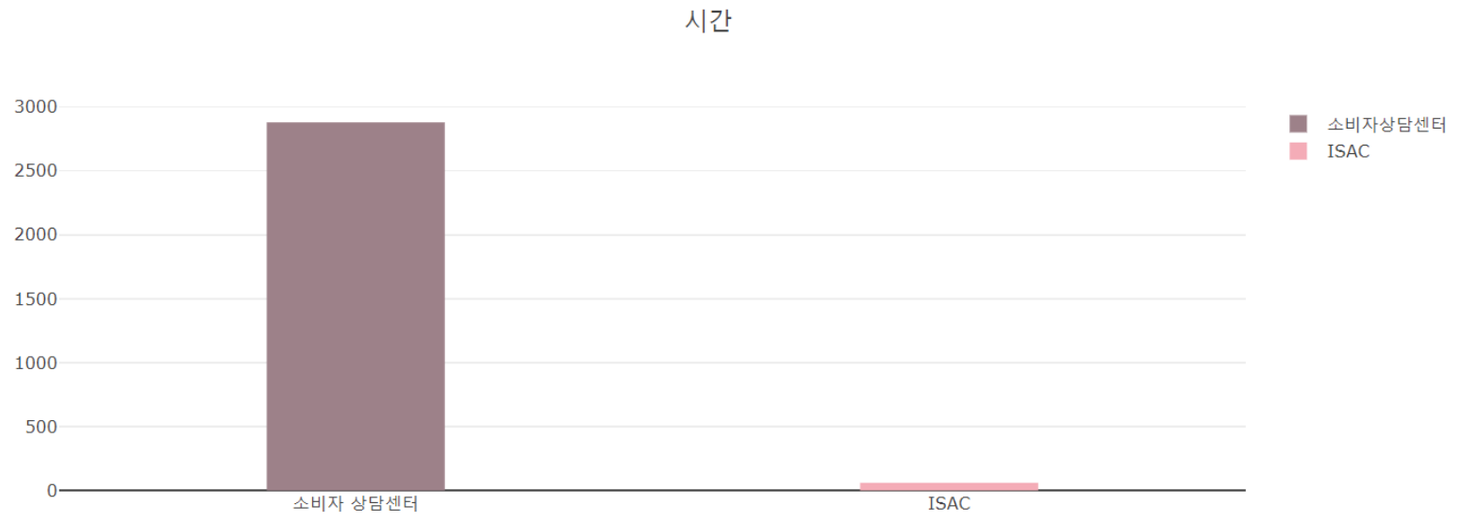
우리는 이러한 문제점들을 바탕으로 정보를 한번에 빠르게 답변해줄 수 있는 통합적인 Q&A 시스템을 구축하려고 한다. 이러한 시스템을 만듦으로써 얻는 기대효과는 다음과 같다.

프로젝트의 기대효과

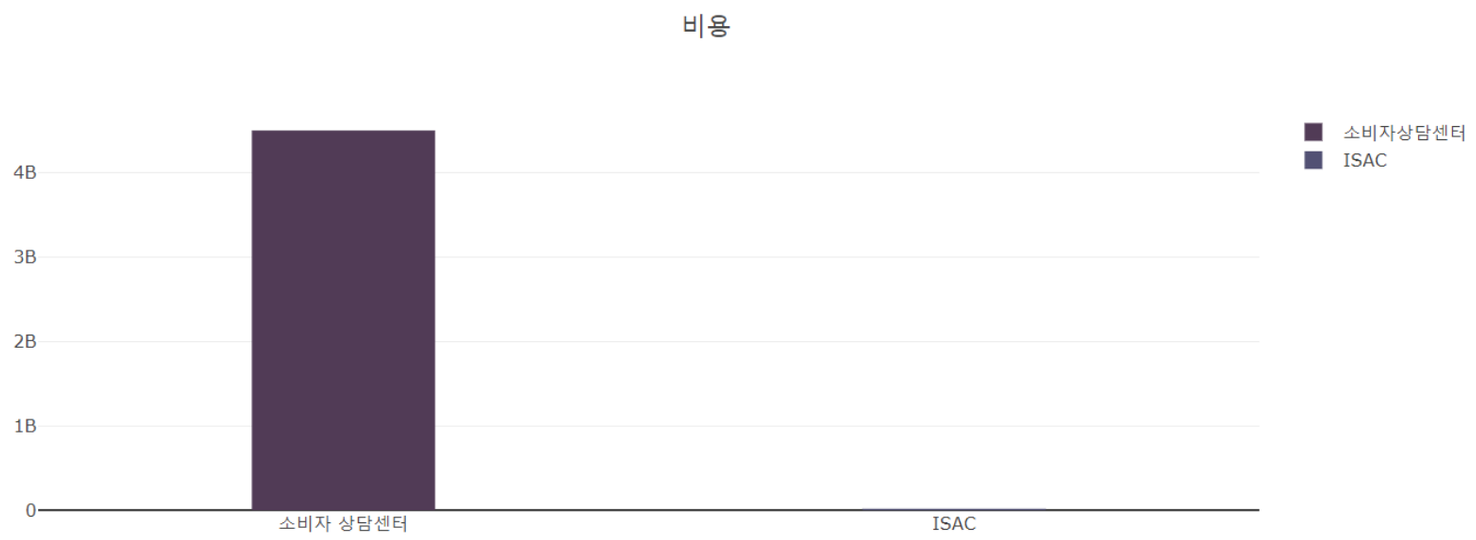
현재 공정거래 위원회에서 운영중인 1372 소비자 상담센터에서는 여러가지 문제점이 존재한다. 가장 먼저 느린 답변시간이다. 소비자 상담 센터의 경우 상담원이 직접 상담글을 읽고 답변을 해주는 형식으로 운영되고 있다. 이러한 운영방식의 문제점은 답변이 달리는데 걸리는 시간이 길다는 점이다.

하지만 우리가 제공할 ISAC Q&A 시스템은 약 1분 이내에 질문을 분석하고 상황을 파악하며, 사용자에게 적절한 답변을 제공하게 된다.

이러한 시간단축의 효과로 사용자들은 즉각적인 대처가 가능하고, ISAC에서 제공한 타 기관 연계 통합적 솔루션을 통해 피해 구제 절차를 간소화 시키는 효과를 누릴 수 있다. 또한 소비자 상담센터의 운영비용 절감의 효과를 가져올 수 있으며, 공정한 시장경제를 구성하고, 소비자의 권익을 제고하는 기대효과를 가진다.



소비자 상담센터와 ISAC의 상담에 걸리는 시간 비교 (min)



소비자 상담센터와 ISAC의 상담에 걸리는 비용 비교 (bilion)

데이터 수집 및 전처리

DATA CRAWLING & PRE-PROCESSING

데이터 수집

- 1372 소비자 상담센터에서 다수의 소비자가 공유할 필요성이 있어 질문자의 동의를 얻어 공개한 상담 답변 약 14-15만건
- 모범상담 약 1만건
- 소비자 상담의 판단 기준이 되는 해결기준 크롤링

데이터 전처리

- 맞춤법 검사
- 질문에 대한 형태소 분석
- 질문 상황을 인식하여 태깅 작업
- 답변에 대한 분류 추출
- 답변에 대한 중요부분(알맹이) 추출
- word2vec / doc2vec 임베딩 적용을 위한 데이터 전처리 작업
- 해결기준 형태소 분석

ISAC model

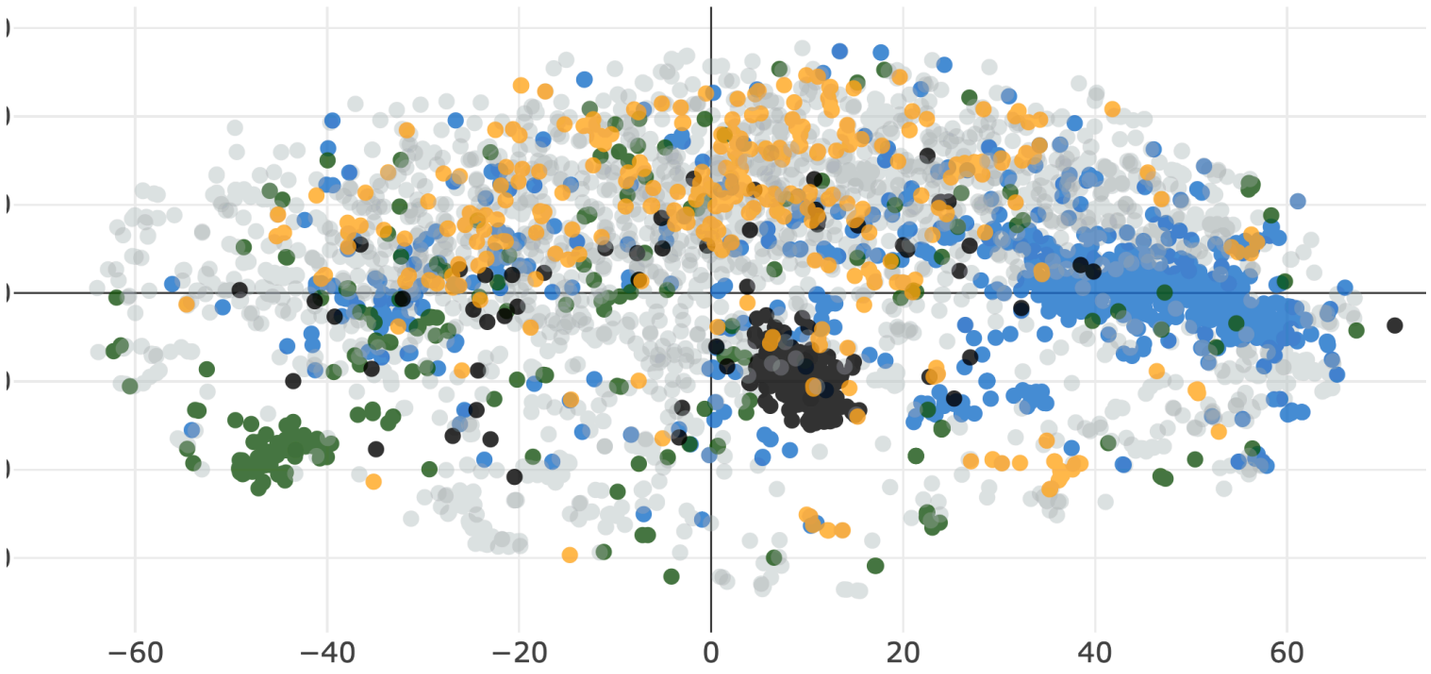
WORD2HYBRID MODEL

word2hybrid model

word2hybrid 라고 명명한 이 모델은 입력한 제목을 기반으로 유사한 답변을 뽑아주는 모델이다.

보통 상담의 중요한 단어는 제목에 넣는 경향이 크다. 그래야 소비자가 처한 상황을 잘 전달 할 수 있기 때문이다. 우리는 이러한 점에 주목하여 입력받은 제목을 가지고 기존에 있었던 유사한 질문과 답변을 가져오도록 만들어 보았다.

이때 가장 먼저 사용한 방법은 tf-idf 알고리즘을 사용한 유사도 측정이다. 이는 문장간의 유사도를 직접 구해주기 때문에 쉽게 적용할 수 있고, 입력한 input의 단어들을 기존에 존재하는 제목속의 단어들과 비교했을 때 자주 등장한다는 의미를 가진다. 따라서 해당 문장과 유사하다고 볼 수 있다. 이를 이용하여 첫번째 유사도를 구한다. (1)



word embedding

두번째 방법은 word embedding을 적용하였다. 기존에 가지고 있던 제목들에서 뽑은 단어들과 본문 태깅을 통해 본문의 내용까지 담고있는 태깅된 단어들 총 29만개의 데이터 (200만개의 단어)를 가지고 word embedding을 실시하였고, 이를 통해 약 7500개의 단어를 300차원의 벡터로 표현하였다.

위의 그림은 300차원의 벡터를 2차원공간에 그린 그래프이다.

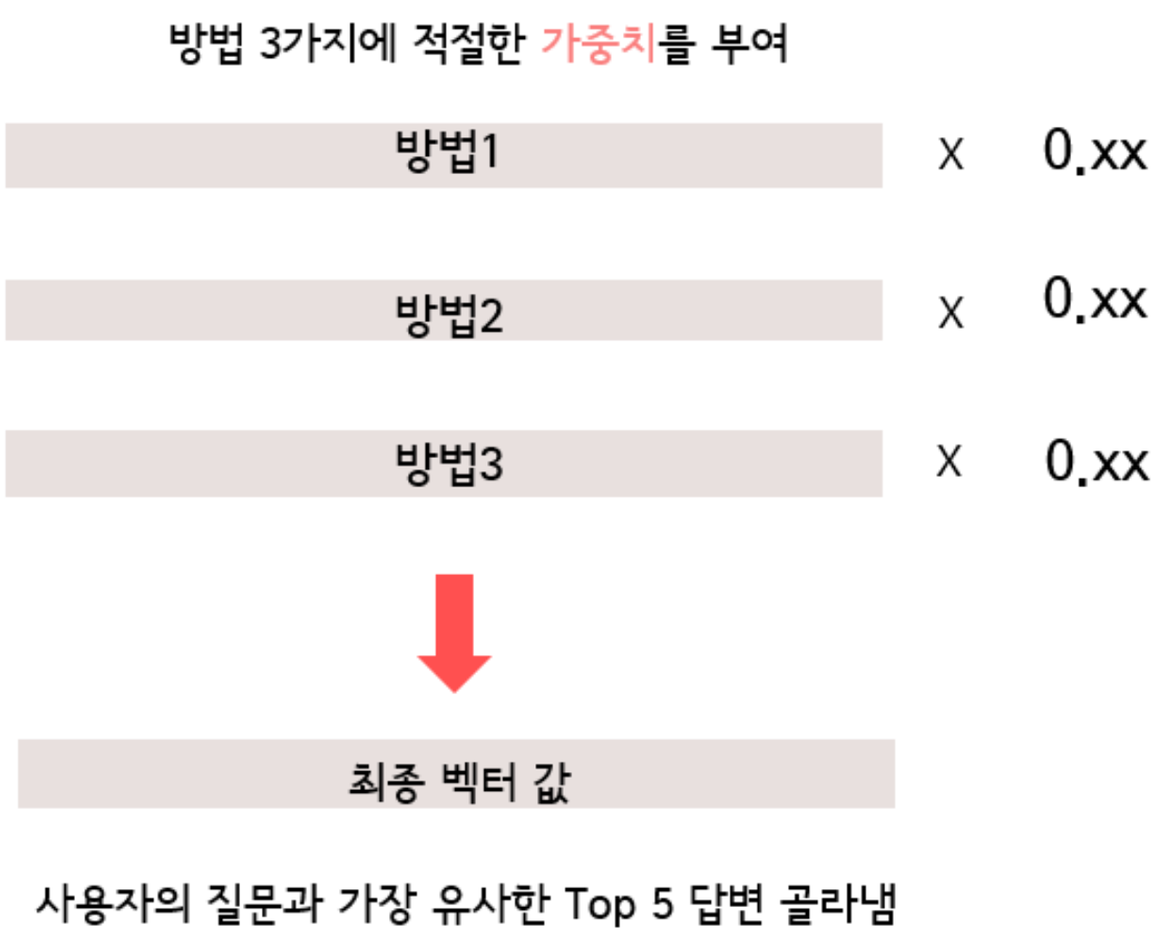
두번째 방법에서는 입력된 제목에서 뽑은 단어들과 비슷한 벡터값을 가진 단어들을 뽑아 해당되는 단어들을 같이 고려하여 유사한 질문들을 찾고 유사도를 구한다.(2)

word2hybrid model

세번째 방법은 document embedding이다. 이는 문장 자체를 하나의 벡터로 임베딩하여 비교하는 특성을 가진다. 이를 수행하기 위해 전처리 작업이 필요해 질문자들이 입력한 질문의 제목들과 태깅된 단어들을 이어붙여 하나의 document를 만드는 작업을 진행하였다.

위의 데이터를 이용해 각각의 document들을 doc2vec 모델을 적용하여 트레이닝 시켜주었다. 이렇게 각각의 document들을 300차원의 벡터로 표현하고, 새로운 입력을 받아 입력받은 문장을 vector화 하여 해당 문장의 벡터를 구한다. 구한 벡터와 다른 벡터들간의 유사도 측정을 통해 가장 유사한 벡터들을 구하고 해당하는 벡터들의 상담 내용을 가져오도록 모델을 구성하였다.

세가지 모델이 합쳐져서 하나의 유사도를 이루기 때문에 word2hybrid라고 이름 지어보았다. 각각의 유사도의 비율은 수많은 시도를 통해 직접 가중치를 주었고, 그 결과로 제목간의 유사한 질문들도 잘 가져올 뿐만 아니라, 제목엔 쓰여있지 않지만 해당하는 내용을 가진 질문들도 같이 가져오는 것을 확인 할 수 있었다. 예를 들어 '천재지변'에 대해 검색 했을 때, 제목은 ' 빠른답변 바랍니다...' 였지만 해당 내용이 천재지변으로 인한 환불에 관한 내용을 가져오는 성능을 보였다.



ISAC model

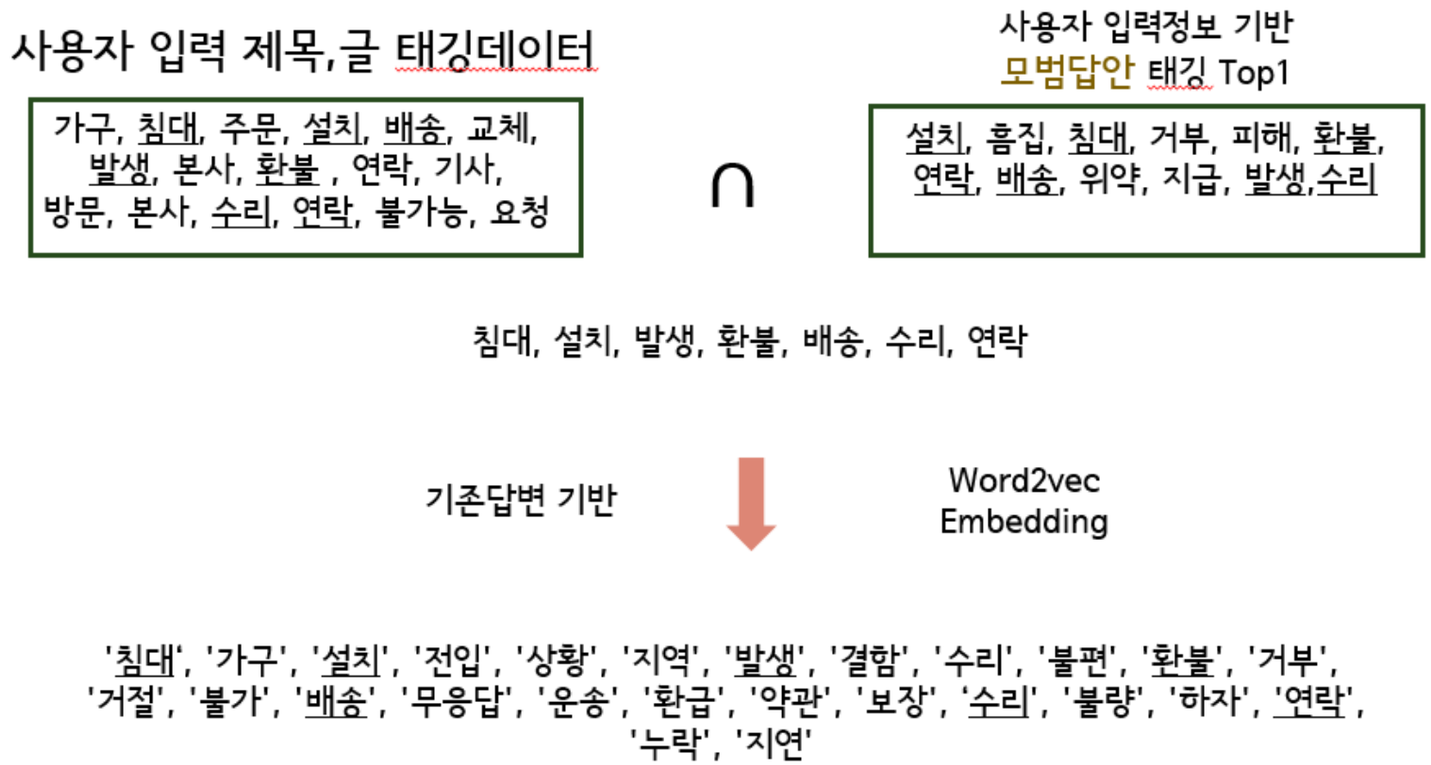
WORD2CRITERIA MODEL

word2criteria model

사용자의 input으로부터 제목과 분류, 그리고 질문내용을 가져온다. 선택한 중분류와 일치하는 모범상담 태그를 가져와 저장한다. ETRI API를 이용하여, 제목과 질문을 합친 텍스트의 명사 형태소를 생성하여 저장한다.

생성한 명사 형태소들과 모범상담 태그 데이터들을 활용하여, 명사 형태소 묶음과 각각의 모범상담 태그 데이터를 비교하였다. 여기서 가장 높은 수의 공통 단어를 기록한 모범상담을 기반으로 태그 달 단어로 선정한다.

태깅된 데이터를 학습시킨 word2vec 모델을 통해 생성한 태그의 각각의 대표 유사단어 4개를 추출한다.



word2criteria process

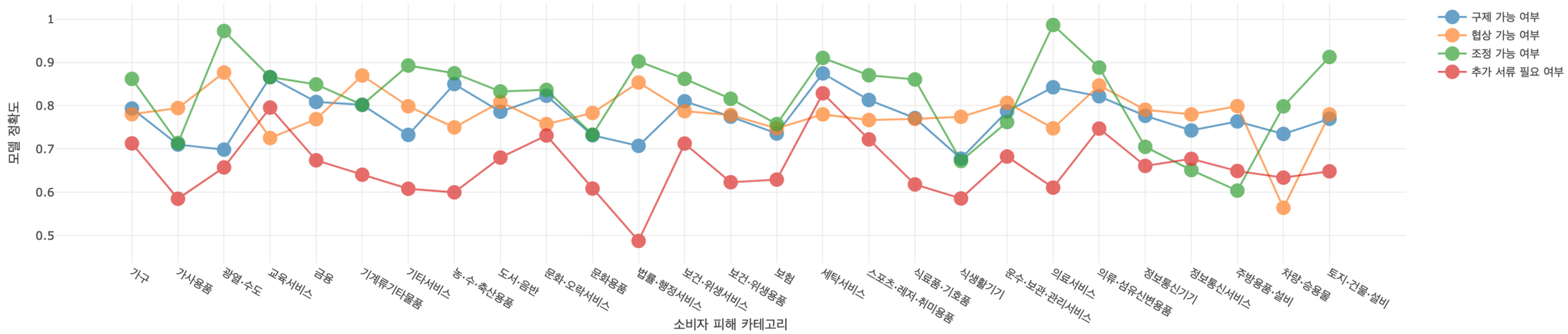
이를 기반으로 선택한 소분류 또는 중분류와 일치하는 해결기준을 불러오게 되고 일치하는 것이 없으면 해결기준이 없다고 판단하도록 모델을 구성하였다.

선택된 해결기준 또한 ETRI API를 사용하여 명사 형태소를 추출하고, 가장 높은 카운트를 기록한 해결기준의 상세내용을 가져오도록 구성하였다.

ISAC model

WORD2JUDGE MODEL

word2judge model



word2criteria process

사용자의 input으로부터 제목과 분류, 그리고 질문내용을 가져온다. 선택한 중분류와 일치하는 모범상담 태그를 가져와 저장한다. ETRI API를 이용하여, 제목과 질문을 합친 텍스트의 명사 형태소를 생성하여 저장한다.

생성한 명사 형태소들과 모범상담 태그 데이터들을 활용하여, 명사 형태소 묶음과 각각의 모범상담 태그 데이터를 비교하였다. 여기서 가장 높은 수의 공통 단어를 기록한 모범 상담을 기반으로 태그 달 단어로 선정한다.

태깅된 데이터를 학습시킨 word2vec 모델을 통해 생성한 태그의 각각의 대표 유사단어 4개를 추출한다.

이를 기반으로 선택한 소분류 또는 중분류와 일치하는 해결기준을 불러오게 되고 일치하는 것이 없으면 해결기준이 없다고 판단하도록 모델을 구성하였다.

ISAC UI

PROJECT MODEL UI

Donec Quis Nunc

⑤ 태깅 데이터를 학습시킨 word2vec 모델을 통해, 생성한 태그 각각의 대표 유사단어 4개를 추출

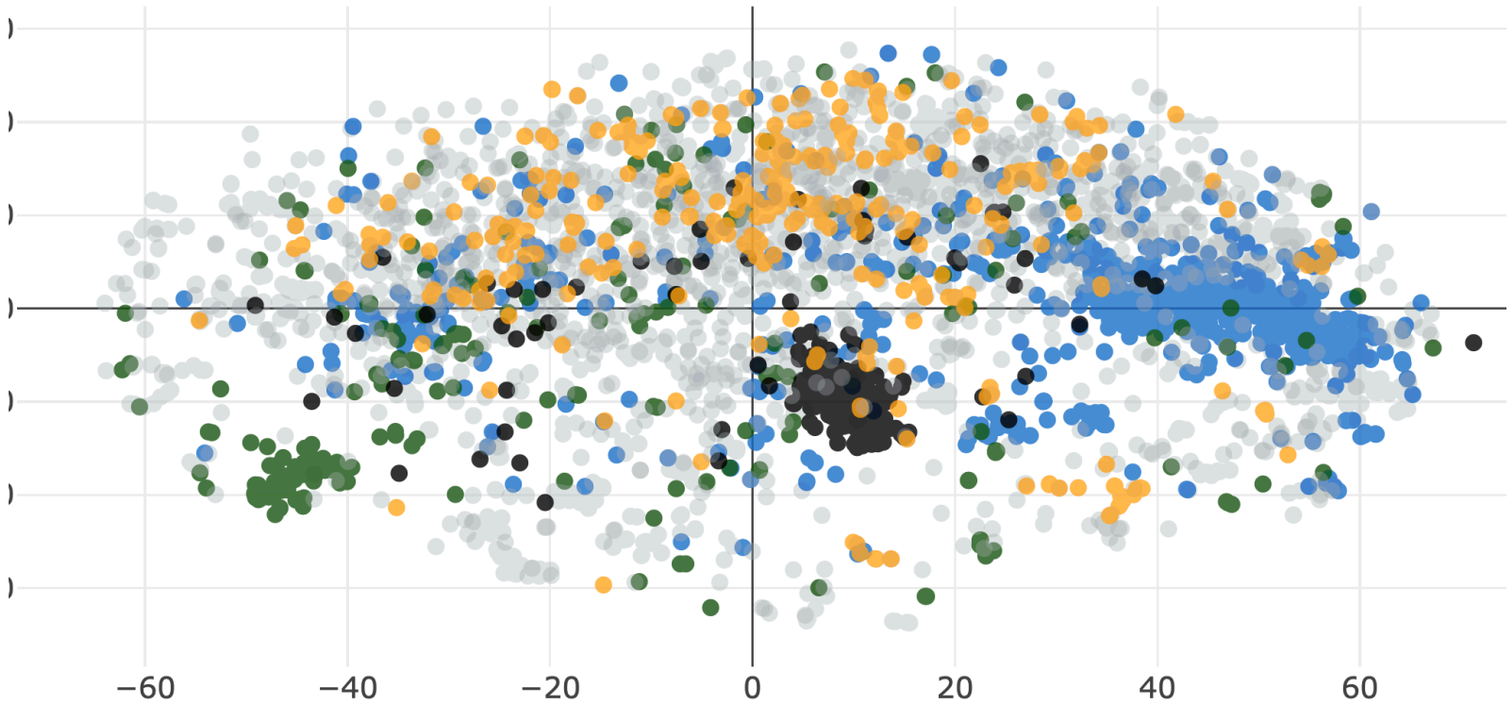
--> getSimilarTags( ) // 파라미터 : 생성 태그 리스트 // return : 생성 태그 포함 대표유사단어 태그 리스트

⑥ 선택한 소분류 또는 중분류와 일치하는 해결기준 불러오기(일치하는 것이 없으면 해당하는 해결기준이 없다고 판단)

⑦ 불러온 해결기준의 type1(최상위기준) 또한 ETRI API를 이용하여 명사 형태소

⑨ 상세 내용 str 형식으로 출력하기 끝!

--> showGijun( ) // 파라미터 : 선택된 특정 해결기준 리스트 // return : 상세 내용 문자열



Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum.

Donec Quis Nunc

Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit, congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet, sagittis felis sodales, dolor sociis mauris, vel eu libero cras.

Arcu habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec pellentesque leo, temporibus scelerisque nec. Ac dolor ac adipiscing amet bibendum nullam, lacus molestie ut libero nec, diam et, pharetra sodales, feugiat ullamcorper id tempor id vitae. Mauris pretium aliquet, lectus tincidunt.

Donec Quis Nunc

Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit, congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet, sagittis felis sodales, dolor sociis mauris, vel eu libero cras. Faucibus at. Arcu habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec pellentesque leo, temporibus scelerisque nec. Ac dolor ac adipiscing amet bibendum

Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum.

nullam, lacus molestie ut libero nec, diam et, pharetra sodales, feugiat ullamcorper id tempor id vitae. Mauris pretium aliquet, lectus tincidunt. Porttitor mollis imperdiet libero senectus pulvinar. Etiam molestie mauris ligula laoreet, vehicula eleifend. Repellat orci erat et, sem cum, ultricies sollicitudin amet eleifend dolor nullam erat, malesuada est leo ac. Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel, wisi at, id praesent.