자연스러운 범용 O2O 애플리케이션 사용자 인터페이스를 위한 상품 정보 자동 분류

이하나, 임은수, 조영인, 윤 영
홍익대학교 컴퓨터공학과
{ha21na, eundong933, 01n2100}@gmail.com, young.yoon@hongik.ac.kr

Automatic Classification of Product Data for Natural General-purpose O2O Application User Interface

Hana Lee, Eunsoo Lim, Youngin Cho, Young Yoon
Dept of Computer Engineering, Hongik University

요 약

본 논문은 현재 영역 별로 파편화된 여러 O2O(Online to Offline) 서비스들을 통합적으로 제공하기 위해 자연어를 통한 NUI(Natural User Interface)를 개발하여 사용자가 명시한 상품 정보의 항목을 자동으로 분류하고자 한다. 이를 위해 e-commerce 도메인 정보 학습에 적합한 나이브 베이즈 분류(Naive Bayes Classifier) 알고리즘을 사용한다. 학습에는 미국 e-commerce 사이트 Groupon의 상품 정보와 분류 체계를 사용하며, 학습 데이터의 특징을 분석하여 상품 정보에 특화된 학습 데이터 정제 및 TF-IDF(Term Frequency - Inverse Document Frequency)를 통한 단어 별 가중치를 적용하여 알고리즘의 정확도를 향상시킨다.

1. 서론

최근 '배달의 민족', '직방' 등 여러 영역의 O2O(Online to Offline) 앱들은 온라인 상 소비자의 요구에 오프라인 공급자가 즉각적으로 반응하고 저비용 고효율의 상품 홍보가 가능하게 하여 각광을 받고 있다. O2O 앱은 주로 요식업, 숙박업 등 특정 영역에 국한되어 파편화된 상태에서 서비스된다. 여러 영역의 상품을 판매하는 기존 E-commerce 사이트들의 경우, 공급자는 복잡하고 계층적인 분류 체계에 따라 상품 정보를 수동으로 분류해야 하는 등, 해당 사이트에서 요구하는 업종별 다양한 상품 정보 등록 및 검색 양식에 일일이 맞춰야 하는 불편함을 감수해야한다. 이와 같은 불편함을 해소하기 위해 사용자의 상품 홍보 또는 요청을 자연어로 받아들이고, 해당 상품을 자동 분류할 수 있는 NUI(Natural User Interface)를 개발하여, 공급자의 홍보 문구 입력 공수를 절감하고 소비자의 상품 검색 편의성을 향상시키고자 한다.

위와 같은 배경 속에서 본 논문은 새로운 상품 정보 분류 기법 연구에 초점을 둔다. 사용자가 명시한 상품 정보의 항목을 분류하기 위해 나이브 베이즈 분류(Naive Bayes Classifier) 기계 학습 알고리즘을 사용한다. 나이브 베이즈 분류는 지도 학습 데이터 크기가 클수록 좋은 성능을 보이며[1], 단순한 모델로도 (의료 데이터, 중국어 텍스트, RNA 분류 등) 많은 복잡한 실제 상황에서 잘 작동한다[2][3][4]. 우리는 분류될 항목들이 이미 주어져 있고 크롤링(crawling)을 통해 대량의 e-commerce 도메인 상품 정보를 점진적으로 수집하면서, 갱신된 상품 정보를 바탕으로 분류 모델을 지속적으로 진화시킬 수 있다는 측면에서 나이브 베이즈 분류가 가장 적합하다고 판단한다.

미국의 e-commerce 사이트인 Groupon[1]의 항목과 상품 정보를 학습 데이터로 사용하여 나이브 베이즈 분류 알고리즘을 테스트한 결과, 사용자가 만족하기엔 낮은 정확도를 보였다. 이에 우리는 학습 데이터의 특성을 분석하여 데이터를 정제하고 중요도에 따른 단어 별 가중치를 적용하여 정확도를 향상시키고자 한다.

본 논문의 나머지 내용은 다음의 구조로 서술된다. 2장은 나이브 베이즈 분류 알고리즘 선택 이유와 데이터 정제 방법, 가중치 적용 방법을, 3장에서는 구체적인 구현 내용을 설명한다. 4장에서는 개발된 기법의 성능을 평가하고 5장에서는 관련 연구에 대해 설명한다. 6장에서는 결론 및 향후 연구 방향을 설명한다.

2. 방법론

2.1 나이브 베이즈 분류 알고리즘

E-commerce 도메인은 다음과 같은 특징이 있다: (1) 웹상에 존재하는 상품 정보는 대부분 분류 항목이 존재한다 (2) 수집할 수 있는 데이터 량에 제한이 없다 (3) 데이터 형식이 다양하고 여러 변수가 많다. 웹상에 존재하는 많은 e-commerce 데이터는 대부분 분류된 항목을 가지고 있고, 크롤링을 통해 대량으로 수집할 수 있다. 데이터들은 각 사이트마다 제공하는 형식이 다르기 때문에 다양한 형태로 존재한다. 또한, 사람이 내용을 직접 작성하기 때문에 내용에 여러 가지 변수(간단히 작성, 유행어 사용, 작성하지 않음 등)가 존재한다. 이러한 특징을 고려해 지도학습이 e-commerce 도메인에 적합하다고 판단했다. 지도학습 알고리즘 중 나이브 베이즈 분류는 표준 텍스트 분류 알고리즘이며, 오랜 시간동안 성공적으로 사용되어 왔다[5]. 우리는 가변적이고 복잡한 상황에서도 잘 동작하며 데이터가 많을수록 높은 정확도를 낼 수 있는 나이브 베이즈 분류 알고리즘이 가장 이 도메인에 적합하다고 판단했다[1][2][3][4].

2.2 학습 데이터 정제

나이브 베이즈 분류를 더 효과적으로 사용하려면 도메인 내의 데이터 특성을 분석해 데이터를 정제해야 한다. 이를 위해 우리는 1차적으로 크롤링한 학습 데이터 6만 건을 분석해 다음 특징들을 찾았다: (1) 중복해서 등장하는 단어가 적고 대다수가 불용어(stop word)다 (2) 숫자, 기호 등 알파벳 외의 특수문자가 존재한다 (3) 단어 수가 매우 적은 데이터가 존재한다. 이런 특징을 고려해 다음과 같이 데이터를 정제했다: (1) 학습에 단어의 어근(語根) 사용, 불용어 제외 (2) 단어의 알파벳만 추출 (3) 단어가 2개 이하인 데이터 제외. 나이브 베이즈 분류는 단어가 어떤 항목에 등장할 확률에 기초한다. 그런데 우리가 수집한 데이터에는 같은 단어가 자주 사용되지 않았기 때문에 단어의 등장 확률을 높이기 위해 단어의 어근을 추출하여 학습시켰다. 예를 들면, run, runs, running의 등장 횟수는 어근인 run의 등장 횟수로 합쳐져 run의 등장 확률을 높였다. 또한, 자주 사용하는 단어를 확인한 결과, and, or, he 등 대부분의

1) www.groupon.com

항목에 존재해 학습에 혼란을 초래하는 불용어의 비중이 높아 정확한 학습을 위해 학습에서 제외했다. 또한, 데이터에 숫자나 기호 등 특수문자가 다수 있었고, 같은 단어에 숫자만 다른 데이터도 존재하여(ex. iphone6, iphone5 등) 학습의 정확성을 높이기 위해 알파벳만 추출해 사용했다. 마지막으로 적은 단어(2-4개)로 이루어진 데이터를 살펴본 결과, 단어 3, 4개로 이루어진 데이터는 의미 있는 단어들로 이루어져 학습 가치가 있을 것이라 판단했다. 하지만 단어 2개로 이루어진 데이터는 숫자로만 이루어진 데이터도 있고 충분히 의미 있는 단어의 비중이 적어 학습에 도움이 되지 않아 학습 데이터에서 제외했다. 우리는 위의 방법들을 적용하여 학습 데이터 정제를 적용한 나이브 베이즈 분류 알고리즘의 정확도를 측정하였고, 그 결과는 4장에서 설명한다.

2.3 단어 별 가중치 적용

학습 데이터 정제를 적용하여 실험한 결과, 정확도가 향상되었지만 중/소분류의 정확도는 사용자가 만족하기엔 낮았다.(약 62-78%) 우리는 정확도를 더 높이고자 단어들의 중요도 차이를 이용해 가중치를 부여했다. 다른 항목의 데이터에도 자주 등장할 수 있는 일반적인 단어는 학습 중요도가 떨어진다. 우리는 각 항목을 대표할 수 있는 단어를 추출하고, 중요도에 따른 가중치를 부여하고자 했다. 이를 위해 먼저 문서의 주제를 분류하는 데 자주 쓰이는 LDA(Latent Dirichlet Allocation) 방식을 고려했다[6]. LDA는 비지도 학습으로 주제의 수를 정해주면 스스로 문서들을 그 수에 맞게 분류한다. 그런데 우리의 항목 체계는 Women's Fashion, Men's Fashion과 같이 성별만 다르고 세부 항목은 같은 항목이 존재하는데 LDA를 적용했을 때 이를 구분하지 못했다. 즉, 스스로 주제를 분류하기 때문에 우리의 항목 체계에 맞게 주제를 분류할 수 없었다. 이런 문제점으로 인해 우리는 정해진 항목 체계 안에서 단어 간의 중요도 수치를 알아내기 위해 TF-IDF(Term Frequency-Inverse Document Frequency)를 이용했다[7]. TF-IDF는 어떤 단어가 사용된 문서에서의 비율과 전체 문서에서 쓰인 비율의 역수 값을 적절히 곱해 구한다. 즉, 단순히 단어의 빈도수로 중요도를 결정하지 않고 문서 전체에서의 희소성을 같이 고려한다. 따라서 TF-IDF 값이 높을수록 한 문서 내에 자주 쓰이면서 다른 문서에는 잘 쓰이지 않는 영향력 높은 중요 단어임을 나타낸다. TF-IDF를 통해 우리의 항목 체계에 맞게 각 항목의 주제어를 추출했고, 중요도를 고려해 객관적인 가중치 값을 부여했다. 이를 통해 중요도 수치가 매우 낮은 단어는 학습에 큰 영향을 미치지 않게 조정할 수 있었다. 우리는 위의 방법들을 통해 단어 별 가중치를 적용한 나이브 베이즈 분류 알고리즘의 정확도를 측정하였고, 그 결과는 4장에서 설명한다.

3. 구현

3.1 데이터 수집

우리는 e-commerce 사이트를 대상으로 연구에 활용하기에 적합한 데이터를 조사했다. 연구에 적합한 데이터란 항목이 빠짐없이 분류되어 있고 상품에 대한 설명이 모든 상품에서 일관성 있게 작성된 즉, 상품 설명의 길이 및 서술 방식이 일정한 데이터를 말한다. 항목이 존재하는 여러 사이트를 조사한 결과, 명사구만으로 이루어진 설명, 주어, 목적어, 동사 등이 모두 존재하는 완전한 문장으로 이루어진 설명, 상품 정보(size, color, width 등)를 표로 표현한 설명 등 상품 설명 방식이 다양했다. 일반적으로 사용자가 상품을 자연어로 설명하는 방식을 생각해보면, 완전한 문장으로 표현하는 사람도 있고 짧게 핵심적인 정보만 명사구로 표현하는 사람도 있다. 이를 고려할 때, 표로 된 설명을 제공하는 사이트는 우리가 사용하고자하는 '자연어'와는 맞지 않아 제외했다. 모든 상품에 대해 완전한 문장 또는 명사구로 이루어진 설명을 제공하는 사이트 중 설명마다 길이의 편차가 크지 않은 사이트를 조사한 결과, 미국 사이트 Groupon의 상품 정보 중 'Nutshell' 요소가 가장 긴 설명의 길이가 최대 34개 단어에 불과해 편차가 크지 않음을 발견했다. 또한, 대부분의 사이트가 설명이 존재하지 않는 상품이 많았던 반면, Groupon은 대부분의 상품에 설명이 존재했다. 따라서 우리는 명사구 또는 완전한 문장으로 상품을 설명하고 상품 설명 길이의 편차가 크지 않은 Groupon의 상품 정보를 학습 데이터로 결정했다. 미국의 e-commerce 사이트 중 소셜 커머스 방면에서 가장 활발한 Groupon은 체계적인 항목 체계(대/중/소분류)를 갖추고 상품에 대한 일관성 있는 설명(Nutshell)을 제공한다. Groupon 상품 데이터에는 Nutshell과 Description, Image 등 여러 요소가 있다. 이중 Nutshell을 항목과 함께 학습 대상으로 정해 상품 정보를 수집했다.

Scrapy 라이브러리[2]를 이용해 Groupon 상품 정보를 1차로 크롤링한 결과, 약 13만개의 Groupon 상품 데이터 중 66,508개의 데이터를 수집했다.(이하 1차 학습 데이터) Groupon의 상품 페이지가 20페이지로 한정되어 있어 웹상에 드러난 정보만을 크롤링했기 때문에 한계가 있었다. 이 후, 정확도를 올리기 위해 더 많은 학습 데이터를 모으고자 2차로 크롤링을 진행했다. 이 때 Groupon 내부에서 항목 체계를 일부 갱신했다. 2차 크롤링에서는 1차 크롤링 때 수집하지 못한 데이터와 그 사이에 추가된 데이터를 수집하기 위해 상품의 이름을 함께 크롤링했다. 이를 통해 이전에 Nutshell이 비어있어 수집되지 않았던 데이터를 수집할 수 있었고, 데이터에 상품의 이름이 추가되었다. 2차 크롤링 결과 86,367개의 데이터를 수집하였다.(이하 2차 학습 데이터)

3.2 학습 알고리즘

나이브 베이즈 분류는 NLTK(Natural Language ToolKit) 3.0 라이브러리[3]의 NaiveBayes Classifier 알고리즘을 기본으로 사용했고, 2.2, 2.3장의 방법을 추가했다. 언어는 Python 2.7.6을 사용했고, 변형된 알고리즘은 대략 [알고리즘 1]과 같다.

[알고리즘 1] 학습 데이터 정제 및 항목 별 핵심 단어에 가중치를 적용한 나이브 베이즈 분류 알고리즘

```
# train
make dataset from input file
    exclude nutshell under 2 words  #2.2
divide dataset into train_set and test_set
train with train_set  {
    load category index(catalog)
    load TF-IDF words and weight  #2.3
    make vocabulary(word) list
        apply stemming, remove stop words, extract alphabet  #2.2
    count the number of appearance of all each vocabulary
      apply weight to the number of appearance  #2.3   }
# test
test with test_set  {
     classify category of test data
     count the number of data that classified_category equals ac
     tual_category   }
calculate percentage of correct result
```

먼저 데이터를 파싱해 학습과 테스트 데이터(train_set, test_set)를 만든다. 학습 과정은 먼저 각 분류(대/중/소) 별로 항목 인덱스를 가져오고 존재하는 모든 단어의 리스트를 만든 뒤 각 단어의 등장 횟수를 구한다. 학습 후엔 테스트를 통해 정확도를 계산한다. 2.2장의 학습 데이터 정제를 적용하기 위해 2개 단어 이하 데이터 제외, 어근 추출, stop word 제거, 알파벳 추출 과정을 추가하였다(#2.2). 2.3장의 가중치 적용을 위해서는 학습 과정에서 TF-IDF 단어와 가중치를 불러오고, 최종적으로 계산된 각 단어의 등장 횟수에

2) https://scrapy.org/
3) http://www.nltk.org/

일정 수를 곱한 가중치를 적용하였다(#2.3).

3.3 TF-IDF 알고리즘 및 적용방식

TF-IDF를 이용해 항목 별 대표 단어들의 가중치를 구한 알고리즘은 [알고리즘 2]와 같다. 각 항목 별 대표 단어와 TF-IDF 값을 구하기 위해서 먼저 전체 항목(total_categories)에서 각 항목(category) 별로 단어들의 총 합(sum)을 구한다. 그 후, 해당 항목에서 사용된 단어 리스트(category's_words)의 단어(word)에 대한 TF-IDF 값을 구한다. 'TF' 값은 해당 항목에서 해당 단어가 등장하는 비율이고, 'IDF' 값은 전체 항목에서 해당 단어가 등장한 항목의 수다. 이 두 값을 곱하여 해당 단어의 TF-IDF 값을 얻는다.

[알고리즘 2] 항목 별 대표 단어의 TF-IDF값 추출 알고리즘

```
for category in total_categories:
    sum = total count of category's words
    for word in category's_words:
        tf[word][category] = word_count in (category / sum)
        idf[word][category]
            = log(total_categories + 1 / category_count
              that the word appears in total_categories
        tfidf[word][category]
            = tf[word][category] * idf[word][category]
```

우리는 위의 알고리즘을 통해 85,000건의 학습 데이터를 가지고 항목 별로 문서를 구분하여 각 항목에 해당하는 단어에 대해 TF-IDF 값을 구했다. 구해진 TF-IDF 값은 바로 학습에 반영하기에는 매우 작았다. 이를 위해 먼저 각 항목 별 TF-IDF 값들을 1-100 사이의 상대적인 값으로 정규화(normalization)해 적용하였다. 그 결과, 대/중/소 모든 분류의 정확도가 이전 실험을 통해 얻은 정확도보다 낮아졌다. 이를 통해 정규화의 역효과를 확인하고 정규화 없이 기존 가중치를 그대로 적용하였다. 그러나 여전히 모든 분류의 정확도가 기존 정확도보다 낮았다. 이에 가중치를 더욱 크게 주고자 TF-IDF 값을 단어에 적용할 때 일정 수를 곱해 가중치 크기를 증가시켰다. 곱하는 값은 대/중/소분류 별로 여러 수(100-100000 사이의 수)를 곱해보고 각 분류마다 가장 높은 정확도를 보이는 수를 적용했다. 그 결과, 대부분의 정확도가 이전 실험들에 비해 향상되었고, 대분류에서는 100, 중분류에서는 10000, 소분류에서는 1000이 가장 높은 정확도를 보였다. 마지막으로 가중치를 적용한 수치가 적용하지 않은 수치보다 작아지는 것을 방지하기 위해 일정 수를 곱한 가중치에 최종적으로 1을 더해 가중치를 적용하였다.

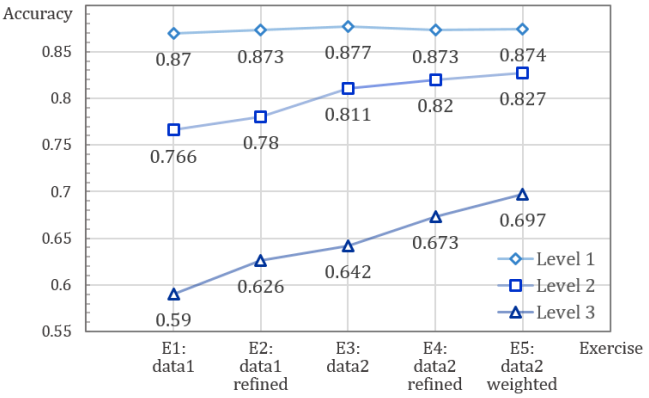
4. 성능평가

이 장에서는 기본적인 나이브 베이즈 분류에 2.2, 2.3장에서 제안한 방법을 적용해 성능을 평가한다. 성능 평가를 위해 Intel i5 CPU(3.20GHz, 3.33GHz), RAM 4G, 64bit 컴퓨터 5대와 Intel Quad CPU(2.83GHz, 2.83GHz), RAM 4G, 64bit 컴퓨터 2대를 사용했다. 모든 컴퓨터는 Ubuntu 14.04.4 LTS OS를 사용했고, 모든 알고리즘은 python 2.7.6 언어로 작성했다. 실험은 대/중/소분류 별로 진행했고, 분류 별 총 항목 수는 <표 1>과 같다. 2차 크롤링 진행 당시 Groupon의 항목 체계가 갱신되어 분류 별 항목 수에 변동이 있다.

분류	1차	2차
대분류	11	11
중분류	81	79
소분류	567	507

<표 1> 분류 별 항목 수

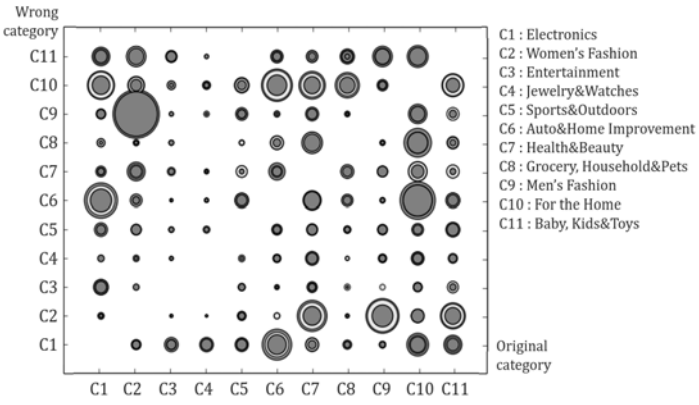
실험 결과는 (그래프 1)과 같다. 정확도는 테스트 데이터의 분류될 항목을 예측하여 크롤링해서 뽑은 실제 항목과 일치하는 수를 계산했다. 실험에는 1차 학습 데이터 60,000건, 2차 학습 데이터 80,000건을 사용했다. 실험 종류는 총 5개로, 'E1:data1'은 1차 학습 데이터로 학습시킨 NLTK에서 제공하는 기본 나이브 베이즈 분류의 정확도를 측정했다. 'E2:data1,refined'는 'E1:data1' 알고리즘에 2.2장의 학습 데이터 정제를 적용했다. 'E3:data2'는 2차 학습 데이터로 학습시킨 기본 나이브 베이즈 분류의 정확도를 측정했다. 'E4:data2, refined'는 'E3:data2' 알고리즘에 학습 데이터 정제를 적용한 실험이다. 'E5:data2,weighted'는 'E4:data2,refined' 알고리즘에 2.3장의 가중치를 적용했다. 각 실험은 학습과 테스트 데이터의 비율을 6:4, 7:3, 8:2로 바꾸며 정확도를 측정해 평균값을 계산했다.



(그래프 1) 실험 정확도 그래프

대분류를 제외한 중/소분류는 학습 데이터 정제, 가중치 적용을 통해 정확도의 향상이 확인되었다. 각 실험의 정확도 평균값을 기준으로 최저와 최고 정확도 사이의 정확도 향상 폭은 대분류 약 0.7%, 중분류 6.1%, 소분류 10.7%다. 실험 간 정확도 향상 폭의 평균은 대분류 약 0.1%, 중분류 1.5%, 소분류 2.7%다. 분류해야 할 항목 수가 많아질수록 정확도 향상 폭이 커지는 것을 알 수 있다.

대분류는 2차 학습 데이터를 사용해 데이터가 80,000건으로 증가된 후엔 학습 데이터 정제, 단어 별 가중치 적용을 통해 정확도가 향상되지 않는다. 우리는 그 이유를 찾기 위해 실험 E5의 대분류 테스트 결과에서 항목이 잘못 예측된 데이터를 분석했다. 해당 데이터의 원래 항목 별로 나눈 후, 각 항목 안에서 잘못 분류된 항목의 비율을 구해 데이터가 어떤 항목으로 잘못 예측될 확률이 높은지 계산했다. 계산 결과는 (그래프 2)와 같다.



(그래프 2) 잘못 분류된 항목 분석 그래프

(그래프 2)의 x, y축은 대분류의 11개 항목(c1부터 c11)을 나타낸다. x축은 데이터의 원래 항목이고, y축은 잘못 분류한 항목이다. 원의 크기는 전체 오답률 중 해당 항목의 오답률이 차지하는 비율을 나타내며, 해당 항목(x축 값)에서 잘못 분류된 항목들(y축 값) 사이의 비율을 나타내기도 한다. 가장 안쪽의 회색 원이 학습과 테스트 데이터 비율이 8:2일 때의 결과고, 그 밖의 흰색 원이 7:3, 가장 바깥쪽의 회색 원이 6:4일 때의 결과다. 이 중 큰 값을 가지는 부분을 살펴보면, c2(Women's Fashion)가 c9(Men's Fashion)로, c10(For the Home)과 c1(Electronics)이 c6(Auto&Home Improvement)으로 잘못 분류된 경우가 매우 많음을 알 수 있다. Women's Fashion과 Men's Fashion 항목의 데이터는 Women이나 Men을 제외

하면 의류를 설명하는 단어가 양쪽에 공통적으로 존재했다. For the Home과 Electronics 항목이 Auto&Home Improvement 항목으로 잘못 분류된 데이터에는 light, bulb, fan, flashlight 등 가정에서 사용하는 전기용품에 대한 단어가 많았다. 이를 통해 2차 크롤링 때 Groupon 항목이 갱신되면서 서로 간의 분류가 모호한 항목의 조합이 늘어났음을 알 수 있다. 여기에 해당하는 데이터들을 실제로 살펴본 결과, 원래 항목 뿐 아니라 잘못 분류된 항목으로 분류된다 해도 큰 어색함이 없는 것으로 판단되었다.

5. 관련 연구

본 장에서는 상품 데이터(홍보문구)의 항목을 자동으로 분류한 기존의 연구들을 소개한다. 먼저, 웹페이지와 광고의 항목을 분류하여 주제적 관련성에 기초해 적절한 웹페이지와 광고를 연결시키는 연구[8]에서는 Open Directory Project의 항목 체계를 정제 및 변형하여 분류 체계를 만들고, Merge-Centroid Classifier라는 새로운 분류기를 만들어 웹페이지 및 광고 데이터를 분류했다. Open Directory Project의 항목 체계는 모든 도메인의 항목을 포함하는 반면, 우리가 사용한 Groupon의 항목은 상품과 관련된 항목만 포함한다는 점에서 더욱 상품 정보 학습에 더 적합하다고 할 수 있다. 또한, [8]은 계층적인 분류에 중점을 두어 Centroid Classifier를 사용했지만, 우리는 복잡한 상황을 지속적이며 점진적으로 처리할 수 있는 단순한 모델에 초점을 두어 나이브 베이즈 분류를 사용했다. [9]에서는 information retrieval과 기계 학습을 이용해 상품 데이터를 분류했다. UN표준 상품 및 서비스 분류체계(UNSPSC)를 항목 체계로 사용하며, 학습 데이터에 전처리(어근 추출, 명사구 추출, 불용어 제외 등)를 적용하고 41,913개의 상품 데이터로 나이브 베이즈 분류의 정확도를 측정한 결과 최대 약 78%의 정확도를 보였다. 반면에 우리는 많은 사용자를 보유한 Groupon이라는 새로운 항목 체계를 도입하였고, 상품 설명 데이터에 맞는 전처리 단계를 추가했다. 또한, TF-IDF를 통해 가중치를 적용하여 알고리즘의 정확도를 최대 87.4%까지 향상시켰다.

나이브 베이즈 분류 알고리즘으로 상품 데이터를 분류한 연구도 있다[10]. [10]에서는 LDA를 통해 feature를 추출하고 나이브 베이즈, SVM(Support Vector Machine), KNN(K-Nearest Neighbor) 3개의 알고리즘을 이용해 상품 분류 정확도를 측정했다. 그 결과 데이터가 큰 경우 세 알고리즘 중 나이브 베이즈가 가장 높은 정확도를 보였다. [10]은 10,000개의 단어로 나이브 베이즈 분류를 사용해 86.2%의 정확도를 보였다. 우리는 이에 더하여 데이터 정제 및 가중치를 적용해 최대 87.4%의 정확도를 보였다. E-catalog 분류를 위한 확장된 나이브 베이즈 분류에 관한 연구[11]에서는 항목의 구조적 특성을 이용해 나이브 베이즈 분류를 확장한다. [11]은 이름, 가격, 설명 등 각 요소들을 정규화 한 후, 실험을 통해 각 요소마다 임의로 가중치를 주면서 가장 좋은 정확도를 보이는 요소를 찾았다. 그 결과 최대 86%의 정확도를 보였다. 우리는 TF-IDF 값을 구해 정규화 없이 일정 수를 곱해 요소가 아닌 단어 별로 더욱 세밀하게(fine-grained) 가중치를 적용했고, 대/중/소분류의 가중치를 각각 다르게 줬다. 그 결과, 최대 87.4%의 정확도를 보였다.

6. 결론 및 향후 연구 방향

본 연구는 여러 영역의 O2O 서비스를 '자연어'를 통해 하나의 인터페이스로 제공하는 범용 O2O 앱을 위해 먼저 상품 정보의 항목을 자동으로 분류하는 기술을 개발하였다. 미국의 e-commerce 사이트 Groupon의 상품 데이터를 학습 데이터로 사용했고, Groupon의 분류 체계를 따랐다. 학습 알고리즘으로는 나이브 베이즈 분류를 사용했고, 학습 데이터 정제 및 가중치 적용, 학습 데이터 증가를 통해 성능을 향상시켰다. 그 결과, 기본 나이브 베이즈 분류에 비해 최대 10.7%까지 정확도를 향상시킬 수 있었다.

분류 별 실험 결과를 보면, 중/소분류는 위 방법을 통해 정확도가 향상됐으나, 대분류에서는 미비했다. 이에 대분류 테스트 결과 중 항목이 잘못 분류된 데이터를 분석하여 모호한 항목의 집합을 발견했고, 이 집합 내의 데이터들이 서로에게 잘못 분류된다 해도 어색하지 않다고 판단했다. 따라서 우리는 예측 오답률 중 일부를 항목의 모호성 때문이라 보고 해당 데이터를 비슷한 항목 모두에 중복으로 분류하는 방법을 제안한다. 이를 통해 항목 간 모호함을 제거해 정확도를 더욱 향상시킬 수 있을 것이다.

본 연구를 통해 우리가 기여한 바는 다음과 같다. 먼저, 상품 정보의 자동 분류를 위해 많은 사용자를 보유한 e-commerce 사이트 Groupon에서 학습 데이터를 수집하는 새로운 체계를 마련했다. 또한, 학습 데이터 정제, 단어 별 가중치 적용 등 여러 방법을 적용해 Groupon 상품 정보 기반의 학습 가능성을 타진하였다. 실험 결과, 지금의 학습 체계가 최대 대분류 87.4%, 중분류 82.7%, 소분류 69.7%의 정확도를 보여 상품 정보 자동 분류에 가능성을 보임을 입증할 수 있었다. 실험 결과 분석을 통해서는 구분이 모호해 서로 혼동될 수 있는 항목의 집합이 존재함을 발견했다. 이런 상품은 모호한 항목 집합에 중복해서 분류하여 범용 O2O 앱에서 해당 상품의 분류 정확도 및 노출도를 향상시킬 수 있는 방법을 제안한다.

우리는 TF-IDF 뿐만 아니라 다른 가중치를 사용해서 가중치를 추가적으로 적용하거나 모델링을 계층적으로 세분화하는 등 여러 방법을 적용해 상품 정보 자동 분류의 정확도를 더욱 향상시킬 것이다. 사용자 요청의 항목을 자동으로 분류한 후에는 각 항목 양식에 맞게 가격, 재고 등 세부 정보를 추출하여 처리할 계획이다.

이 논문은 2016년도 정부재원(미래창조과학부 여대학(원)생 공학연구팀제 지원사업)으로 미래창조과학부, 한국연구재단과 한국여성과학기술인지원센터의 지원을 받아 연구되었습니다.

참고문헌

[1] Kohavi, Ron. "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid." KDD. Vol. 96. 1996.

[2] 한송이, 정용규. "의료데이터마이닝에서 클러스터링 기반의 나이브 베이지안 학습." 한국정보과학회 2010 한국컴퓨터종합학술대회 논문집 제 37 권 제 1 호 (C) 37.1C (2010): 410-413.

[3] Lu, S. H., Chiang, D. A., Keh, H. C., & Huang, H. H. "Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence threshold values." Knowledge-based systems 23.6 (2010): 598-604.

[4] Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. "Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy." Applied and environmental microbiology 73.16 (2007): 5261-5267.

[5] Gottipati, Srinivasu. "E-Commerce Product Categorization Srinivasu Gottipati and Mumtaz Vauhkonen."

[6] Blei, D. M., Ng, A. Y., & Jordan, M. I. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.

[7] Robertson, Stephen. "Understanding inverse document frequency: on theoretical arguments for IDF." Journal of documentation 60.5 (2004): 503-520.

[8] Lee, J. H., Ha, J., Jung, J. Y., & Lee, S. "Semantic contextual advertising based on the open directory project." ACM Transactions on the Web (TWEB) 7.4 (2013): 24.

[9] Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., ... & Fensel, D.. "Goldenbullet: Automated classification of product data in e-commerce." Proceedings of the 5th International Conference on Business Information Systems. 2002.

[10] Gottipati, Srinivasu. "E-Commerce Product Categorization Srinivasu Gottipati and Mumtaz Vauhkonen."

[11] Kim, Y. G., Lee, T., Chun, J., & Lee, S. G. "Modified naïve bayes classifier for e-catalog classification." Data Engineering Issues in E-Commerce and Services. Springer Berlin Heidelberg, (2006): 246-257.