# 4. CONTENT-BASED RECOMMENDER SYSTEMS

# 4.3 PREPROCESSING AND FEATURE EXTRACTION

---

- 모든 content-based(이하 CB) 모델에서 첫 단계는 item을 표현할 수 있는 discriminative features를 추출하는 것입니다.
- 여기서 discriminative features라고 하는 것은 user의 interests를 예측하기에 좋은 특징들입니다.
- 이 작업은 어떤 어플리케이션이냐에 따라 크게 달라집니다. 예를 들어 Web page recommendation system과 product recommendation system 에서의 작업이 달라져야 할 것입니다.

## 4.3.1 Feature Extraction

---

- 다양한 items에 대한 descriptions를 추출할 수 있을 것입니다. Multidimensional data representation(가격, 평점 등)과 같이 추출할 수도 있지만, 가장 일반적인 접근 방법은 data에 알맞는 keywords를 추출하는 것입니다.
  왜냐하면 많은 경우 제품에 대한 설명이 특정한 form을 따르지 않은채 글 형태로 묘사되어 있기 때문입니다.
- Multidimensional(structured) representation이 직접적으로 사용되는 경우도 있습니다. 예를 들어 attributes가 numerical quantities(e.g., price) 또는 색상 정보 같이 특정 범위에서 표현되는 필드 등 입니다.
- classification process에서 이런 features들은 적절하게 weighted되어야 합니다.
- *Featrue weighting* 은 *feature selection* 의 soft 버전이라 할 수 있겠습니다.
- feature seleciton의 경우 attributes는 포함되거나 포함되지 않거나로 결정됩니다. 반면에 feature weighting은 다른 가중치를 가진 attributes가 feature로 사용됩니다.

## 4.3.1.1 Example of Product Recommendation

- IMDb 영화 추천 사이트를 생각해보겠습니다. 각 영화들은 영화에 대한 설명과 감독, 배우, 장르 등의 정보를 담고 있습니다.
- 이 정보들을 가지고 user를 타깃으로 하는 attributes를 keywords형태로 뽑아 낼 수 있을 것입니다.
- 그런데 이 keywords들이 모두 같은 중요도를 가지는 것은 알 수 있습니다. 예를 들어 영화의 시놉시스 보다 어떤 배우 인가가 더 중요한 요소일 수 있습니다. 이런 경우 두 가지 방법으로 다룰 수 있습니다.
    1. Domain-specific knowledge를 이용할 수 있습니다. 예를 들어 영화 제목과 주연 배우들 이름에 더 큰 가중치를 임의로 줄 수 있습니다. 이런 중요도를 결정하는 기준은 도메인에 따라 다르게 고려될 것입니다. 이런 과정은 heuristic 하게 여러번의 시행착오를 통해 알아낼 수 있습니다.
    2. 다른 방법은 자동으로 feaures에서 상대적인 중요도를 학습하는 것입니다. 이런 과정을 feature weighting이라 불립니다. 이후 section에서 자세히 다뤄보겠습니다.

## 4.3.1.2 Example of Web Page Recommendation

- Web documents는 특수한 전처리 기술이 필요합니다. 왜냐하면 구조적으로 일반적인 특성들과 내부적인 연결성이 있기 때문입니다.
- Web document preprocessing는 두 가지 측면을 포함하고 있습니다. 일단 documents에 필요하지 않은 부분을 없애는 것입니다. 그리고 document의 실질적인 구조를 잡아내는 것입니다.
- Web document에서 모든 필드는 동등하게 중요하지 않습니다. HTML documents는 다양한 필드를 가지고 있습니다. 예를 들어 title, meta-data 그리고 document의 body입니다. 필드에 따라 중요도는 다르게 적용되어야 할 것입니다.
- Web document에서 또 다르게 특별히 처리해야하는 것은 anchor text입니다. anchor text에는 링크로 가리킨 web page에 대한 설명이 포함되어 있습니다. 이 text는 link page에 대한 축약된 설명이 될 수 있습니다. 하지만, 이 해당 페이지의 주제와 연관성이 떨어질 수 있기 때문에 document의 text에서 주로 삭제됩니다.
- Web page는 content blocks들로 구성될 때도 있습니다. 이 block들은 페이지의 메인 주제와 연관이 없습니다. 예를 들어 광고나 알림 같은 것들입니다. text mining의 quality를 결정하는 요인 중 하나가 바로 이런 main block과 연관 없는 blocks를 반영하지 않는 것입니다.
- blocks를 구분지어 분류하는 것은 쉬우나 main block을 구별해내는 것은 어렵습니다. 대부분의 자동화된 방법은 모든 documents들이 비슷한 layout을 따를 것이라는 전제를 기반하고 있습니다. 이때 layout의 구조를 학습하기 위해 사용하는 것이 사이트의 *tag trees* 입니다. *tree-matching algorithm* 으로 main block을 찾을 수 있을 것입니다. Machine learninig 방법으로는 main block을 labeling하고 classification 문제로 해결 할 수 있을 것입니다.

### 4.3.1.3 Example of Music Recommendation

## 4.3.2 Feature Representation and Cleaning

- unstructured format이 representation으로 사용될때 지금 다루게 될 방법들이 중요합니다.
- feature extraction phase는 제품이나 web pages에 대한 unstructured descriptions로 부터 bags of words를 정의할 수 있습니다. 하지만 프로세스를 위해 적합한 format을 가져야하고 정제되어야할 필요가 있습니다. 몇 가지 cleaninig process를 소개하겠습니다.
    1. Stop-word removal : 자주 등장하지만 구체적이지 않은 단어들을 걸러냅니다. 기준이 되는 stop-words를 가지고 다양한 text에 적용 가능합니다.
    2. Stemming : 파생어가 하나로 통합됩니다. 예를 들어, 단복수나 다른 시제의 표현들이 통합됩니다.
    3. Phrase extraction : 등장 빈도를 기반으로 함께 등장하는 단어 그룹을 찾아냅니다. 직접 dictionary를 만들어서 검색해도 되고 자동으로 추출하는 여러가지 방법들을 참고할 수 있습니다.
- 위 과정을 거치고 난뒤 우리는 keywords를 vector-space representation 해야합니다.
- Vector-space representaiton에서 documents들은 bags of words와 등장 빈도로 함께 표현됩니다. 단어의 등장빈도를 있는 그대로 사용할 수도 있지만, 이렇게 raw한 빈도 수를 직접 적용하지는 않습니다.
- 자주 등장하는 단어들은 대게 오히려 설명력이 떨어지기 때문입니다. 그래서 이런 단어들에는 낮은 가중치를 적용시킵니다. 이것은 stop-words와 비슷해보이지만 극단적으로 단어를 삭제하는 것이 아니라 soft하게 가중치를 낮추는 것입니다.
- 어떻게 할 수 있을까요? 한 가지 방법은 *inverse document frequency* $id_i$ 입니다.

$id_i = log(n/n_i)$ $\quad$ n:전체 단어 개수, $n_i$: i 번째 단어의 등장 횟수

- 여기에 극도로 자주 등장하는 단어의 중요도를 떨어트리기 위해서 damping function f() 를 곱해줍니다.
- 예를 들어, 검증되지 않은 사이트나 오픈 플랫폼에서 수집한 설명들에는 많은 spam들이 포함되어 있을 가능성이 큽니다.

$$f(x_i) = \sqrt{x_i}$$
$$f(x_i) = \log(x_i)$$

- Frequency damping 은 optional 이고 빠지기도 합니다. damping function을 뺀다는 의미는 $f(x_i)$ 를 $x_i$ 로 세팅하는 것과 같습니다.
- The _normalized frequency $h(x_i)$ for the i th word is defined by combining the inverse document frequency with the damping fucntion:

$$h(x_i) = f(x_i)id_i$$

- 이런 모델을 tf-idf model라고 부릅니다. tf 는 term frequency , idf 는 inverse document frequncey 를 뜻합니다.