December 10, 2025

▣ **강의명:** CSCI E-89B: 자연어 처리 입문

▣ **주차:** Lecture 05

▣ **교수명:** Dmitry Kurochkin

▣ **목적:** Lecture 05의 핵심 개념 학습

Contents

□ 핵심 요약

개요: 이 노트의 핵심 이 문서는 자연어 처리의 핵심적인 두 가지 텍스트 표현 기법을 다룹니다.
첫째, **TF-IDF**는 어떤 단어가 특정 문서 내에서는 자주 나타나지만, 전체 문서 집합에서는 드물게 나타날수록 중요하다고 판단하는 가중치 계산 방법입니다.
둘째, **단어 임베딩(Word Embeddings)**은 단어를 의미가 풍부한 저차원의 실수 벡터로 표현하여 단어 간의 의미적, 문법적 관계를 포착하는 기법입니다.
이를 통해 컴퓨터가 단순히 단어의 빈도를 세는 것을 넘어, 단어와 문서의 '의미'를 이해하도록 돕는 원리와 실제 구현 방법을 학습합니다.

□ 예제:

학습 로드맵

1. **기초 다지기:** 단어의 빈도만 세는 Bag-of-Words(BoW) 방식의 한계를 이해합니다.
2. **핵심 개념 (TF-IDF):** BoW를 개선하여 '중요한 단어'에 가중치를 부여하는 TF-IDF의 원리를 배웁니다.
3. **심화 개념 (단어 임베딩):** 단어의 '의미' 자체를 벡터 공간에 표현하는 단어 임베딩의 개념으로 나아갑니다.
4. **주요 모델:** 대표적인 단어 임베딩 모델인 Word2Vec과 GloVe의 차이점을 파악합니다.
5. **실습:** Python의 Scikit-learn과 Gensim 라이브러리를 사용해 TF-IDF와 Word2Vec을 직접 구현해봅니다.

주요 용어 정리

용어	쉬운 설명	원어	비고
TF-IDF	특정 문서에서 중요하지만 전체에서는 흔치 않은 단어에 높은 점수를 주는 가중치	Term Frequency-Inverse Document Frequency	키워드 추출, 문서 분류에 사용
단어 임베딩	단어를 의미를 담은 촘촘한(dense) 숫자 벡터로 변환하는 기법	Word Embedding	단어 간 의미 관계 포착 가능
Bag-of-Words (BoW)	문서를 단어의 순서는 무시하고, 출현 빈도만 담은 가방(bag)으로 보는 표현 방식	Bag-of-Words	가장 단순한 텍스트 표현
One-Hot Encoding	단어 사전에 있는 단어 중 하나만 1이고 나머지는 0인 벡터로 단어를 표현하는 방식	One-Hot Encoding	희소 (sparse), 고차원, 의미 없음
코사인 유사도	두 벡터 사이의 각도를 이용해 얼마나 유사한지 측정하는 지표. (1에 가까울수록 유사)	Cosine Similarity	단어/문서 벡터의 유사도 측정
불용어	분석에 큰 의미가 없는 단어들 (예: a, the, is, in)	Stop Words	전처리 과정에서 보통 제거
어간 추출 (스테밍)	단어의 어미를 잘라 어간(기본 형태)을 추출하는 과정 (예: cats → cat)	Stemming	형태적으로 단순화

1 TF-IDF (Term Frequency-Inverse Document Frequency)

1.1 핵심 개념: 왜 TF-IDF가 필요한가?

단순히 단어의 빈도만 세는 Bag-of-Words(BoW) 방식은 큰 한계가 있습니다. 예를 들어, 보스턴 지역 뉴스를 분석할 때 '보스턴(Boston)'이라는 단어는 모든 기사에 자주 등장할 것입니다. BoW 방식에서는 이 단어가 매우 중요하다고 판단하겠지만, 실제로는 모든 문서에 나타나므로 문서를 구별하는 데 아무런 도움이 되지 않습니다.

TF-IDF는 이러한 문제를 해결하기 위해 등장했습니다. 핵심 아이디어는 다음과 같습니다.

> **한 문서 안에서 자주 등장하는 단어(TF, Term Frequency)일수록 중요하지만, 여러 문서에 걸쳐 공통적으로 자주 나타나는 단어(DF, Document Frequency)일수록 중요도는 낮아진다.**

즉, 특정 주제를 잘 나타내는 핵심 단어에 높은 가중치를 부여하는 방식입니다.

1.2 계산 원리

TF-IDF는 **Term Frequency (TF)**와 **Inverse Document Frequency (IDF)** 두 값의 곱으로 계산됩니다.

□ 핵심 요약

TF-IDF 계산 공식

- **TF (단어 빈도):** 특정 문서 내에서 단어가 얼마나 자주 등장하는가?

$$\mathrm{TF}(t,d) = \frac{\text{문서 } d\text{에서 단어 } t\text{의 등장 횟수}}{\text{문서 } d\text{의 전체 단어 수}}$$

- **IDF (역문서 빈도):** 특정 단어가 전체 문서 집합에서 얼마나 희귀한가?

$$\mathrm{IDF}(t) = \ln\left(\frac{\text{총 문서의 수}}{\text{단어 } t\text{를 포함하는 문서의 수}}\right)$$

  - 이 공식에서 분모가 0이 되는 것을 방지하고, 모든 단어가 최소한의 값을 갖도록 실제 구현에서는 분모와 분자에 1을 더하는 '스무딩(smoothing)' 기법이 자주 사용됩니다.

- **최종 TF-IDF:**

$$\text{TF-IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t)$$

1.3 계산 예시

다음 4개의 문서가 있다고 가정해봅시다.

- **Doc 1:** "cat dog cat"
- **Doc 2:** "dog mouse dog"
- **Doc 3:** "dog mouse"
- **Doc 4:** "mouse cat dog"