구글 Trigram의 활용 사례: 문서 편집기로의 활용

황명권, 이효갑, 최동진, 김판구
조선대학교 컴퓨터공학과
e-mail:mghwang@chosun.ac.kr

Use Case of Google Trigram: Application to Text Editor

Myung-Gwon Hwang, Hyo-Gap Lee, Dong-Jin Choi, Pan-Koo Kim
Dept of Computer Engineering, Chosun University

요 약

본 논문은 구글(Google)에서 제공하는 n-gram 데이터가 사람들에게 실제로 어느정도의 효율성을 제공할 수 있는지에 대한 내용을 다루고 있다. 이를 위해, 구글의 trigram만을 이용하며, 규모를 줄이기 위해 필터링 과정을 거쳐 trigram 데이터베이스를 형성하였다. 그리고 이를 반영할 수 있는 텍스트 에디터를 구현하여, 사람의 타이핑 속도에 따라 얻을 수 있는 효율성을 측정하였다. 실험 결과에서 n-gram 데이터가 업무 향상에 효율성을 제공할 수 있지만, 여전히 큰 규모의 데이터집합에서 검색하는데 소모되는 시간의 한계점을 발견할 수 있었다.

1. 서론

본 논문은 구글(Google)에서 제공하는 n-gram 데이터가 사람들에게 실제로 얼마나 효과를 제공할 수 있는지에 대한 내용을 다룬다. N-gram 데이터는 웹에 존재하는 초대용량의 다양한 문서들을 분석하여 함께 연속으로 출현하는 단어(또는 음절) 집합 (특정 단어 (또는 음절)의 뒤에 오는 단어 (또는 음절))의 수치를 측정해 놓은 것이다. 이 수치는 단어들 사이의 밀집도(lexical cohesion)로 불릴 수 있으며, 다양한 분야에 활용될 수 있는 잠재력을 갖고 있다. 이러한 N-Gram은 대표적으로 문서 편집기[1]에서 가장 많이 사용되며, 최근에는 검색엔진에서 질의어 추천(Query Recommendation)[2], 음성 인식 후처리 분야[3] 및 장애인의 동공기반 타이핑[4,5]에 까지 연구에 활용되고 있다. 이러한 N-Gram 데이터는 사용자가 입력하고자 하는 단어의 정확도를 향상 시키며 입력 시간을 대폭 축소할 수 있다. 또한 이는 독립적인 입력장치를 갖는 데스크 탑 컴퓨터(desktop computer)보다 제한된 입력장치를 이용하는 모바일 단말기(mobile device)에서 활용할 때 더 큰 효과를 제공할 수 있다. 이러한 n-gram 데이터 중 최근 가장 대표적인 것은 Google n-gram이다. 하지만 그 규모가 너무 방대하기 때문에 활용단계에서는 어려움이 발생한다. 이 때문에 이를 활용하기 위한 연구만 이루어져 있을 뿐 실제 적용한 사례는 거의 없다. 이에 본 논문에서는 Google n-Gram 중 trigram을 실제 문서 편집기에 적용하여 성능을 평가해보고 그것의 원활한 활용을 위해 필요한 방법을 고찰한다.

본 논문의 구성은 다음과 같다. 2장에서 구글 n-gram에 대한 설명과 텍스트 에디터에 적용하기 위한 trigram 데이터 추출 방법을 기술한다. 3장에서는 trigram을 적용한 텍스트 에디터에 대해 설명하고, 4장에서 실험을 통하여 얻은 에디터의 성능을 기술한다. 마지막으로 5장에서 본 연구의 결론과 향후 과제에 대해 다룬다.

2. 구글 n-gram과 trigram 데이터 추출

구글 N-Gram은 LDC(Linguistic Data Consortium) 에서 제공하고 있다. N-gram은 영어 단어에 대해 5-gram 까지 형성하고 있으며, 2006년 1월까지 수집된 웹 문서에서 40번 이상 출현한 단어 토큰(tokens)에 대해 약 1 조개를 포함하고 있다[6]. [표 1]은 이와 같이 형성된 각 n-gram의 통계를 보이고 있다.

[표 1] 구글 n-gram 통계

n-gram	토큰 수
Unigram	13,588,391
Bigram	314,843,401
Trigram	977,069,902
4-gram	1,313,818,354
5-gram	1,176,470,663

구글은 세계 대표적인 검색 엔진이라 할 수 있으며, 이러한 구글에서 제공하는 n-gram 데이터는 다양한 분야에 활용할 수 있는 기반 데이터로써 가치가 충분하다. 이는 확률에 기반한 방법이긴 하지만 수많은 사람들의 집단 지

성을 포함하는 데이터이기 때문에 의미(semantics)가 내재되어 있다고 볼 수 있기 때문이다. 하지만, 이러한 구글의 n-gram은 그 규모가 약 86GB에 도달(압축을 해제한 실제 텍스트 파일 크기)하기 때문에 이를 효율적으로 이용할 수 있는 방법이 필요하다. 본 논문은 n-gram을 활용하기 위한 초기 연구이며, trigram을 활용한 결과만을 포함하고 있다.

구글의 n-gram은 단어, 숫자, 기호 들을 모두 포함하고 있다. 본 연구에는 영문으로 구성된 활용도가 높은 단어들을 중심으로 trigram을 구성하고, 이에 대한 성능을 분석하는 것이 목표이다. 이에, 구글에서 제공하는 trigram에서 다음과 같은 조건에 따라 필터링(filtering)을 거친다.

(a) 기호, 특수 문자를 포함하는 토큰는 제거
(b) 숫자를 포함하는 토큰 제거
(c) trigram으로 출현하는 횟수가 450번 이상인 것만을 이용

이러한 필터링 과정은 검색에 소요되는 시간을 단축할 수 있고, 활용도가 높은 토큰들로만 구성할 수 있다.

3. Trgiram 기반 텍스트 에디터

앞의 과정에서 구글 trigram을 활용하기 위한 전반적인 준비를 마쳤다. 본 장에서는 구성된 trigram을 텍스트 에디터에 적용하는 간단한 방법을 기술한다. 기존까지 n-gram 데이터를 적용하는데 자주 활용되는 것은 마코브 모델[7]과 베이지안 확률이다[8]. 하지만, 본 연구에서는 구글에서 제공하는 분포도에만 의존하며, 그에 따라 순위를 두어 단어를 추천한다. 그 활용 예는 다음과 같으며, 사용자가 “Korea information processing society (한국정보처리학회)”를 입력하려는 경우이다.

1. trigram 데이터를 이용하기 위해서는 최소 앞의 2단어를 기반으로 하기 때문에 2 단어가 입력될 때까지 시스템은 대기한다. 사용자는 ‘Korea information’을 입력하고 스페이스 바를 누른다. (총 18번의 타이핑 소요)
2. 사용자가 ‘Korea information”을 입력하면, 에디터는 {security, science, service, society, strategy, processing, certificate, ...} 을 순서대로 추천한다. 사용자는 ‘processing’을 선택한다(‘processing’을 선택하기 위해 1번의 타이핑을 소요).
3. 3번째 단어를 완성하면, 에디터는 “information processing”을 기반으로 {standards, systems, and, letters, standard, society, in, systems, ...}를 순서대로 추천한다.
4. 추천된 단어 리스트에 매치되는 단어가 없으면, 입력하고자 하는 단어의 첫번째 문자를 입력한다. 그러면 에디터는 앞의 두 단어와 입력한 문자를 함께 이용하여 다시 단어 집합를 추천한다. 하지만 3의 경우에는 바로 'society'가 존재하므로 이를 선택한다('society'를 선택하기 위해 1번의 타이핑 소요).

예제에서 사용자가 입력하고자 하는 문구는 공백을 포함하여 36개의 타이핑을 요구 하지만, trigram을 기반으로 할 경우는 20번으로 줄일 수 있다. 이와 같이 구글에서 작성한 trigram은 타이핑의 수를 줄이는 데는 효율적이긴 하지만, 그 데이터 자체의 양이 아주 방대하기 때문에 검색하는 시간에 영향을 받을 수 있다. 이에 다음 장에서 trigram이 제공할 수 있는 편리성과 시간절약을 사용자의 타이핑 속도에 따라 측정하여 보았다.

4. 실험

본 장에서는 구글에서 제공하는 trigram의 효율성을 평가한다. 먼저 trigram을 적용한 텍스트 에디터를 제작하였으며, 타이핑 속도를 달리할 수 있는 모듈을 제작하였다. 그리고 위키피디아(Wikipedia) 문서들의 작성에 기여하는 효율성을 평가하였다. 위키피디아 문서에서 초록부분만을 이용하여 200개 이상의 문자를 포함하는 것을 20개 선정하였다. 20개 문서가 포함하는 총 문자는 13,754개 (공백 포함)이다. [표 2]는 실험 결과를 보이고 있다.

[표 2] 실험 결과

타이핑 속도	시간 (초)		
	수동	Trigram	차이
180타/분	4580.3	3563.3	1017
300타/분	2751.3	2300.5	450.8
420타/분	1953.2	1709.5	243.7
타이핑 수	13,754	11,350	2,404
추천 수	0	763	

실험결과에서 trigram에 의해 추천된 단어 수는 총 763개 이며, 2404번의 타이핑 횟수를 감소시키는 효과가 있었다. 또한 타이핑이 느린 사람의 경우 (180타/분), 1017초 (대략 17분)의 이득을 볼 수 있는 것으로 확인되었다. 이는 시간적인 측면만 고려할 때 28.5%(수동타이핑을 trigram기반으로 나눈 것에서 1을 제외한 값)의 업무향상을 가져올 수 있음을 의미한다. 하지만, 타이핑 속도가 빠른 사람일수록 그 효율성은 낮아짐을 확인하였다. 이는 trigram 데이터가 대용량이기 때문에 단어를 추천하기 위해 걸리는 시간이 오히려 더 많이 소모되는 것으로 판단되었다.

5. 결론

본 논문에서 우리는 구글의 trigram을 실제 텍스트 에디터에 적용하여 봄으로써 그 효율성을 평가하였다. 실제로 trigram을 반영한 에디터는 속도가 느린 사람일수록 높은 효율성(업무향상)을 제공할 수 있는 것으로 나타났다. 우리는 주변에 무수히 많은 모바일 단말기를 갖고 있다. 가장 기본으로 모바일 폰(mobile phone)을 예로 들 수 있으며, 일반적으로 하나의 영어 문자를 입력하기 위해서는 1~3번의 키를 눌러야 한다. 또한 동공기반의 타이핑에서는 하나의 문자를 입력하기 위해 1~4회의 동공 스캔(pupil scan)이 필요하다. 이에 타이핑 속도는 현저히 떨어질 수

밖에 없다. 이러한 부분에 n-gram 정보를 적용한다면 그 효과는 아주 클 것으로 기대된다. 하지만, 이 부분에도 아직 해결해야할 부분이 존재한다. n-gram 자체의 규모가 너무 방대하다는 것이다. 큰 규모의 데이터는 휴대 단말기에서 설치상의 한계가 존재할 수 있으며, 많은 검색 시간을 소모하고 유지 보수를 어렵게 한다. 이에 사용자의 특성에 맞는 n-gram을 구축하는 방법이 필요할 것으로 보인다. 이는 지속적인 연구를 통해 해결해할 부분이다.

감사의 글

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No.2010-0011656)

참고문헌

[1] W. B. Cavnar and J. M. Trenkle, "N-Gram-Based Text Categorization," In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161-175, 1994.

[2] R. Baeza-Yates, C. Hurtado and M. Mendoza, "Query recommendation using query logs in search engines," Current Trends in Database Technology - EDBT 2004 Workshops, Vol. 3268, pp. 395-397, Nov. 2004.

[3] S. Khudanpur and J. Wu, "A Mazimum Entropy Language Model Integrating N-Grams and Topic Dependencies for Conversational Speech Recognition," In Proceedings of ICASSP'99, pp.553-556, 1999.

[4] 김순백, 이수흠, "특징 가중치 벡터를 적용한 능동 형태 모델 기반의 눈동자 움직임 추적", 한국신호처리시스템학회 2005년도 추계학술대회 논문집, pp.205-208, 11월, 2005.

[5] C. H. Morimoto, D. Koons, A. Amir, M. Flickner, "Frame-Rate Pupil Detector and Gaze Tracker", ICCV'99 FRAME-RATE workshop, Sep. 1999.

[6] Thorsten Brants and Alex Franz, Web 1T 5-gram Corpus Version 1.1 (LDC2006T13), April 2006.

[7] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Readings in speech recognition, pp. 267-296, 1990.

[8] E. T. Jaynes, "Probability Theory: The Logic of Science", June 1994.