Default 연산 알고리즘을 적용한 통계적 문맥의존 철자오류 교정 기법의 성능 향상

이정훈○, 김민호, 권혁철
부산대학교
{it_leejh, karma, hckwon}@pusan.ac.kr

Improving the Performance of Statistical Context-Sensitive Spelling Error Correction Techniques Using Default Operation Algorithm

Jung-Hun Lee○, Minho Kim, Hyuk-Chul Kwon
Pusan National University

요 약

본 논문에서 제안하는 문맥의존 철자오류 교정은 통계 정보를 이용한 방법으로 통계적 언어처리에서 가장 널리 쓰이는 샤논(Shannon)이 발표한 노이지 채널 모형(noisy channel model)을 기반으로 한다. 선행 연구에서 부족하였던 부분의 성능 향상을 위해 교정대상단어의 오류생성 및 통계 데이터의 저장 방식을 개선하여 Default 연산을 적용한 모델을 제안한다. 선행 연구의 모델은 교정대상단어의 오류생성 시 편집거리의 제약을 1로 하여 교정 실험을 하지만 제안한 모델은 같은 환경에서 더욱 높은 검출과 정확도를 보였으며, 오류단어의 편집거리(edit distance) 제약을 넓게 적용하더라도 신뢰도가 있는 검출과 교정을 보였다.

주제어: default 연산, 통계적 문맥의존 철자오류 교정, n-gram, 대용량 말뭉치

1. 서론

본 논문은 영어 맞춤법 교정을 실험하였으며, 전체 문장을 대상으로 연속된 세 어절(continuous trigram)을 사용한 통계적 문맥의존 철자오류 교정 기법[1, 2]을 기반으로 연구한 것이다. Google 1T 말뭉치와 같은 대용량 말뭉치의 연구[3-9]는 Google 1T가 처음 발표 되었던 2010년 ~ 2014년 사이에 활발히 이루어졌고 현재는 국내 부산대학교에서 연구를 진행 하고 있다. 통계적 맞춤법 교정은 말뭉치의 규모가 커질수록 경험적으로 정답에 가까운 데이터의 표본이 많아지므로 효과가 좋아지나 말뭉치의 규모가 클수록 통계 데이터의 검색 속도 및 저장 공간의 부족 문제가 생긴다.

기존의 방식[1, 2]에서는 교정을 시행할 때 3-gram의 교정대상 단어 또는 후보 단어의 통계 빈도를 얻기 위해서는 모두 검색을 시행하였고 말뭉치의 전체 통계 데이터를 n-gram의 형태로 저장 시 실제 말뭉치보다는 작지만 각각의 희소한 n-gram의 숫자도 통계 데이터의 규모에 따라 무시하지 못 하는 크기이다. 그렇기 때문에 본 논문에서는 이를 해결하고 더 나은 교정 성능을 위해서 자료구조와 알고리즘 개선을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 통계적 문맥의존 철자오류 교정 관련 연구현황에 대해서 분석하고, 기존 연구[1, 2]에 대해 설명을 한다. 3장에서는 통계적 문맥의존 철자오류 교정에 관해서 설명하고 논문에서 제시한 Default 연산, n-gram 저장 구조, 교정후보단어 및 오류단어생성 방식에 관해서 설명한다. 4장에서는 실험 결과를 바탕으로 기존 모델과의 성능을 비교 평가하고 편집거리에 따른 추가적 성능 평가를 한다. 마지막 5장에서는 결론 및 향후 연구에 관한 설명을 한다.

2. 관련 연구

문맥의존 철자오류(context-sensitive spelling error, real-word spelling error)는 대상 어절만을 볼 때는 맞을 수 있지만, 좌우의 문맥을 고려하였을 시에는 오류가 되는 것이다. 오류의 교정이 시급함은 영어에 대해서 문맥의존 철자 오류가 전체 철자오류의 30~40%임[10, 11]을 보이는 데에서 엿볼 수가 있다. 위 사실을 미뤄 봤을 때 문맥의존 철자오류만 잘 수정한다면 교정 성능에 상당한 영향을 줄 것이다. 문맥의존 철자오류를 교정하는 방법은 크게 규칙을 이용한 교정 방법과 통계정보를 기반으로 한 교정 방법으로 나뉘며, 다음은 본 논문에서 다루는 통계적 문맥의존 철자오류 교정 관련연구를 살펴본다.

통계적 문맥의존 철자오류 교정은 현재에도 연구가 꾸준히 되고 있는데 Church와 Gale[12]은 본 논문에서 기반이 된 샤론의 노이지 채널 모델(noisy channel model)로 영어 자소 사이의 오타 빈도를 통계적 행렬로 구축하여 오류 모델(error model)에 적용함으로써 오류를 교정하였다. Brill과 Moore[13]는 [12]의 연구를 기반으로 전체 영어자소 사이의 오타가 아닌 서로 쉽게 오타가 발생할 수 있는 자소들의 집합(set)으로 제약하였고 이를 적용하여 성능을 향상하였다. Golding[14]은 문맥의존 철자오류 교정 문제를 어의 중의성 해결(word sense

diambiguation)과 같은 문제로 해결한다. Mays[15]는 n-gram에 기반을 둔 언어모델로 대용량 말뭉치에서 어절 3-gram을 구하고, 이를 바탕으로 한 빈도를 기반으로 확률을 비교하는 방법으로 교정하는 연구를 진행하였다. Islam과 Inkpen[3-6]은 Google 1T말뭉치가 발표되면서 이를 이용하여 문맥의존 철자오류를 교정을 시행하고 마지막 논문에서는 백 오프(backoff)기법을 적용하여 교정 성능을 높이고 있다. Youssef Bassil과 Mohammad Alwani[7]에서는 Google 1T의 5-gram 정보를 이용하여 본 논문과 같이 노이지 채널을 이용한 문맥의존 교정을 통해 오류를 교정하였다. Xinxin Kou와 Evangelos[8]는 Google 1T를 이용한 문서 유사도 측정 방법인 GTM(Google Trigram Method)의 병렬화에 사용을 하여 속도의 향상을 보여주었다. Guymon R. Hall와 Dr. Kazem Taghva[9]은 Google 1T의 5-gram을 이용하여 검색엔진에서의 FCA(Formal Concept Analysis)의 군집화 기술의 성을 높였다. 마지막으로 이정훈[2]은 노이지 채널 모델을 기반으로 두 대용량 말뭉치의 보간(interpolation)을 통해 각 말뭉치의 장점을 공유하여 문맥의존 철자오류 교정의 성능을 높였다.

3. 통계적 문맥의존 철자오류 교정

본 논문에서는 통계적 언어처리에서 가장 널리 쓰이는 샤논의 노이지 채널 모형을 기반으로 한다. 이는 자연언어 처리 문제를 디코딩 문제(decoding problem)로 보고 아래 수식과 같이 베이즈 이론(Bayes'theorm)을 기반으로 수식(1)을 유도하여 사용한다.

$$\hat{I} = \arg\max_{I} p(I|O) = \arg\max_{I} \frac{p(I)p(O|I)}{p(O)} = \arg\max_{I} p(I)p(O|I) \quad (1)$$

출력 데이터의 확률 $p(O)$는 상수이며, 수식 (1)에는 두 개의 확률분포가 존재하는데, 언어 모형(language model)인 $p(I)$와 채널 확률(channel probability)인 $p(O|I)$이다. 그리고 수식(2)는 수식(1)을 기반으로 n-gram 모델에 맞게 해석하여 적용한 수식이다.

$$\hat{T} = \arg\max_{w_t \in T} \prod_{k=t-(N-1)}^{t+(N-1)} p(w_k|w_{k-(N-1)}, \cdots, w_{k-1}) p(Y|W)$$

$$\text{단, } \begin{cases} w_k \in L_C & \text{if } k<t \\ w_k \in f(T) & \text{if } k=t \\ w_k \in R_C & \text{if } k>t \end{cases} \quad (2)$$

L_C와 R_C는 T가 단어열 t의 위치에 나타날 때 함께 단어열에 나타나는 주변 문맥 단어로서 각각 왼쪽 문맥과 오른쪽 문맥을 나타낸다. 함수 $f(T)$의 공역은 노이지 채널의 잡음에 의해 T로 출력될 수 있는 모든 단어의 집합이다. N은 n-gram 모형의 차수를 나타내는 수로서 $N=3$인 3-gram 모형을 사용한다. 채널 확률 $p(Y|W)$에서 입력 단어열과 출력 단어열의 길이가 같고 $(m=n)$, 각 문자열에서 단어의 발생이 독립이라면 채널 확률 $p(Y|W)$는 수식(3)과 같다.

$$p(Y|W) \approx \prod_{k=1}^{n} p(y_k|w_k) \quad (3)$$

3.1. Default 연산

본 논문에서 제시하는 Default 연산은 후보 어절(candiate words)을 찾는 알고리즘으로 문장의 윈도우 길이5에서 (*a b* *) ∪ (*a* * *b*) ∪ (* *a b*)에 의해 동적으로 대상 어휘('*'를 만족하는)를 찾는 역할을 한다.'*'는 교정대상단어의 자리를 대체 할 후보단어로 오류단어생성이나 교정후보단어로 이용한다.

Default 연산이 필요한 이유는 다음과 같다. n-gram 중 가장 널리 쓰는 3-gram을 사용한다고 가정하고, 대상 단어 T에 대응하는 단어 w_i를 찾는 문제를 생각해보자, T에서 w_t를 구하는 문제는 confusion matrix를 이용하여 철자 오류를 찾거나, 시소러스를 이용해 유사 단어나 형제어 따위를 찾는 문제 등 다양한 문제로 해석할 수 있다.

$$c_{i-2}c_{i-1}w_t c_{i+1}c_{i+2} \quad (4)$$

앞에서 제시한 바와 같이 모든 $x_i \in V$ (모든 어휘)에서 관계 함수 f를 만족하는 $f(T, x_i)$로 w_t를 구하는 방법은 현실적으로 사용할 수 없다. 현재 가장 널리 사용하는 방법은 T에서 $f'(T)$, 즉 T에서 생성 가능한 모든 어절을 생성하고, 이 어절을 바탕으로 n-gram을 모두 사전에서 찾는 방법이다. 단, 함수 f'는 T에 연산을 적용해 가능한 어절(candidate word)을 모두 찾아서 집합으로 넘겨주는 함수다. 그러나 이 방법은 사전 검색이 엄청나게 많다. $f'(T)$의 모든 원소에 대해 각각 3번의 사전 검색을 해야 한다. 즉, (c_{i-2}, c_{i-1}, w_i), (c_{i-1}, w_i, c_{i+1}), 그리고 (w_i, c_{i+1}, c_{i+2})를 찾아야 한다. 만약 $f'(T)$가 n개의 원소라면 3*n번 사전 검색이 필요하며. 이 연산이 모든 어절에 대해 행해져야 한다. 그래서 Default 연산을 도입했다.

먼저 n-gram에서 정보를 (n-1)-gram에 Default를 적용해 구할 수 있는 연산을 도입한다. 그리고 Default에 해당하는 값을 연속으로 저장해 찾을 수 있게 한다. 예를 들어 $<w_1, w_2, *>$는 "$w_1 w_2$"로 시작하는 모든 3-gram의 세 번째 위치의 어절과 빈도를 가지게 된다. 이 연산을 바탕으로 먼저 T의 위치에 올 수 있는 모든 어절을 구한 결과를 CL(candidate lexicon)(5)이라고 하면 일반식은 아래 식처럼 된다. 이 식을 3-gram에 적용하면 $CL3$(6)가 된다.

$$CL = <L_C, *> \cup <L_C, *, R_C> \cup <*, R_C> \quad (5)$$

$$CL3 = <w_{-2}, w_{-1}, *> \cup <w_{-1}, *, w_1> \cup <*, w_1, w_2> \quad (6)$$

실제 자료에서 어절 n-gram에서 n이 3 이상으로 커지면 매우 희소한 행렬이 되므로, CL의 원소 개수는 많지 않다. 따라서 Default 연산을 쉽게 할 수 있는 자료구조가 만들어진다면 사전검색 횟수를 크게 줄일 수 있다.

3.2. 자료구조

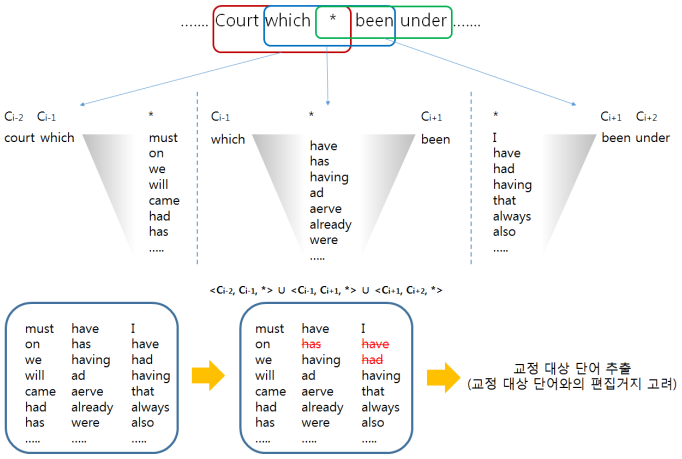

그림 1 Default 연산을 통한 교정 후보단어 검색

앞에서 제시한 Default 연산은 위해서 여기에 적합한 자료구조를 요한다. 가장 효과적인 자료구조로는 Trie 구조를 제시한다. Trie를 이용해 1-gram부터 단계적으로 자료에 접근하면 된다. 즉, $(c_{i-2}, c_{i-1}, *)$ 에서 '$c_{i-2}(1-gram) \rightarrow c_{i-1}(2-gram)$'을 따라가면 그 아래에 모든 가능한 어절에 접근할 수 있게 한다. 같은 방법으로 $(*, c_{i+1}, c_{i+2})$도 사전에서 접근할 수 있다면 두 집합의 합집합을 구하면 가능한 $w_i(t$에 올 수 있는 모든 어절)을 찾을 수 있기 때문에 Default 연산에 가장 적합한 자료구조로 본다. 하지만 현재는 이와 유사한 형태로 자료구조를 만들어 그림1과 같이 3-gram에서 추출한 2-gram정보에 연결되는 모든 '*'에 해당하는 단어와 빈도를 연결하여 사용하고 있으며 앞으로 실험을 진행하면서 자료구조 또한 Trie구조로 변경할 예정이다. 그림 1은 Default 연산을 통한 교정 후보단어의 검색 방식을 보여주고 자료구조로 인해서 통계 데이터의 저장 시 생기는 이점을 보여준다. 예를 들어 '*'를 중심으로 왼쪽 3-gram인 "court which *"의 '*'후보가 10개라고 가정하고 후보에 대한 3-gram의 데이터가 실제로는 "court which must", "court which on" 등으로 총 10(후보 단어 수) * 3(3-gram단어 수) = 30(총 단어 수)이지만 "court which"에 연결 된 단어들을 미리 연결해 두게 된다면 2("court which"단어 수) + 10(후보 단어 수) = 12(총 단어 수)으로 상당한 수가 줄어듦을 알 수 있다. 실제 예로는 표1을 참고로 Default연산에 맞는 자료구조를 적용 하였을 시에 속도가 향상되었음을 알 수 있다. 브라운 말뭉치에서 무작위 2000문장 총 56630단어를 교정 속도를 측정한 것으로 검색 속도가 향상 된 이유는 교정 후보의 통계정보를 모두 검색하던 방식에서 3-gram에서 교정 중심 단어 외의 두 단어를 검색 하면 후보가 연결되어 있는 구조로 바뀌었기 때문이다. 하지만 하나의 검색 데이터에 연결 된 모든 후보의 숫자가 몇 만개가 존재하였을 시에 처리 과정에서 성능 저하를 보일 수 있으므로 적절한 편집거리, 정규화, 키보드의 혼동 셋 필터링을 통해 적절하게 후보를 선택해야 한다.

	기존[1, 2]	적용 후
전체 검색 수행 시간	96 초	4.6 초
단어 당 검색 시간	0.00169 초	0.00008 초

표 1 교정 통계 데이터 검색 속도 비교

3.3. Default 연산 및 자료구조를 이용한 교정

그림 2 교정 순서

기존의 단어 생성 및 교정 후보는 예로 'had'라는 단어가 있을 시에 키보드 환경을 고려하며 혼동 집합을 미리 만들고 교정 대상 단어와의 편집거리에 따라 단어를 변경하여 'has', 'hat', 'hay' 등을 생성 후 1-gram 사전에 존재하는 단어 들을 후보 단어로 선택한다. 선택된 후보들을 교정 후보로 사용하기도 하고 그중에서 오류 단어를 임의로 선택하여 사용한다. 기존 방식의 문제점 중 하나의 예로 'the'라는 단어의 후보로 'rhe'라는 단어가 1-gram사전에 존재하므로 만약 'rhe'가 오류 단어로 선택되어 이를 교정하였다면 'the'와 'rhe'의 확률 비교를 통한 교정에서 당연히 'the'가 교정 될 것이므로 이는 교정 성능의 신뢰도를 낮추는 예일 것이다. 본 논문에서는 Default 연산을 통해 후보를 구하는 방식이 달라지며 3.2절에서 설명한 방식으로 후보를 미리 자료구조에 연결해 두어 한 번의 검색을 통해 후보 정보를 얻게 하는 방식이다. 즉, 말뭉치의 3-gram에서 실제 교정 후보의 자질이 있는 단어들 구하여 실험하기 때문에 교정 난도가 높아지므로 다음과 같은 환경에서 교정이 잘 되었을 시에는 신뢰도 또한 높아질 것이다.

오류 단어를 실험 문장에 적용한 오류 문장의 교정은 3절에서 제시한 통계 교정 모델을 이용하였다. 교정대상 단어와 후보단어의 확률 비교는 최대가능도추정(Maximum Likelihood Estimation)을 이용하여 비교를 한다. 수식3에서의 채널 확률 $p(Y|W)$는 오류율로 실제 교정 성능의 정밀도와 재현율을 사용자의 의도에 맞게 조절하여 사용하게 된다. 수식7을 참고로 교정대상단어와 교정후보 단어의 오류율 적용은 1을 100%로 본다면 오류율3%에서는 0.03을 교정후보에 곱하고 1에 0.03을 뺀 0.97은을 교정대상단어에 곱하여 두 단어 간에 차등을 두어 교정한다. 본 노문에서는 오류율 3%를 기준으로 실험을 진행하였고 이를 조절한 실험도 진행하였다.

$$p(Y|W) = \begin{cases} 1-\varepsilon & Y = W \\ \dfrac{\varepsilon}{N} & Y \neq W \end{cases} \quad (7)$$

4. 실험 및 평가

본 성능 실험에서는 1경 어절에서 구한 통계 자료인 Google 1T 말뭉치를 사용하며 평가 데이터는 브라운 말뭉치에서 무작위로 추출한 2,000문장을 사용하였다. 기존 교정 실험[1, 2]에서는 문장에서 편집거리 1의 제약을 주어 오류단어를 무작위로 생성하여 평가하였지만 본 실험에서 제안하는 방식은 데이터베이스에 저장된 Google 1T 데이터의 3-gram에 존재하는 후보 단어 중 편집거리를 계산 후 조건에 맞는 단어를 오류단어로 생성하여 평가한다.

오류 교정은 크게 검출(detection)과 교정(correction)으로 나뉘며, 오류의 검출 후 Default 연산을 통해 검색된 3-gram에 존재하는 많은 교정 후보 중에서 가장 높은 후보로 교정한다. 참고로 검출과정에서 나타나는 오류의 교정은 배제하지 않았으며 실험 결과에도 영향을 주며, 기존 교정 실험에서보다 실험 수치상 성능은 높으나 정확히 교정되는 것이 아닐 수 있다. 평가의 척도는 정밀도(precision), 재현율(recall) 그리고 이들의 관계를 하나의 수치로 표현한 F-Measure를 이용한다. 확률 추정은 최대 우도 추정(MLE : maximum likelihood estimation)을 사용한다. 오류율은 3%로 고정하여 사용하고 평탄화는 라플라스(add-one)를 사용하였다.

시스템	기존 시스템		제안한 시스템	
	검출	교정	검출	교정
정밀도	98.62%	96.73%	99.46%	97.94%
재현율	76.17%	74.71%	87.30%	85.96%
F1	85.95%	84.31%	92.98%	91.57%

표 2 기존 시스템과의 비교실험

표2는 기존 실험[2]와 Default 연산 알고리즘 및 새로운 저장 구조를 적용한 비교를 위해서 통계데이터, 오류율, 평가데이터 및 오류단어 생성방식을 동일하게 하여 어떤 시스템이 더욱 높은 성능을 보이는가를 실험하였다. 왼쪽은 기존 실험의 결과고 오른쪽은 새롭게 적용한 방식의 교정 결과로 검출에서는 큰 차이가 있으며 교정에서도 높은 차이를 보이고 있다. 즉, 위 실험은 교정 시스템 외에 모든 환경을 동일하게 진행하였으므로 두 시스템 간의 결과 값 차이는 신뢰도가 높다고 본다. 이런 차이의 이유는 자료구조의 개선에 따른 오류단어 및 교정 후보단어 생성을 3-gram에서 얻은 후보단어로 이용하므로 3.3절의 기존방식에서 보다 평탄화가 필요한 경우인 자료의 누락이 줄어 크게 영향을 주었다.

	검출	교정	검출	교정	검출	교정
오류율	1		2		3	
정밀도	95.89%	94.98%	95.18%	93.71%	94.67%	93.23%
재현율	65.70%	65.45%	69.64%	68.57%	72.26%	71.16%
F1	77.97%	77.50%	80.43%	79.19%	81.96%	80.71%
오류율	4		5		6	
정밀도	94.20%	92.72%	93.98%	92.49%	93.73%	92.23%
재현율	74.54%	73.37%	76.10%	74.89%	77.13%	75.89%
F1	83.22%	81.92%	84.10%	82.76%	84.62%	83.27%
오류율	7		8		9	
정밀도	93.43%	91.93%	93.25%	91.77%	93.06%	91.54%
재현율	78.42%	77.16%	79.26%	77.99%	80.18%	78.87%
F1	85.27%	83.90%	85.69%	84.32%	86.14%	84.73%

표 3 오류율 변화에 따른 교정 실험

표3은 Default 연산 알고리즘 및 새로운 저장 구조를 적용하고 오류 단어 생성은 Google 1T 3-gram에 존재하는 후보를 검색하여 편집거리1의 제약을 설정하여 오류율을 조절하며 진행한 실험이다. 오류율 3%로 설정하였을 경우 표2에서의 결과보다는 검출 및 교정 성능이 떨어지는데 표2에서는 오류 단어 생성을 키보드 상의 근접 단어 대치를 이용하여 생성한 후 1-gram 사전에 존재하는 단어 중 무작위로 선택하였기 때문에 실제 문맥과 관련된 단어를 오류 후보로 3-gram에서 선택한 표3의 실험과 교정 난이도가 달라진 것이다. 표3에서 보여준 오류율의 조절은 오류율을 높일수록 검출 및 교정의 정밀도는 떨어지고 재현율은 올라간다. 정밀도가 낮아지는 이유는 수식8, 9에서 교정으로 판단 된 단어와 바르게 교정 된 단어의 수가 오류율의 증가에 따라 높아지지만, 오류라고 판단 된 단어 수도 같이 늘어나면서 전체 교정단어 중에 바르게 교정 된 단어의 비중이 작아진 것이다. 그리고 재현율이 올라가는 이유는 수식10, 11에서의 생성된 전체 오류 단어보다 교정 대상으로 판단 된 단어 및 바르게 교정 된 단어의 수가 증가하면서 높아진 것이다. 오류율 증가에 따른 교정 대상이 증가하는 이유는 수식7을 이용하여 설명하자면 교정대상단어에 곱해지게 되는 정답률(1-ε)이 낮아져 오류율(ε)이 낮을 때 비해서 높을 경우 쉽게 교정이 이루어진 것이다.

$$\text{정밀도(검출)} = \frac{\text{교정대상으로판단된단어수}}{\text{오류라고판단된단어수}} \quad (8)$$

$$\text{정밀도(교정)} = \frac{\text{바르게교정된단어수}}{\text{오류라고판단된단어수}} \quad (9)$$

$$\text{재현율(검출)} = \frac{\text{교정대상으로판단된단어수}}{\text{생성된전체오류단어수}} \quad (10)$$

$$\text{재현율(교정)} = \frac{\text{바르게교정된단어수}}{\text{생성된전체오류단어수}} \quad (11)$$

	검출	교정	검출	교정	검출	교정
편집거리	1		2		3	
정밀도	94.67%	93.23%	94.12%	89.88%	92.73%	86.21%
재현율	72.26%	71.16%	75.19%	71.80%	75.39%	70.09%
F1	81.96%	80.71%	83.59%	79.83%	83.17%	77.32%

표 4 편집거리 변화에 따른 교정 실험

표4는 오류 단어 생성을 편집거리 제약에 따라 교정한 실험결과이다. 오류율은 3%로 편집거리가 늘어날수록 F1을 기준으로 교정 성능이 조금씩 떨어짐을 볼 수 있는데 이는 교정 대상 단어를 구할 때 편집거리가 늘어날수록 더 많은 교정 대상 단어들이 나타나게 되고 이들과의 비교가 많을수록 실제 확률 비교에서 정답 단어가 아닌 다른 단어로 교정이 이루어질 변수가 높아지게 되므로 생기는 현상이다. 이는 후보단어의 길이가 짧을수록 단어간의 편집거리에 따른 유사도가 크게 차이가 나기 때문에 길이에 따른 단계적 편집거리를 적용하여 실험한다면 성능이 더욱 올라갈 것이다.

5. 결론 및 향후 연구

본 논문에서는 샤논의 노이지 채널 모델을 기반으로 영어 자소 사이의 오류를 교정하였다. 본 논문에서는 Default 연산을 제안하여 데이터베이스에 저장되는 통계 데이터 구조를 변형하여 대상 단어의 교정 후보 단어들의 통계 데이터의 검색을 최소화하였다. 기존의 방식에서는 오류 단어의 생성을 편집거리1의 단어 중 1-gram사전에 존재하는 단어를 대상으로 교정하였지만, 논문에서는 Default 연산을 통해 검색 된 단어 중 편집거리를 계산해서 3-gram에서 존재하는 후보 단어 중 편집거리를 구하여 교정하는 방식으로 교정을 하고 있다. 즉, 평가 데이터의 경우 기존에는 무작위로 생성하므로 객관성이 떨어졌지만 실제 오류를 바탕으로 평가를 하여 오류 단어 생성 및 교정 단계에서 더욱 신뢰도가 있는 결과를 기대할 수 있었다.

향후 연구에서는 제안하는 모델에 맞는 방식으로 보간(interpolation) 및 평탄화(smoothing)를 적용하여 교정 성능을 높일 것이다. 통계 데이터를 검색하는 과정에서 Default 연산을 통해 데이터베이스 접근이 줄어든 대신 여기에서 구해진 실제 3-gram에 존재하는 후보 대상 단어의 개수는 많게는 몇 만개 이상이 되므로 이 부분에서 생기는 처리 속도의 저하 문제를 해결할 것이다. 마지막으로 Google 1T에 40 미만 빈도의 부제에 대한 다양한 보상 실험 통해 성능을 높일 예정이다.

참고문헌

[1] 김경식, 최성기, 권혁철. “극한 언어사용 환경에 적응적인 문맥의존 철자오류 교정 기법,” *한국정보과학회 학술발표논문집*, pp 654-656. (2015)

[2] 이정훈, 김민호, 권혁철. “말뭉치 간 보간 평탄화를 사용한 통계적 문맥의존 철자오류 교정 기법의 성능 향상” *한국정보과학회 학술발표 논문집*(2016)

[3] Islam, Aminul and Diana Inkpen "Real-Word Spelling Correction using Google Web 1T 3-grams," *Proceeding of International Conference on Natural Language Processing and Knowledge Engineering*, vol.3, pp.1241-1249, 2009.

[4] Islam, Aminul and Diana Inkpen "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions on Knowledge Discovery from Data*, vol.2, no.2, pp.1-25, 2008.

[5] Islam, Aminul and Diana Inkpen "Real-word spelling correction using Google Web 1T n-gram data set," *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp.1689-1692, 2009.

[6] Islam, Aminul and Diana Inkpen "Real-word spelling correction using Google Web 1T n-gram with backoff," *Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference*, pp.24-27, 2009.

[7] Youssef Bassil and Mohammad Alwani "Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information," *Computer and Information Science*, Vol. 5, No. 3, May 2012.

[8] Xinxin Kou and Evangelos Milios "Efficient Parallelization of the Google Trigram Method for Document Relatedness Computation," *ICPPW '15 Proceedings of the 2015 44th International Conference on Parallel Processing Workshops (ICPPW)*, Pages 98-104, 2015.

[9] Guymon R. Hall and Dr. Kazem Taghva "Using the Web 1T 5-gram Database for Attribute Selection in Formal Concept Analysis to

Correct Overstemmed Clusters," *12th International Conference on Information Technology - New Generations*, Pages 13-15, 2015.

[10] C. W. Young, C. M. Eastman, and R. L. Oakman, "An analysis of ill-formed input in natural language queries to document retrieval systems," *Information Processing and Management*, vol.27, no.6, pp.615-622, 1991.

[11] A. M. Wing, and A. D. Baddeley, *Spelling errors in handwriting: a corpus and distributional analysis, Cognitive processes in spelling*, London: Academic Press, pp.251-285, 1980.

[12] Kenneth W. Church and William A. Gale. "Probability scoring for spelling correction," *Statistics and Computing*, vol.1, No.2, pp 93-103, 1991.

[13] Eric Brill and Robert C. Moore. "An improved error model for noisy channel spelling correction," *Proceeding ACL '00 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp.286-293, 2000.

[14] Golding, Andrew R. and Dan Roth and J. Moon. "A Winnow-Based Approach to Context-Sensitive Spelling correction," *Machine Learning*, Vol. 34, pp.107-130, 1998.

[15] E. Mays, F. J. Damerau, and R. L. Mercer, "Context Based Spelling Correction," *Information Processing & Management,* vol.23, no.5, pp.517-522, 1991.