Denoising Transformer기반 한국어 맞춤법 교정기

박찬준[O1], 정솔[2], 양기수[1], 이수미[2], 조재춘[3], 임희석[1]

고려대학교 컴퓨터학과[O], LLsoLLu, 상명대학교 스마트정보통신공학과

bcj1210@naver.com , hisoka087@naver.com, willow4@korea.ac.kr smaseba@naver.com, jae@smu.ac.kr ,

limhseok@korea.ac.kr

Korean Spell Correction based on Denoising Transformer

Chanjun Park[O1], Sol,Jeong[2], Kisu Yang[1], Sumi Lee[2],Jaechoon Joe[3] Heuiseok Lim[1]

[1] Korea University Dept.Computer Science, [2] LLsoLLu, [3]SangMyung University.

요 약

맞춤법 교정이란 주어진 문장에서 나타나는 철자 및 맞춤법 오류들을 올바르게 교정하는 것을 뜻하며 맞춤법 교정 시스템이란 컴퓨터가 이를 자동으로 수행하는 것을 의미한다. 본 논문에서는 맞춤법 교정을 기계번역의 관점으로 바라보고 문제를 해결하였다. 소스문장에 맞춤법 오류문장, 타겟 문장에 올바른 문장을 넣어 학습시키는 방법을 제안한다. 본 논문에서는 단일 말뭉치로 한국어 맞춤법 병렬 말뭉치를 구성하는 방법을 제안하며 G2P(Grapheme to Phoneme)를 이용한 오류 데이터 생성, 자모 단위 철자 오류 데이터 생성, 통번역 데이터 기반 오류 데이터 생성 크게 3가지 방법론을 이용하여 맞춤법 오류데이터를 생성하는 방법론을 제안한다. 실험결과 GLEU 점수 65.98의 성능을 보였으며 44.68, 39.55의 성능을 보인 상용화 시스템보다 우수한 성능을 보였다.

주제어: 기계번역, 한국어맞춤법검사기, Transformer, 오탈자리스트

1. 서 론

맞춤법 교정이란 주어진 문장에서 나타나는 철자 및 문법적인 오류들을 올바르게 교정하는 것을 뜻하며 맞춤법 교정 시스템이란 컴퓨터가 이를 자동으로 수행하는 것을 의미한다.

맞춤법 교정은 음성인식 결과에 대한 후처리 모듈, 실시간 통역 시스템에서 번역 결과의 품질을 높이기 위한 사후처리 등 다양한 분야로 응용이 가능하다. 현재 한국에서 부산대, 네이버 등에서 성공적으로 맞춤법 교정기 서비스를 운영하고 있다. 이러한 서비스들은 대용량의 규칙기반 시스템으로 이루어져 있다. 규칙기반의 장점으로는 문장의 구조를 흐트러트리지 않고 정확히 틀린 부분만 고쳐낸다는 점을 들 수 있다. 그러나 규칙에서 벗어날 경우 수정하지 못한다는 단점이 있으며 대용량의 규칙을 구축하기란 쉽지 않은 문제이다. 본 논문에서는 맞춤법 교정시스템을 기계번역의 관점으로 바라보았다. 기계번역이란) 소스문장(Source Sentence)을 타겟문장 (Target Sentence)로 번역하는 시스템을 뜻하며 이를 맞춤법 교정시스템에 적용하게 될 경우 소스문장은 오류문장, 타겟문장은 교정문장으로 바라볼 수 있게 된다. 본 논문은 단일 말뭉치로 한국어 맞춤법 병렬 말뭉치를 구성하는 Unsupervised 방법을 제안하며 G2P(Grapheme to Phoneme)를 이용한 오류 데이터 생성, 자모 단위 철자 오류 데이터 생성, 통번역 데이터 기반 오류 데이터 생성 크게 3가지 방법론을 이용하여 맞춤법 오류데이터를 생성하는 방법론을 제안한다.

또한 최근 가장 성능이 좋다고 알려진 Transformer[1] 기반의 한국어 맞춤법 교정기 시스템을 제안한다. 실험결과 GLUE[2]점수 최대 65.98점이 나왔으며 기존에 상용화 되어 있는 맞춤법 교정 시스템보다 우수한 성능을 보였다.

2. 관련연구

한국어 맞춤법 교정기의 경우 부산대학교에서 활발한 연구가 이루어져왔으며 네이버, 다음 카카오에서도 상용화 서비스가 이루어지고 있다. 이전에 맞춤법 교정 시스템의 방식들을 살펴보면 규칙기반 맞춤법 교정 시스템[3,4], 통계기반 맞춤법 교정방식[5,6]을 거쳐 기계학습을 이용한 교정 시스템, 최근에는 신경망 기반 교정 시스템[7], 등 다양한 연구가 진행되어 왔다. 그러나 규칙기반 방식 같은 경우 규칙을 구축하는 것이 쉽지 않으며 구현이 어렵다는 단점이 있고 기계학습 방법의 경우 탐지 대상 단어의 주위 문맥이 올바르다고 가정하는 구조

적인 약점이 존재한다. [7] 기계번역의 관점으로 맞춤법 교정시스템을 바라보게 될 경우 고품질의 병렬 말뭉치만 있으면 별도의 규칙을 구축하지 않아도 다양한 양상의 맞춤법 오류들을 고쳐낼 수 있다는 장점이 있다. 그러나 병렬 말뭉치를 구축한다는 것은 쉽지 않은 문제이며 고품질의 병렬 말뭉치를 구축한다는 것은 더더욱 어려운 문제이다. 본 논문에서는 병렬 말뭉치 구축의 어려움을 해소하기 위해 단일 말뭉치만을 가지고 병렬 말뭉치를 구축하는 방법론을 제안한다.

3. Unsupervised 방식을 이용한 한국어 맞춤법 교정 병렬 코퍼스 생성

본 논문은 단일 말뭉치만을 이용하여 Unsupervised 방식을 이용한 한국어 맞춤법 Noise 생성 방법론을 제안한다. 핵심은 어떻게 Noise 데이터를 생성하는지에 있으며 해당 방법론은 아래와 같다.

- **Grapheme to Phoneme을 이용한 Noise 데이터 생성**
- **Edit Distance의 특성을 기반으로 하는 자모 단위 철자 Noise 생성**
- **실시간 통번역 시스템의 오탈자 리스트를 이용한 Noise 생성 방법론**

3.1 Grapheme to Phoneme을 이용한 Noise 데이터 생성 방법론

G2P란 Grapheme to Phoneme의 약자로 문장을 발음 나는 대로 바꾸어 주는 기술이다. 사람이 맞춤법을 틀리는 유형 중 대개 많은 부분이 발음 나는 대로 적어서 틀리는 경우가 많다. 이에 착안하여 G2P 기술을 통해 Noise 데이터를 생성하였다. G2P 기술을 통해 Noise를 생성하는 예시는 아래와 같다.

- 신을 신고 얼른 동사무소에 가서 혼인 신고 해라 ➔ 시늘 신꼬 얼른 동사무소에 가서 호닌 신고 해라
- 나의 친구는 계산이 아주 빠르다 ➔ 나의 친구는 계사니 아주 빠르다

G2P 프로그램 같은 경우 g2pk를 사용하였다.[1]

결론적으로 언어학에서 음성학적인 특성을 기반으로 하는 Noise 생성 방법론이다.

3.2 Edit Distance의 특성을 기반으로 하는 자모 단위 철자 Noise 생성 방법론

편집거리 알고리즘이란 두 문자열의 유사도를 판단하는 알고리즘으로 어떠한 문자열을 삽입, 삭제, 변경 총 3가지 연산을 기반으로 몇번의 연산을 통해 해당 문자열과 유사한지 다루는 척도이다. 대표적으로 레벤슈타인 알고리즘이 존재한다. 레벤슈타인의 수식은 아래와 같다.

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Edit Distance의 특성을 기반으로 하는 자모 단위 철자 Noise 생성 방법론 이란 “안녕하세요” 라는 문장이 들어왔을 때 “안년하세뇨”를 출력으로 내보내는 시스템으로 자모 단위로 랜덤하게 자음은 자음으로, 모음은 모음으로 변경 혹은 삭제 혹은 추가하는 Noise 생성 방법론이다. Noise를 생성하는 예시는 아래와 같다.

- 자모단위 랜덤 삭제
  예시: 안녕하세요 ➔ 안녀하세요 (‘ㅇ’ 삭제)
- 자모단위 랜덤 추가
  예시: 안녕하세요 ➔ 안녕하세용 (‘ㅇ’ 추가)
- 자모단위 랜덤 교체
  예시: 안녕하세요 ➔ 안령하새요 (‘ㅇ to ‘ㄹ’ 교체)

3.3 실시간 통번역 시스템의 오탈자 리스트를 이용한 Noise 생성 방법론

오탈자 리스트란 있어용, 있어요 등 단어 단위 오탈자 병렬 쌍을 의미한다. 오탈자 리스트는 LLsoLLu[2]에서 상용

[1] https://github.com/Kyubyong/g2pK

[2] http://www.llsollu.com/

화 서비스를 진행하고 있는 ezTalky[3] 통역비서 데이터를 이용하였다. 또한 국립국어원 맞춤법 교정 관련 자료를 해당 단어에 대한 오탈자 리스트를 구축하였다. 추가적으로 수작업을 통하여 오탈자 리스트를 수시로 추가하였다. 또한 임의로 자모 단위로 철자를 분리한 후 철자를 빼거나 다른 철자로 교체하여 오탈자 리스트를 보강하였다. 총 45,711개의 오탈자 리스트를 최종적으로 구축하였다. 실제 서비스를 하면서 구축한 오탈자 리스트이기에 신뢰성이 높은 데이터이다. 또한 실제 서비스를 진행하면서 구축한 데이터이기에 키보드 편집거리 에러가 포함된 데이터라고 볼 수 있다. 해당 리스트가 구축이 되면 입력으로 들어온 문장에서 오탈자 리스트에 있는 단어가 Matching이되면 Noise를 자동으로 생성하게 된다. Noise의 예시는 아래와 같다.

<그림1> 오탈자 리스트를 이용한 Noise 생성

4. 실험

4.1 데이터 셋

먼저 약 300만개의 신문기사 데이터 Crawling 진행하여 단일 한국어 말뭉치를 구축한다. 300만개 중 100만개는 G2P를 이용한 Noise 데이터, 100만개는 Edit Distance 기반 자모 단위 랜덤 Noise 데이터, 100만개는 오탈자 리스트를 적용한 Noise 데이터를 적용한다. Edit Distance 기반 자모 단위 랜덤 Noise 데이터 같은 경우 삭제, 추가, 교체 비율은 각각 33%,33%,34%의 비율로 선정하였다.

본 논문은 한국어 맞춤법 교정기 시스템을 구축할 때 규칙기반, 통계기반 시스템의 방법론은 일절 사용하지 않고 오직 기계번역의 관점으로 본 Task를 해석한다.

소스 문장에 Noise를 적용한 문장이 타겟 문장에 올바른 문장이 들어가게 된다. 추가적으로 소스문장에 기호를 붙이지 않고 타겟 문장에는 기호를 붙여 학습을 진행하였다. 이러한 데이터 변환으로 얻을 수 있는 효과는 문맥에 맞게 "?", ",." 등 즉 기호를 붙여준다는 특징이 있다.

<표1> 데이터 셋

Dataset	Size
Training (Total)	3.0M
G2P Noise	1.0M
Edit Distance Noise	1,0M
Error Lists Noise	1.0M
Validation	5,000

4.2 학습

병렬 말뭉치를 구축한 후 Transformer 기반으로 기계번역 훈련을 진행하게 된다. Transformer[1]란 Convolution과 Recurrence 없이 오직 Attention만을 이용한 기계번역 모델로 구글에서 2017년 제안하였다. Query, Key, Value를 기반으로 하는 Multi Head Attention을 기반으로 입력과 출력에 대해 각각 Self Attention을 학습하고 이후 입력과 출력 사이의 Attention을 학습하는 구조이다.

$$MultiHead(Q, K, V) = Concat(head_1, .., head_n)\, W^O$$
$$head_i = Attn(QW_i^Q, KW_i^K, VW_i^V)$$
$$Attn(Q, K, V) = softmax(QK^T/\sqrt{d_k})\, V$$

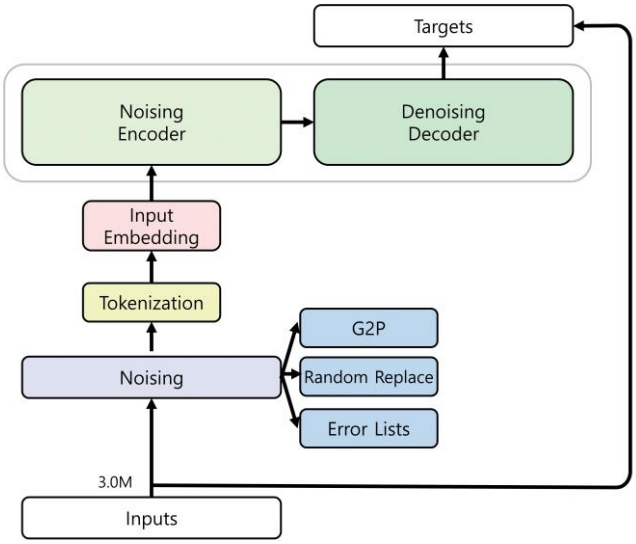
연산의 병렬화가 가능하여 다른 모델보다 학습시간이 빠르다는 장점이 존재하며 현재 기계번역 분야에서 좋은 성능을 보이고 있는 모델이다. Tokenize 같은 경우 단순 BPE[8]를 사용한다. 학습에 사용한 데이터와 vocab 사이즈 그리고 Hyper-parameter는 아래표와 같다.

<표2> Vocab 사이즈와 Hyper-parameter

Hyper-parameter	**Setting**
Source Vocabulary	32,000
Target Vocabulary	32,000
Batch Size	4,096
Word Vector Size	512
Attention Head	8
Transformer FF	2,048
Dropout	0.1
Optimizer	Adam
Decay Method	Noam

[3] http://www.systransoft.com/eztalky/

전체적인 시스템구조는 아래 그림과 같다.



<그림2> Model Architecture

5. 실험 결과

실험은 실제 한국에서 상용화되고 있는 맞춤법 검사 시스템과 성능비교를 진행하였다. 성능 평가 지표는 GLEU[2]를 사용한다. GLEU 같은 경우 BLEU와 유사하나 소스정보까지 고려한다는 점이 다르며 교정 시스템에 특화된 성능 평가 지표이다.

$$\mathrm{GLEU}\,(C,R,S) = \mathrm{BP}\cdot\exp\left(\sum_{n=1}^{N} w_n \log p'_n\right) \quad (1)$$

C는 교정한 문장, R은 Reference S는 Source 문장 즉 입력을 의미한다. 본 논문에서 N은 4를 이용하였으며 기타 설정은 기본 BLEU와 동일한 값을 사용하였다.

테스트셋 같은 경우 학습을 진행하기 전 미리 3000개의 문장을 랜덤하게 추출하여 사용하였다. 또한 객관성을 위하여 자모 단위 및 음절 단위로 추가적으로 노이즈를 강화하였다. 실험결과는 아래와 같다.

<표3> GLEU 기반 실험결과

Model	GLEU	BLEU
Commercial 01	39.55	48.11
Commercial 02	44.68	48.39
Ours	**65.98**	**67.65**

실험결과 상용화 시스템보다 높은 GLEU 점수 및 BLEU 점수를 보였다. N과 P는 각각 국내 상용화 시스템을 의미한다.

추가적으로 어절 단위 Precision, Recall, F-1 Score의 점수 비교 또한 진행하였다.

<표4> Precision, Recall, F-1 Score 기반 실험결과

Model	Precision	Recall	F1-score
Commercial 01	0.4024	0.1831	0.2517
Commercial 02	0.3404	0.3191	0.3294
Ours	**0.6727**	**0.7249**	**0.6978**

마찬가지로 Precision, Recall, F1-score 모두 본 시스템이 기존 상용화 시스템보다 우수한 성능을 보임을 볼 수 있었다.

부가적인 효과로 자동 문장분리, 자동 띄어쓰기, 문맥에 맞는 기호 부착 효과 등을 볼 수 있었다.

<표5> 자동 문장 분리 효과

입력	죄송합니다 모든 좌석이 매진됐습니다
출력	죄송합니다. 모든 좌석이 매진됐습니다.

<표6> 자동 기호 부착 효과

입력	여기 가까운 식당이 어디있습니까
출력	여기 가까운 식당이 어디 있습니까?

이를 통해 더 나아가 본 시스템을 음성인식 후처리 모듈로 사용할 수 있다. STT결과는 대개 기호가 부착되어 나오지 않으며 띄어쓰기가 간혹 올바르지 않을 경우가 있다. 또한 소리나는 대로 STT 결과를 도출하기에 사람이 느끼기에 맞춤법이나 문장의 흐름이 알맞지 않는 경우가 있다. 이러한 경우 본 논문에서 제안한 맞춤법 교정기를 이용하여 해당 문제를 해결하는데 도움이 될 수 있다.

6. 결론

본 논문은 Transformer를 한국어 맞춤법 교정기에 적용한 첫 시도이다. 또한 기존 상용화시스템을 능가하는 성능을 보여주었다. 또한 차별성 있는 Noise 생성 방법론을 제안하였다. 추후 Noise를 강화하는 방법에 대해서 연구를 진행할 예정이다.

7. 감사의 글

이 논문은 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터지원사업 (IITP-2018-0-01405), 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.

2018R1D1A1B07051369)

8. 참고문헌

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[2] Napoles, Courtney, et al. Ground truth for grammatical error correction metrics. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Vol. 2. pp. 588-593. 2015.

[3] Kwon, Hyuk-Chul & Kang, Mi-young & Choi, Sung-Ja. (2004). Stochastic Korean Word-Spacing with Smoothing Using Korean Spelling Checker. Int. J. Comput. Proc. Oriental Lang.. 17. 239-252. 10.1142/S0219427904001103.

[4] Kim, Minho & Choi, Sung-Ki & Kwon, Hyuk-Chul. (2014). Context-Sensitive Spelling Error Correction Using Inter-Word Semantic Relation Analysis. ICISA 2014 - 2014 5th International Conference on Information Science and Applications. 1-4. 10.1109/ICISA.2014.6847379.

[5] Lee, Jung-Hun & Kim, Minho & Kwon, Hyuk-Chul. (2017). The Utilization of Local Document Information to Improve Statistical Context-Sensitive Spelling Error Correction. KIISE Transactions on Computing Practices. 23. 446-451. 10.5626/KTCP.2017.23.7.446.

[6] Lee, Jung-Hun & Kim, Minho & Kwon, Hyuk-Chul. (2017). Improved Statistical Language Model for Context-sensitive Spelling Error Candidates. Journal of Korea Multimedia Society. 20. 371-381. 10.9717/kmms.2017.20.2.371.

[7] Woo Cho, Seung & Kwon, Hong-seok & Jung, Hun-young & Lee, Jong-Hyeok. (2018). Adoption of a Neural Language Model in an Encoder for Encoder-Decoder based Korean Grammatical Error Correction. KIISE Transactions on Computing Practices. 24. 301-306. 10.5626/KTCP.2018.24.6.301.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proc. of ACL