Journal of The Korean Association of Information Education Vol. 20, No. 2, April 2016, pp. 197-206

http://dx.doi.org/10.14352/jkaie.2016.20.2.197

통계 언어모델 기반 객관식 빈칸 채우기 문제 생성

박영기

서울대학교 컴퓨터공학부

요 약

빈칸 채우기 문제는 학생들이 학습 내용을 제대로 이해했는지 확인하기 위해 널리 사용되어 왔다. 이런 유형의 문제를 컴퓨터 알고리즘에 의해 자동으로 생성하는 많은 방법들이 제안되어 왔지만, 대부분 어떤 부분을 빈칸으로 만들면 좋을지에 대해 집중했기 때문에 적절한 보기를 자동으로 생성하는 연구는 미흡했다. 본 논문에서는 빈칸이 주어졌다고 가정하고, 이에 어울리는 보기를 자동 생성하는 알고리즘을 제안한다. 본 알고리즘은 통계 언어 모델에 기반하여 보기를 생성하기 때문에, 사람이 생성하는 경우보다 출제자에 편향되지 않은 보기를 제공할 수 있다. 또, 확률값에 기반하여 난이도를 자동으로 조절하는 것이 가능하기 때문에, 직접 사람이 문제를 만드는 것에 비해 상당한 비용 절감 효과가 있다. TEPS 문법, 어휘 시험에 대해 적용하여 실험한 결과, 사람과 유사한 결과를 생성함을 확인하였다. 향후 스마트 교육 분야에서 높은 활용도를 보일 것으로 기대한다.

키워드 : 통계 언어 모델, 객관식 문제 생성, 빈칸 채우기

Automatic Generation of Multiple-Choice Questions Based on Statistical Language Model

Youngki Park

School of Computer Science and Engineering, Seoul National University

ABSTRACT

A fill-in-the-blank with choices are widely used in classrooms in order to check whether students' understand what is being taught. Although there have been proposed many algorithms for generating this type of questions, most of them focus on preparing sentences with blanks rather than generating multiple choices. In this paper, we propose a novel algorithm for generating multiple choices, given a sentence with a blank. Because the algorithm is based on a statistical language model, we can generate relatively unbiased result and adjust the level of difficulty with ease. The experimental results show that our approach automatically produces similar multiple-choices to those of the exam writers.

Keywords : Statistical Language Model, Multiple-Choice Questions, Fill-in-the-blank with Choices

논문투고 : 2016-04-13
논문심사 : 2016-04-13
심사완료 : 2016-04-21

1. 서론

객관식 빈칸 채우기 문제는, 학생들이 학습 내용을 제대로 이해했는지 확인하기 위해 다양한 분야에서 사용되고 있다. 학교에서 수행 평가 또는 퀴즈 시험의 형태로 활용되는 것이 대표적이다. 또, TEPS와 같은 영어 시험에서는 <Table 1>과 같이 적절한 시제 또는 어휘를 고르도록 요구하기도 하고, 스탠포드 대학의 컴퓨터과학과에서는 운영체제의 동작 원리를 이해시키기 위해 소스코드의 빈칸을 채우는 방식으로 교육하기도 한다[10].

컴퓨터과학 분야에서는 이런 유형의 문제들을 자동으로 생성하는 방법이 연구되고 있다[2][6][9]. 자동 생성을 위해서는 어떤 부분을 빈칸으로 선택하느냐를 먼저 결정해야 한다. 이는 기계 학습 알고리즘에 기반할 수도 있고[6], TF-IDF와 같은 휴리스틱을 사용할 수도 있다. 그렇지만 어떤 단어(또는 구)가 빈칸이 되는지는 출제자의 성향과 의도에 따라 다르기 때문에, 이를 잘 모델링하기 위해서는 더 복잡한 알고리즘과 휴리스틱이 필요하다.

주관식 문제를 만든다면 위 과정만으로 모든 일이 끝나겠지만, 객관식 빈칸 채우기 문제를 만든다면 적절한 보기를 생성하는 단계가 반드시 추가되어야 한다. 상대적으로 이 단계는 학계에서 많이 연구되지 않았는데, 왜냐하면 어떤 단어나 구를 생성하는 것은 선택하는 것보다 훨씬 어렵기 때문이다. 이것은 컴퓨터뿐만 아니라 사람에게도 적용되는 사실이기 때문에, 이 문제를 해결할 수 있다면 더 큰 파급 효과를 기대할 수 있다.

본 논문에서는 객관식 빈칸 채우기 문제의 보기를 자동으로 생성하는 알고리즘을 제안한다. 이를 통해 얻을 수 있는 이점은 크게 두 가지다. 첫째, 사람은 자신이 평소 사용하는 어휘의 양에 한계가 있기 때문에 스스로의 경험에 편향된 보기를 생성할 가능성이 높지만, 컴퓨터는 훨씬 더 많은 어휘를 학습하기 때문에 그럴 가능성이 상대적으로 낮다. 둘째, 동일한 빈칸 채우기 문제라도 어떤 보기가 주어지느냐에 따라 난이도는 크게 달라진다. 본 알고리즘은 보기에 대한 확률값을 통계 언어 모델에 기반하여 계산함으로써 난이도 조절을 쉽게 할 수 있다. 본 논문의 구성은 다음과 같다. 2장에서는 연구 배경과 관련 연구를 소개하고, 3장에서는 객관식 빈칸 채우기 문제의 보기를 자동 생성하는 알고리즘을 제시한다. 4장에서는 자동 생성 알고리즘의 성능을 사람이 생성한 결과와 비교하여 나타내고, 자세히 분석한다. 마지막으로 5장에서 결론 및 향후 연구 방향을 제시한다.

2. 이론적 배경

스마트교육을 위해 필요한 핵심 요소 중 하나는 스마트 컨텐츠이다[8][12]. 그러나, 스마트 컨텐츠를 개발하고 성공적으로 활용한 사례들이 제시되고 있는 반면[7][3], 아직까지 컨텐츠 개발을 자동화할 수 있는 연구는 많지 않은 실정이다. 이를 지원할 수 있는 도구들이 일부 개발되어 있기는 하지만, 대부분 보조적 역할만을 수행하기 때문에 실제 컨텐츠를 만들기 위해서는 많은 노력이 필요하다. 교육 컨텐츠의 종류는 다양하기 때문에 모든 컨텐츠 제작을 완전히 자동화한다는 것은 불가능하지만, 일부 컨텐츠에 대해서는 현 기술로 자동 생

<Table 1> Example Sentences with a Blank

Sentence with a blank [4]	choice 1	choice 2	choice 3	choice 4
Joseph is preparing for tomorrow's big ___ to the president	and	is	enough	presentation
Mr. Singh listened ___ to the president's announcement	to	is	intently	carefully
The PR person is the one in charge of ___ meetings and finding accommodations for our associates.	the	of	these	scheduling
Ms. Havlat received a memo from the CEO ___ the employees' conduct.	of	the	and	regarding
The amount of money in the budget decreased ___ over the past year.	by	all	from	significantly
Mr. Gomez is ___ quickly; however it will be a log time before he gets used to the job.	a	as	not	learning
The boss can never get around to ___ off his desk.	the	cut	take	cleaning

성하는 것이 가능할 수 있다.

최근 들어 학생들에게 출제할 문제를 자동 생성하는 연구가 수행되고 있다. 자동으로 생성할 수 있는 문제의 종류는 다양한데, 그중 가장 쉬운 형태는 빈칸을 포함한 문장과 여러 개의 보기를 제시하고, 그 중 가장 적합한 한 가지를 고르는 것이다. 조금 더 복잡한 형태는, 주관식 문제를 생성하는 것인데, 앞선 경우와 달리 기계가 완전히 새로운 형태의 문장을 생성해야 하기 때문에, 아직은 실험적 단계에서 연구가 진행되고 있다. 더 복잡한 형태는 텍스트가 아닌 이미지까지 활용하여 문제를 생성하는 것으로, 해결하기 어려운 문제이지만 최근 딥러닝(Deep Learning) 기술이 발전하면서 기대를 모으고 있다.

본 논문에서는 이미 여러 분야에서 그 효과가 증명된 [13][14][15] 객관식 빈칸 채우기 문제를 자동 생성하는 것에 초점을 맞춘다. 이 문제는 빈칸을 생성하고 객관식 보기를 제시하는 2가지 단계로 이루어진다. 대부분의 기존 연구들은 첫 번째 단계인 어떤 것을 빈칸으로 만들지에 대해서는 많이 고민했지만[4], 두 번째 단계인 보기를 만드는 작업에 대해서는 연구가 미흡했다. 빈칸을 효과적으로 만들더라도 보기가 충실하지 못하다면 문제의 가치는 떨어질 수밖에 없기 때문에, 두 번째 단계에 대한 깊은 고민이 필요하다.

본 논문의 기본 아이디어는 어떤 것이 좋은 보기이고 나쁜 보기인지 순위를 매긴 다음, 그것들을 사용자 요구에 맞게 적절히 섞는 것이다. 예를 들어 <Table 1>의 'choice 1'은 바로 직전 단어와의 상관 관계만 고려하고, 'choice 2'는 바로 다음 단어와의 상관 관계만을 고려하여 제시된 것이다. 'choice 3'은 직전 단어와 다음 단어와의 상관관계가 모두 큰 것을 자동 생성한 것이고, 'choice 4'는 문장에 있는 모든 단어를 고려하여 생성하였다. 명백히 choice 1, 2에 비해 choice 3이 더 좋을 것이라 기대할 수 있고, choice 4는 3보다 더 좋은 것으로 알고리즘에 의해 예상될 수 있다. 이 예에서는 4가지 단어에 대한 순위만을 매겼지만, 만약 주어진 빈칸에 대해 모든 단어의 순위를 매길 수 있다면 사용자는 어떤 순위의 단어를 보여줄지 결정하기만 하면 된다.

3. 통계 언어 모델 기반 객관식 문제 생성

3.1절에서는 객관식 문제 생성을 위해 선행되는 통계 언어 모델에 대해 설명한다. 3.2절에서는 이를 활용한 객관식 문제의 자동 생성 방법에 대해 논의한다.

3.1. 통계 언어 모델

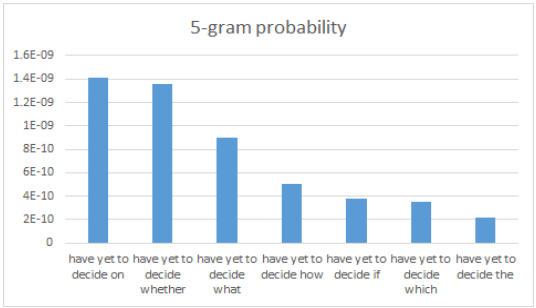
통계 언어 모델은 학습 말뭉치 내에 나타난 연속된 단어들의 빈도수를 활용한다. 본 논문에서는 n-gram을 활용한 언어 모델을 사용한다.

(정의 1) n-gram: i번째 단어를 w_i라 할 때, n-gram은 연속된 n개의 단어 $w_i, w_{i-1}, ..., w_{i-(n-1)}$로 정의한다. 예를 들어 'I go to school'은 4개의 단어가 연속되므로 4-gram이다.

(정의 2) n-gram 확률: n-gram 확률은 n-gram이 나타날 확률을 의미한다. $w_i, w_{i-1}, ..., w_{i-(n-1)}$으로 구성된 n-gram이 나타날 확률은 다음과 같이 계산된다.

$$P(w_i, w_{i-1}, ..., w_{i-(n-1)}) = \prod_{k=i-(n-1)}^{i} P(w_k | w_{i-(n-1)}, w_{i-(n-2)}, ..., w_{k-1})$$

(Fig. 1)은 5-gram 확률 값들의 예를 나타낸 것이다. 그림에서와 같이 'have yet to decide on'이 'have yet to decide the'보다 5-gram 확률 값이 높다.



(Fig. 1) Example 5-gram probability

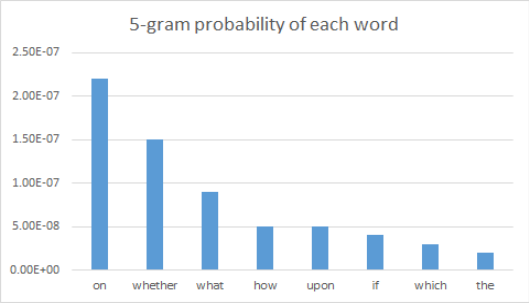
'I __ to school'과 같은 빈칸이 주어지면 해당 빈칸에

는 반드시 하나의 단어가 대응된다고 가정했을 때, 빈칸에 들어갈 어떤 단어 w_i의 n-gram 확률을 다음과 같이 정의한다.

(정의 3) 어떤 빈칸에 대응하는 단어 w_i의 n-gram 확률:

$$P(n-gram(w_i)) = \max_{i \leq j \leq i+(n-1)} P(w_j, w_{j-1}, \ldots, w_{j-(n-1)})$$

Fig. 2는 'I have yet to decide __ the exact terms' 문장에 대해 각 단어들의 5-gram 확률을 나타낸 것이다. 예를 들어 'on'의 5-gram 확률 값은 다음의 5-gram 확률 값들 중 최대치를 취한 것이다:



(Fig. 2) The 5-gram probability of each word for "I have yet to decide ___ the exact terms"

- 'have yet to decide on': $1.6*10^{-9}$
- 'yet to decide on the': $1.0*10^{-11}$
- 'to decide on the exact': $2.2*10^{-9}$
- 'decide on the exact terms': 0

이 경우, 'on'의 5-gram 확률 값은 $2.2*10^{-9}$로 설정된다. 위 수치는 Google n-gram 모델을 이용해 측정한 것으로, 어떤 데이터를 사용하느냐에 따라 다른 수치가 나올 것이다. 데이터가 충분치 않을 경우 'decide on the exact terms'와 같이 확률값이 0이 나올 수도 있다. 이 예제에서는 4개의 5-gram 확률 값만을 계산했지만, 만약 주어진 문장의 'terms' 뒤에 한 단어가 더 있었다면 'on'으로 시작하는 하나의 5-gram 확률 값을 더 계산했을 것이다.

보통 n 값이 클수록 확률 값이 낮아지는 것이 특징이다. 따라서 데이터가 부족하여 'decide on the exact terms'와 같이 확률값이 0이 나오거나, 또는 너무 낮은 확률값이 나올 경우에는 더 작은 n을 사용하는 것이 좋다. 'on'의 3-gram 확률 값은 다음의 3-gram 확률 중 최대치를 취한 것으로, 5-gram 확률보다 훨씬 수치가 큼을 볼 수 있다.

- 'to decide on': $1.1*10^{-6}$
- 'decide on the': $9.0*10^{-7}$
- 'on the exact': $3.3*10^{-7}$

3.2. 자동 객관식 보기 생성

본 논문에서는 객관식 문제를 내기 위해 이미 빈칸을 선정하였다고 가정한다. 또, 편의상 빈칸은 하나의 단어라고 가정한다. 한 단어에 해당하는 빈칸이 주어졌을 때, k개의 보기를 생성하는 것이 본 연구의 목표이다. 예를 들어, 'I go to school ___ bus'이라는 문장에서 빈칸에 해당하는 단어 후보인 'by', 'on', 'in', 'over' 등을 생성해야 한다. k개의 단어 후보들 중 일부는 아주 자연스러운 문장을 만들어 낼 수도 있고, 일부는 문법적으로는 맞지만 문맥상으로 오류가 있거나 잘못된 표현일 수도 있다. 우리의 목표는 출제자가 원하는 만큼 자연스러운 표현과 부자연스러운 표현을 생성할 수 있도록 하는 것이다.

본 연구의 아이디어는, 1-gram 모델의 확률값이 높은 문장보다 n-gram(단, n>1) 모델의 확률값이 높은 문장이 더 자연스러운 문장일 가능성이 높다고 판단하는 것이다. 예를 들어, 'I go to school ***on***'은 'I go to school ***by***'보다 5-gram 확률이 높다. 그렇지만 'I go to school ***by*** bus'가 'I go to school ***on*** bus'보다 6-gram 확률이 더 높기 때문에, 이 문장에서 on보다 by가 더 자연스러운 문장을 만드는 단어라 볼 수 있다. 그렇기 때문에 n-gram 확률을 비교하고, 그 다음에 (n-1)-gram 확률을 비교하고, 또 (n-2)-gram 확률을 비교하는 과정을 반복하면서 모든 단어들에 대한 순위를 매길 수 있다. 객관식 문제를 출제하기 위해서는 자연스러운 문장과 부자연스러운 문장을 동시에 보기로 제시해야 하기 때문에, 일반적으로는 높은 순위의 단어

를 하나 제시하면 낮은 순위의 단어도 제시하는 것이 바람직하다. 만약 난이도가 높은 문제를 내고 싶다면 높은 순위의 단어들만으로 구성하는 것이 좋고, 난이도가 낮은 문제를 내고 싶다면 아주 높은 순위의 단어 하나와, 아주 낮은 순위의 단어들을 다른 보기로 지정하면 될 것이다. 주의해야 할 점은, 만약 어떤 단어의 n-gram 확률이 높지 않다면 다른 단어의 n-gram 확률보다 높다고 하더라도 더 순위가 높을 것이라 단정할 수 없다는 것이다. 따라서 이때는 두 단어에 대해 (n-1)-gram 확률까지 비교해 보는 작업이 필요하다. 또 다른 문제점은, 만약 n이 아주 크게 되면 데이터의 양이 충분하지 않을 경우 확률값을 계산하지 못하는 경우가 발생하기 때문에 n을 크게 키울 수가 없다는 점이다. n을 크게 키울 수 없으면 문맥 정보를 충분히 활용하지 못하기 때문에 부정확한 결과를 낳을 수 있다. Encoder-Decoder 아키텍처와 같은 딥러닝(Deep Learning) 기술을 활용하면 이 한계를 극복할 수 있는데, 이는 추후 연구로 남긴다. 그러나 4장에서 수행한 실험 결과에 따르면 5-gram까지만 활용하더라도 어느 정도 괜찮은 결과가 나타남을 확인할 수 있다.

위 아이디어는 구체적으로 다음의 4단계 프로세스를 따라 구현된다.

1. 주어진 말뭉치를 활용하여 1-gram부터 n-gram까지 총 n개의 언어 모델을 미리 학습한다.
2. 빈칸을 채울 후보 단어 사전을 생성한다. 일반적으로 학습 말뭉치에 나타난 모든 단어들로 구성할 수 있다.
3. n'의 초기값을 n으로 설정한다. 각 n' (n >= n' >= 1)마다, 빈칸을 채울 후보 단어들의 n'-gram 확률 중 가장 높은 값을 가지는 단어 k'개를 선정한다. 만약 어떤 단어가 선정되었다면, 그 단어에 대해서는 순위가 결정되고 (n'-1)-gram의 확률 값을 계산하지 않는다. 그렇지 않은 단어들은 순위가 결정된 단어들에 비해 낮은 순위를 가지게 되며, (n'-1)-gram에 대해 동일 작업을 수행한다. n'이 1보다 작아지기 전까지 이 과정을 반복하고, 최종 순위를 결정한다.
4. 몇 순위의 단어들을 보기로 제공할 것인지를 출제자가 파라미터 값으로 입력하고, 그에 따라 최종 보기를 생성한다.

예를 들어 k'=10, n=5로 설정했다고 가정하자. 'I have yet to decide ___ the exact terms'라는 입력이 주어졌을 때, 5-gram 확률이 가장 높은 10개의 단어를 선택하고, 그 다음으로 4-gram 확률이 가장 높은 10개의 단어를 선택하고, 이 과정을 1-gram까지 반복한다. 이렇게 계산된 1위부터 25위까지의 단어 순위는 다음과 같다:

- on, whether, what, how, if, which, the, upon, to, that, between, in, where, about, of, with, and, know, are, were, for, against, at, determine, by

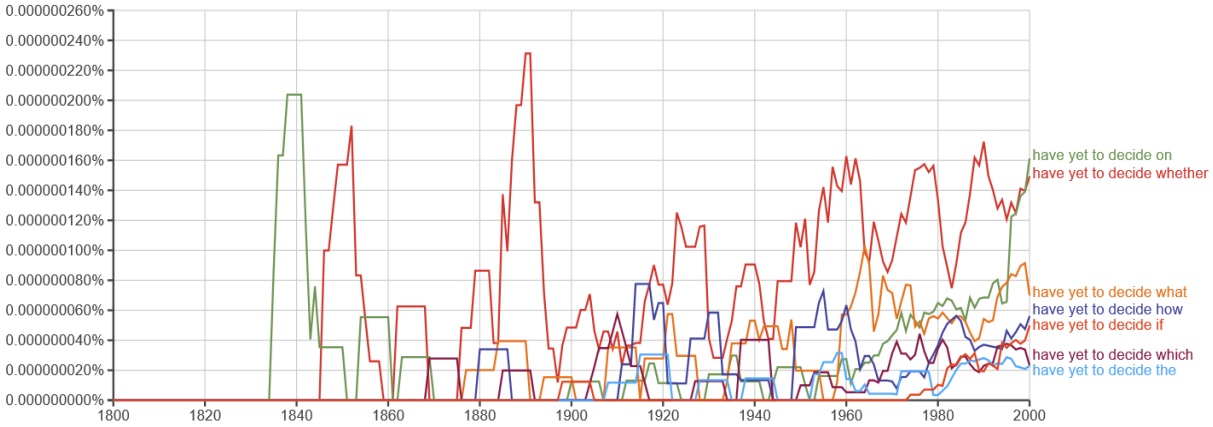
4. 실험 결과

4.1절에서는 3절의 알고리즘을 검증하기 위한 실험 환경 구축 방법을 설명한다. 4.2절에서는 사람이 직접 생성한 3절의 알고리즘이 얼마나 재현해낼 수 있는지 검증했다. 4.3절에서는 자동으로 생성된 객관식 문제를 나타내고, 그 결과를 분석했다.

4.1. 실험 환경 구축

n-gram 확률을 구하기 위해, 책을 디지털화한 데이터베이스인 Google Books로부터 생성된 n-gram 언어 모델을 활용하였다[1][4][5][11]. 이 언어 모델을 이용하면 (Fig. 3)과 같이 시대에 따른 n-gram 확률 변화를 관찰할 수도 있다. 그러나 본 실험에서는 시간대 정보를 별도 고려하지 않았으며, 2000년 데이터에 기반해 n-gram 확률을 계산했다. API를 통해서는 5-gram까지의 확률 값을 구할 수 있기 때문에 본 논문에서도 5-gram까지 사용하도록 설정했다(n=5). 그러나 API를 이용하지 않고 직접 언어 모델을 만들어 그 이상의 n-gram도 준비할 수 있다면 더 나은 성능을 기대할 수 있다. 각 n-gram마다 새로 순위를 매길 단어의 수는 최대 10개(k'=10), 학생들에게 보여줄 보기의 수는 4개(k=4)로 설정했다.

본 알고리즘은 다양한 종류의 문제에 대해 적용해볼 수 있다. 본 논문에서는 영어 시험 데이터를 테스트 데



(Fig. 3) An example visualization generated by Google n-gram Viewer

이터로 활용하여 실험하였다. 구체적으로, TEPS 홈페이지에 공개된 1월 18일/1월 25일/2월 1일 WEEKLY TEPS 문법/어휘 문제들 중 빈칸이 한 단어인 사례를 수집하여 사용하였다.

4.2. 사람이 생성한 보기와 유사성 비교

본 알고리즘은 단어들의 순위를 매기는 방법이기 때문에, 어떤 순위의 단어를 최종 선택하느냐에 따라 사람과 완전히 똑같은 결과를 만들어낼 수도, 완전히 다른 결과를 만들어낼 수도 있다. 따라서 사람이 생성한 보기와 얼마나 유사한 결과를 생성해 내는지를 평가하기는 쉽지 않다. 그렇지만 다음과 같은 방식으로 간접적으로 성능을 유추해 볼 수 있다. 먼저, TEPS 데이터에는 어떤 보기가 정답인지도 기재되어 있지만, 학생들이 어떤 보기를 많이 체크했는지도 알 수 있다. 만약 (a)가 정답이고 (c), (d), (b)의 순서로 오답을 많이 체크했다면, (a), (c), (d), (b)의 순서대로 문장 내에서 자연스러운 단어일 것으로 판단할 수 있다. 만약 3절의 알고리즘이 (a), (c), (d), (b)의 순서대로 순위를 매겼다면, 이 알고리즘이 사람과 비슷한 보기를 생성했다고 말할 수 있을 것이다.

<Table 2>는 알고리즘이 매긴 순위와 학생이 매긴 순위를 비교한 것으로, 두 순위에서 큰 상관관계가 있음을 볼 수 있다. 1번째 문장에서 학생들은 빈칸에 알맞은 단어로 on을 가장 많이 선택했고, 그 다음에는 to, in, at 순이었다. 본 알고리즘도 동일한 순서로 on, to, in, at 순으로 순위를 높게 매겼다. 계산 결과에 따르면 on은 1위, to는 9위, in은 11위, at은 21위로 나타났다. 6, 7번째 문장도 순위를 정확하게 맞추었고, 2, 4, 5번째 문장은 최소한 가장 순위가 높은 단어는 맞추었다. 3, 8번째 문제는 사람의 순위와 크게 달랐는데, 그 이유는 Google n-gram의 데이터에 'quite tidy', 'A leak formed'와 같은 데이터가 매우 부족했기 때문이다.

<Table 2> A Comparison of the Examples Generated by Human and the Computer Algorithm

Sentence	Ranked by % students selecting each item	Ranked by the algorithm
... have yet to decide ___ the ...	on(62%), to(22%), in(10%), at(6%)	on, to, in, at
... find it impossible to ___ for long ...	survive(94%), suffer(3%), outlast(2%), revive(1%)	survive, revive, suffer, outlast
... always kept her home quite ___	tidy(70%), pure(17%), sturdy(9%), patient(4%)	pure, sturdy, patient, tidy
... planned to ___ from her job as ...	resign(83%), divert(10%), remove(4%), deflect(3%)	resign, remove, divert, deflect
... unlikely that they would ___ the ...	reach(70%), arrive(22%), accomplish(6%), obtain(3%)	reach, obtain, accomplish, arrive
The medicine had a(n) ___ effect, ...	instant(87%), eventual(6%), current(6%), present(3%)	instant, eventual, current, present
... of consumers ___ our website ...	visit(98%), run(1%), spend(1%), place(0%)	visit, run, spend, place
A ___ formed in the bottom ...	leak(87%), flow(5%), drip(4%), loss(3%)	flow, drip, loss, leak

4.3. 자동 객관식 문제 생성

빈칸이 있는 TEPS 예문에 대해 3절의 알고리즘을 이용해 객관식 보기를 생성하였다(<Table 3>). <Table 2>의 예문 중 데이터가 부족하여 잘못된 결과를 만들었던 3, 8번째 문장은 제외하였다. 난이도에 따라 3가지 종류의 보기를 생성하였는데, 먼저 어려운 난이도의 보기는 다음의 기준에 따라 선정하였다:

1순위 단어는 반드시 보기에 포함시킨다. 왜냐하면 정답 단어가 최소한 하나 있어야 하기 때문이다.

남은 보기는 2위-10위 이내의 순위 단어 중에서 무작위로 선택하여 보기에 포함시킨다. 이때 1순위 단어는 아니지만 정답인(1순위 단어만큼 자연스러운) 단어가 나올 수 있다. 이 경우 사용자가 수동으로 단어를 제거하고 다시 생성할 수도 있고, 복수 정답을 인정할 수도 있다.

중간 난이도의 경우에는 위 순위 대신 11위~20위 단어를, 낮은 난이도의 경우 21~30위 단어를 사용하였다.

낮은 난이도 세팅에서 생성한 단어들은 몇 개의 주변 단어들만 참고했기 때문에 전체 문장을 보았을 때에는 어색하다는 것을 볼 수 있다. 그렇지만 낮은 순위의 단어라 할지라도 말이 되지 않는 이상한 단어는 없었다. 결국 함께 나타난 단어가 어떤 것이냐는 것에 기반하여 단어를 생성했기 때문에, 흔하지 않은 단어들을 생성해 내기는 어렵기 때문이다. 예를 들어, 사람 이름과 같은 단어들은 빈도 수가 굉장히 낮기 때문에 본 알고리즘을 통해 생성해 내기 어려울 것이다. 단, 그 사람이 유명인이고 그 사람을 칭할 때 쓰는 독특한 표현이 있다면, 보기로 생성될 수 있다.

5. 결론 및 향후 연구

본 논문에서는 빈칸이 주어졌다고 가정하고, 이에 어울리는 보기를 자동 생성하는 알고리즘을 제안하였다. 본 알고리즘은 통계 언어 모델에 기반하여 보기를 생성하기 때문에, 사람이 생성하는 경우보다 출제자에 편향되지 않은 보기를 제공할 수 있다. 또, 확률값에 기반하여 난이도를 자동으로 조절하는 것이 가능하기 때문에, 직접 사람이 문제를 만드는 것에 비해 상당한 비용 절감 효과가 있다. TEPS 문법, 어휘 시험에 대해 적용하여 실험한 결과, 사람과 유사한 결과를 생성함을 확인하였다.

향후 연구 내용은 기술/데이터/응용 관점에서 정리해 볼 수 있다. 기술적 관점으로는 첫째, 추가적인 자원을 사용하는 것을 고려할 필요가 있다. 현재 생성된 단어들은 단순히 빈도 수에 기반하기 때문에, 정확히 출제자의 의도에 맞는 단어들을 생성해냈다고 보기는 어렵다. 예를 들면 시제가 다른 단어들을 보기로 만들어 내거나, 어려운 어휘들만을 보기로 선정하고 싶을 수 있다. 또, 의미가 상반되는 다른 단어들 사이에서 원하는 답을 찾고 싶을 수 있다. 이를 위해서는 유의어/반의어/고유명사 사전 등 다양한 형태의 추가 자원이 필요하고, 그 목적에 맞게끔 알고리즘을 변형해야 할 것이다. 둘째, 더 많은 문맥을 고려할 수 있는 알고리즘을 도입할 필요가 있다. 현재 사용하고 있는 5-gram 모델은 극히 짧은 문맥만을 파악할 수 있다. 5-gram보다 더 큰 모델을 사용하면 문제가 해결되겠지만, 그러려면 관리하기 어려울 정도로 큰 양의 데이터가 필요하기 때문에 현실적으로 불가능하다. 최근 딥러닝 기술을 이용하여 적은 데이터를 이용하여 n-gram보다 더 나은 언어 모델을 만드는 방법들이 개발되고 있다. 이 방법들을 사

<Table 3> A Comparison of English Examples Generated by the Level of Difficulty

Sentence	Difficulty Levels		
	High	Medium	Low
... have yet to decide ___ the ...	on, what, to, that	on, between, where, with	on, against, at, by
... find it impossible to ___ for long ...	survive, do, think, resist	survive, account, wait, sleep	survive, ask, have, reason
... planned to ___ from her job as ...	resign, resigned, move, withdraw	resign, learn, take, obtain	resign, hear, separated, received
... unlikely that they would ___ the ...	reach, have, ever, do	reach, cancel, gained, reaching	reach, seem, give, by
The medicine had a(n) ___ effect, ...	extraordinary, immediate, strong, profound	extraordinary, indirect, beneficial, marked	extraordinary, similar, greater, to
... of consumers ___ our website ...	visit, check, see, in	visit, with, for, from	visit, who, would, all

용하면 문장의 전체 context들을 볼 수 있기 때문에, 원하는 목적에 맞는 더 좋은 결과를 생성해 낼 수 있을 것이다.

우리가 사용하는 말은 시간에 따라 경향성이 변하는데, 그 언어의 시간적 흐름을 고려하여 보기들을 생성하는 것도 고려해 볼 수 있다. 시간의 흐름뿐만 아니라, 도메인(뉴스/강연/여행 등)에 따라서도 사용되는 표현이 다를 수 있으므로 문맥에 따라 선호되는 결과가 다를 것이다. 이를 반영하기 위해서는 해당 시대/도메인의 데이터를 이용해 별개의 언어 모델을 학습하는 것이 필요하다. 결국 하나의 목적을 위해 여러 개의 언어 모델을 이용하는 것인데, 이때 발생하는 메모리 문제 등을 해결하기 위한 방법이 연구되어야 한다.

빈칸 채우기 문제뿐만 아니라 다른 방식의 교육 문제에도 본 알고리즘을 변형하여 적용할 수 있는지 검토할 필요가 있다. 본 논문에서 유사 보기들을 생성했던 것처럼, 같은 방식으로 유사 텍스트를 만들어낼 수도 있다. 예를 들어 컴퓨터 수업 시간에 배웠던 프로그램 코드와 유사한 프로그램을 시험 문제로 내고 싶다면, 이 방식을 도입할 수 있을 것이다. 또, 영어 공부의 목적으로 하나의 한글 문장에 대해 다양한 영어 번역문을 보고 싶을 수도 있다. 예를 들면 '잘 지내십니까'라는 문장에 대응되는 영어 문장의 개수는 많은데, 그런 표현들을 모두 배우고 싶을 수 있다. 그럴 경우 입력으로 주어지는 한글 문장과 유사한 한글 문장들을 본 알고리즘을 통해 만들어 내면, 그 이후 각각에 대해 번역 알고리즘을 수행함으로써 한 문장에 대한 여러 개의 유사한 번역문들을 얻을 수 있을 것이다.

실제 초등 교육에서 활용될 경우, 앱 형태 또는 퀴즈 시험의 형태가 가장 일반적이다. 앱으로 구현될 경우, 초등 교과서에 나오는 주요 개념들을 빈칸으로 표기하고 보기를 제공함으로써 학습한 내용을 자가 테스트하는 용도로 활용될 수 있다. 퀴즈 시험으로 출제된다면 파워포인트를 이용하여 빈칸에 알맞은 보기를 제시할 수도 있고, 쪽지 시험의 형태로 학생들에게 나누어줄 수도 있다. 초등 영어 교육 및 저학년 국어 교육의 경우 본 연구를 바로 적용하여 활용할 수 있지만 타 교과목의 경우 별도의 데이터를 수집하여 언어 모델을 학습해야 할 수 있다. 실 사례에 적용하고 그 효과를 검증하는 것은 향후 연구로 남긴다.

참고문헌

[1] Acerbi, A., Lampos, V., Garnett P. & Bentley RA. (2013). The Expression of Emotions in 20th Century Books. *PLoS ONE, 8*(3).

[2] Alsubait, T., Parsia, B. & Sattler, U. Generating Multiple Choice Questions From Ontologies: Lessons Learnt. Proceedings of the 11th International Workshop on OWL: Experiences and Directions, 73-84.

[3] Bae, Y. & Do, J. (2013). Study on Smart Learning Contents Development using Storyline. *Journal of the Korean Association of Computer Education, 17*(2), 135-146.

[4] Google n-gram viewer. https://books.google.com/ngrams

[5] Google n-gram viewer (Wikipedia). https://en.wikipedia.org/wiki/Google_Ngram_Viewer

[6] Hoshino, A. & Nakagawa, H. (2005). A Real-time Multiple-choice Question Generation for Language Testing: a Preliminary Study. Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, 17-20.

[7] Kim, S. & Kim, K. (2012). Design and Implementation learning English words Smart-phone application for Elementary school students on Android platform by Focus on form. *Journal of the Korean Association of Computer Education, 16*(2), 223-231.

[8] Lee, S. & Ryu, H. (2013). Suggestion on the Key Factors of Smart Education. *Journal of the Korean Association of Computer Education, 17*(2), 101-113.

[9] Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B. & Karger D. R. (2003). The role of context in question answering systems. Proceedings of the Extened Abstracts on Human Factors in Computing Systems, 1006-1007.

[10] Pintos. http://pintos-os.org/

[11] Roth, S. (2014). Fashionable functions. A Google ngram view of trends in functional differentiation

(1800-2000). *International Journal of Technology and Human Interaction, 10*(2), 34-58.

[12] Soul M. & Son. C. (2012). A Survey on Teacher's Perceptions about the Current State of Using Smart Learning in Elementary Schools. *Journal of the Korean Association of Computer Education, 16*(3), 309-318.

[13] Sumita, E., Sugaya F. & Yamamoto, S. (2005). Measuring non-native speakers' proficiency of English by using a test with automatically-generated fill-in-the-blank questions. Proceedings of the second workshop on Building Educational Applications Using NLP, 61-68.

[14] Thompson, A. S. (2015). Are Your Participants Multilingual? The Role of Self-assessment in SLA Research. *Language in Focus, 1*(1), 51-65.

[15] Wood, C. L., Mustian A. L. & Cooke N. L. (2010). Comparing Whole-Word and Morphograph Instruction During Computer-Assisted Peer Tutoring on Students' Acquisition and Generalization of Vocabulary. *Remedial and Special Education, 33*(1), 39-47.

저자 소개

박 영 기

2008 KAIST 전산학전공(학사)

2010 서울대학교 컴퓨터공학부 (석사)

2015 서울대학교 컴퓨터공학부 (박사)

2015~현재 삼성전자 전문연구원

관심분야: 컴퓨터 교육, 기계 학습, 데이터 마이닝, 소프트웨어 공학

e-mail: ypark@europa.snu.ac.kr