특집

통계적 기계 번역 기술의 연구 동향

김 선 호* 윤 준 태** 임 해 창***

목 차

1. 서 론

기계번역은 자연어 처리 및 인공지능 분야에서 가장 어려운 태스크 중의 하나로 인식되어왔다. 이는 정확한 번역이란 텍스트에 대한 이해 없이는 불가능하기 때문이다. 그러한 이유로 연구자들은 한때 기계번역에 대한 부정적인 결론에 도달하기도 하였다.

지금까지 기계번역을 위해 다양한 방법이 연구되어 왔으며 이들 연구에서는 주로 두 언어에 대한 어휘나 구의 대역사전, 숙어사전, 개별 언어의 문법, 혹은 변환규칙 및 변환사전, 문장생성에 관련된 지식, 의미나 실세계 지식, 도메인에 적합한 지식 등 번역의 방식과 목적에 따라 다양한 형태의 지식과 알고리즘이 적용되었으며 그 대부분은 방대한 양의 수작업에 의존적이었다.

방대한 수작업과 언어처리 모듈을 필요로 하는 기존 시스템의 한 대안으로 IBM TJ Watson Research Center에서 통계적 기계번역(SMT)의 실험적인 연구를 발표하였다. Candide 시스템은 SMT 연구의 시초가 되는 시스템으로 기계번역을 정보 이론적 관점 즉, 잡음 채널 모델 측면에서 재해석하고 Hansard라고 하는 방대한 양의 영-불 2개국어 문서 집합으로부터 영어와 불어가 어떻게 관련되는지의 번역 과정을 자동적으로 학습하였다.

통계적 모델의 발전 가능성은 통계적 방법에 기반을 둔 음성 인식 시스템의 정확성 증가를 통해 살펴볼 수 있다. 음성 처리 연구가들은 이러한 성공의 원인을 통계적 말뭉치 기반 방법, 컴퓨터의 빨라진 계산 능력, 공유된 음성처리 툴킷, 더 발전된 음성과 언어모델 등 방대한 양의 학습 데이터, 공유 자료 및 평가 자료의 확보로 분석하고 있다.

그러나 SMT분야에서 이러한 일들은 이제 시작 단계지만 비교적 인상적인 초기결과를 내놓는 정도의 수준에 있어 앞으로 보다 나은 성능을 낼 수 있을 것으로 기대하고 있다. 현재까지의 SMT는 번역된 2개언어 말뭉치를 사용하여, 적어도 영어-불어와 같이 사용가능한 리소스가 풍부한 언어쌍에 대해서는 기존의 상용화된 MT 시스템의 성능을 능가할 수 있다는 다소 희망적인 결과를 보여주고 있다.

* 고려대학교 자연어처리연구실 연구원
** Daumsoft, 자연어처리 연구소 소장
*** ACM Transaction on Asian Language Information Processing Associate Editor

그럼에도 불구하고 다음과 같은 문제점 때문에 통계적인 방법이 아직 널리 적용되지 못하고 있으며 국내의 연구는 거의 전무하다고 할 수 있다. 첫째는 SMT 기법은 수학적으로 복잡한 모델과 학습에 상당한 계산량이 필요하고, 둘째 영어-불어를 제외하면 학습을 위한 충분한 양의 2개국어 텍스트가 부족하며 주로 그 텍스트의 도메인이 제한적이라는 점에 있다. 셋째는 언어처리에 있어 구문적 처리의 중요성에도 불구하고 이의 반영이 제대로 이루어지지 않고 있다는 점이다. 예컨대, SMT 모델에서의 구문적 처리의 역할이 아직까지는 명확하지 않아 이질적 언어에의 적용에 회의적 시각이 많다. 또한 번역 정확률 면에서 개선해야 할 사항도 많이 존재하고 있다.

이러한 몇몇 문제점에도 불구하고 SMT는 현재 기계번역에서 가장 주목받고 있는 분야의 하나다. 이에 본 논문은 IBM 모델을 중심으로 SMT의 개요를 설명하고 연구의 동향과 그 문제점 등에 대해 개략적으로 소개하고자 한다.

2. 통계적 기계번역의 개요

SMT의 그 이론적 근간은 정보 이론(information theory)에 있다. 본 장에서는 정보 이론의 잡음 채널 모델과 통계적 기계 번역이 잡음 채널 모델 하에서 어떻게 모델링되는지를 살펴본다. 또한 채널 오퍼레이션으로서의 번역은 확률적으로 결정되는데, 이 확률값 추정을 위해 SMT 전반에 사용되고 있는 학습 방법인 EM(expectation maximi- ztion) 알고리즘과 그 수학적 전개를 IBM 모델을 바탕으로 알아본다.

2.1 잡음 채널 모델

정보 이론은 잡음이 존재하는 채널 하에서 메시지의 전송 혹은 통신을 처리량과 정확성이라는 측면에서 최적화하기 위해 연구되었다.

한영 번역을 이러한 잡음 채널 모델하에서 살펴보면, 화자는 영어를 통해 말을 전달하려고 하나 잡음 채널을 통과하면서 실제 나타난 텍스트에서는 한국어로 나타나게 된다고 가정한다. 따라서 우리는 출력된 결과인 한국어로부터 입력인 영어를 예측해야 하며 이러한 잡음 채널에 의한 변환은 확률적으로 결정된다. 통계적 기계 번역은 이와 같이 관찰된 출력으로부터 입력을 예측하기 위한 확률적 모델링을 병렬 말뭉치를 기반으로 하게 된다. 채널 모델을 한-영 번역 모델에 응용하여 수식화하면 다음과 같다.

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{k}) = \arg\max_{\mathbf{e}} \frac{p(\mathbf{e})p(\mathbf{k}|\mathbf{e})}{p(\mathbf{k})} = \arg\max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{k}|\mathbf{e}) \quad \text{식 1)}$$

번역 시스템은 한국어 문장 스트링 k가 주어졌을 때 확률값 p(e|k)를 최대로 하는 영어 스트링 $\hat{\mathbf{e}}$를 찾아준다. 위의 식에서 한국어 문장 k는 주어지는 항이므로 주어진 식은 분자를 최대화시키는 e를 찾는 문제가 된다. 이러한 정의 하에서 통계적 기계 번역에 관한 위의 식은 다음의 세 가지 문제로 나누어 질 수 있다.

1) 번역 확률 p(k|e)를 예측하기 위한 번역 모델링 문제
2) 언어 확률 p(e)를 추정하기 위한 언어 모델링 문제
3) (1)과 (2)의 곱을 최대화 시키는 영어 문장 e를 찾는 디코딩 문제

이중에서 2)의 언어 모델링 부분은 주로 통계기반 자연어처리에서 다루어지고 있으며, 기계번역 분야에서 관심있게 다루어지는 부분은 1)과 3), 특히 1)의 번역 모델이다. 따라서 본 논문에서 전반적으로 다루고자 하는 부분은 번역 모델이다.

2.2 IBM 모델1

통계적 기계 번역에서는 특정 영어 문장이 주어

졌을 때, 어떠한 한국어 문장이 나타날 가능성이 가장 큰가를 추정하는 것이다. 그러나 아무리 큰 말뭉치를 사용한다 할지라도 문장 대 문장의 번역 확률 p(k|e)를 직접 구할 수는 없다. 따라서 이 수식은 좀더 간단한 형태로 표현되어야 하며, 이를 위해 어휘간 정렬 파라미터 a를 도입한다. 그러나 일반적으로 말뭉치에 정렬 정보가 나타나지 않으므로 번역 확률을 직접 구할 수는 없으며, 단지 모든 가능한 정렬 A(e,k)의 합으로서 다음과 같은 기대값에 의한 계산이 가능할 뿐이다.

$$p(\mathbf{k}\mid\mathbf{e})=\sum_{\mathbf{a}} p(\mathbf{k},\mathbf{a}\mid\mathbf{e}) \qquad \text{식 2)}$$

여기서 정렬이란 어떤 영어의 단어(들)가 한국어의 어떤 단어(들)과 연결되는지에 대한 정보다. 식 2)의 오른쪽 확률 p(k,a|e)는 영어 문장 e가 특정 정렬의 형태 a를 띠고 한국어 문장 k로 번역될 확률로서, 사실상 다양한 형태로 표현가능한데, IBM 모델에서는 그 확률이 다음과 같이 정의되었다. 여기서 a_j는 j번째 불어 단어와 정렬되는 영어 단어의 위치, m은 불어 문장의 길이, l은 영어 문장의 길이를 각각 의미한다.

$$p(\mathbf{f},\mathbf{a}\mid\mathbf{e})=p(m\mid\mathbf{e})\prod_{j=1}^{m} p(a_j\mid a_1^{j-1},f_1^{j-1},m,\mathbf{e})p(f_j\mid a_1^{j},f_1^{j-1},m,\mathbf{e}) \qquad \text{식 3)}$$

식 3)은 주어진 영어 문장에 대응될 불어 문장의 길이를 결정하고 영어 문장과 대응 불어 문장의 길이가 결정되었을 때, 첫번째 불어 단어와 정렬될 영어 단어의 위치를 결정하고 영어 문장과 불어 문장의 길이, 첫 번째 불어 단어와 정렬될 영어 단어의 위치가 결정되었을 때 해당 첫번째 불어 단어를 생성하는 등의 방식으로 영어문장에 대한 불어문장이 생성되는 체인 룰이다. 그러나 이 수식 역시 고려해야 할 파라미터 수가 너무 많기 때문에 적절한 독립가정을 적용하게 되는데, 이로부터 IBM 모델 1~5의 전개가 이루어진다.

IBM 모델 1에서는 확률 p(m|e)를 상수값 ε으로 두고, 확률 $p(a_j\mid a_1^{j-1},f_1^{j-1},m,e)$는 영어 문장의 단어의 수 l에만 의존하고 $p(a_j\mid a_1^{j},f_1^{j-1},m,e)$는 fj에 연결될 영어 단어에만 의존한다고 가정함으로써 모델을 단순화시키게 된다. 그러면, 다음과 같은 근사화된 식을 얻을 수 있으며 결국 정렬은 주어진 영어 단어에 대한 불어 단어의 발생 확률인 번역 확률 t 에 의해 구해진다.

$$p(\mathbf{f},\mathbf{a}\mid\mathbf{e})=\frac{\varepsilon}{(l+1)^m}\prod_{j=1}^{m} t(f_j\mid e_{a_j}) \qquad \text{식 4)}$$

이때 주어진 영어 문장에 대한 불어 문장의 번역 확률 p(f|e)는 다음과 같이 모든 가능한 정렬의 합 즉, p(f,a|e)에 대한 기대값으로서 표현 가능하다. 여기에서, 식 4)에 상수를 무시하면 다음과 같은 확률식으로 나타낼 수 있다.

$$p(\mathbf{f}\mid\mathbf{e}) \;=\; \sum_{\mathbf{a}} p(\mathbf{f},\mathbf{a}\mid\mathbf{e})=\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\prod_{j=1}^{m} t(f_j\mid e_{a_j}) \qquad \text{식 5)}$$

식 5)는 영어 문장 e가 불어 문장 f로 번역될 확률이란 개별 영어 단어 eaj가 불어 단어 fj로 번역될 확률들의 곱으로서 표현됨을 의미한다. 이러한 경우 순서에 대한 고려가 없으므로 언어 모델 p(e)에 의해 가장 그럴 듯한 문장을 찾게 된다.

IBM 모델의 전반적인 정렬의 제약조건은 불어 문장 f의 개별 단어는 단 하나의 영어 단어 e에만 대응됨을 가정하였다.

2.3 EM을 이용한 파라미터 학습

IBM 모델 1에서 우리의 관심은 결국 2개 언어 말뭉치로부터 전체 확률 p(f|e)를 최대화시키는 각각의 번역 확률 t를 구하는 것이며 본 절에서는 이

t값을 계산하는 방법에 대해 알아본다. 주어진 문제에는 부가되는 제약조건이 주어지는데 그것은 각 영어 단어 e의 번역어 f에 대한 확률의 합은 1이 되어야 한다는 것이다. 따라서, 이는 제약조건이 주어지는 최적화 문제로 볼 수 있는데, 이러한 최적화 문제를 풀기 위한 일반적 방법으로 라그랑지 곱수(Lagrange multiplier)를 도입하면 최대화시켜야 할 부가함수 h는 다음과 같이 나타낼 수 있다.

$$h(t,\lambda) = \sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\prod_{j=1}^{m}t(f_j|e_{a_j}) - \sum_e \lambda_e\left(\sum_f t(f|e)-1\right)$$

식 6)

h를 최대화시키는 t(f|e)는 주어진 식을 특정 t(f|e)에 대해 각각 편미분함으로써 구할 수 있다. 다음은 각 t에 대한 h의 일반화된 편미분을 표현한 식이다.

$$\frac{\partial h}{\partial t(f|e)} = \sum_{a_1=1}^{l}\cdots\sum_{a_m=1}^{l}\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})c(f|e)t(f|e)^{-1}\prod_{k=1}^{m}t(f_k|e_{a_k}) - \lambda_e$$

$$t(f|e) = \lambda_e^{-1}\sum_{a_1=1}^{l}\cdots\sum_{a_m=1}^{l}\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})\prod_{k=1}^{m}t(f_k|e_{a_k})$$

$$t(f|e) = \lambda_e^{-1}\sum_{\mathbf{a}}P(\mathbf{f},\mathbf{a}|\mathbf{e})\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})$$

식 7)

그러나, 위 식에서는 t(f|e)가 식의 좌우변에 모두 나타나기 때문에 직접 계산이 불가능하다. 따라서 다음과 같이 카운트 c를 정의함으로써 반복적 연산에 의한 계산이 가능하도록 한다. 이때, 카운트 c는 단어 e가 단어 f와 정렬될 예상 횟수 혹은 기대 횟수에 해당된다.

$$c(f|e;\mathbf{f},\mathbf{e}) = \sum_{\mathbf{a}}p(\mathbf{a}|\mathbf{f},\mathbf{e})\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})$$

$$= \frac{\sum_{\mathbf{a}}p(\mathbf{a},\mathbf{f}|\mathbf{e})\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})}{p(\mathbf{f}|\mathbf{e})\left(\sum_{\mathbf{a}}p(\mathbf{f},\mathbf{a}|\mathbf{e})\right)}$$

$$= \frac{\sum_{\mathbf{a}}p(\mathbf{a},\mathbf{f}|\mathbf{e})\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})}{\sum_{a_1=1}^{l}\cdots\sum_{a_m=1}^{l}\prod_{j=1}^{m}t(f_j|e_{a_j})}$$

식 8)

여기에서 $\lambda_e p(f|e)$를 λ_e로 바꾸면 식 4)에 의해서 식 9)를 얻을 수 있고 이를 전체 문장에 대해서 표현하면 결국 단어간 번역 확률 t(f|e)는 식 10)과 같이 표현된다.

$$t(f|e) = \lambda_e^{-1}c(f|e;\mathbf{f},\mathbf{e})$$

식 9)

$$t(f|e) = \lambda_e^{-1}\sum_{s=1}^{S}c(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)})$$

$$\lambda_e = \sum_f\sum_{s=1}^{S}c(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)})$$

식 10)

그러나 우리가 계산해야 할 식 5)와 8)은 모든 가능한 정렬에 대한 합과 그에 대한 곱을 포함하고 있기 때문에 c값을 구하기 위해 주어진 식을 그대로 이용한다면 계산상의 복잡도가 지나치게 커진다. 다행히도 모델 1에서 주어진 식 5)는 계산이 용이한 다른 형태의 수식으로 변형이 가능하다. 즉, 곱의 합 형태의 식 5)는 그와 동일한 합의 곱 형태인 식 11)로 대신함으로써 현실적으로 계산 가능한 형태를 만들 수 있다.

$$p(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}}p(\mathbf{f},\mathbf{a}|\mathbf{e}) = \prod_{j=1}^{m}\sum_{i=1}^{l}t(f_j|e_i)$$

식 11)

그러면, 식 8)의 카운트 값 c는 다시 식 12)와 같이 표현할 수 있다.

$$c(f|e;\mathbf{f},\mathbf{e}) = \frac{\sum_{\mathbf{a}}p(\mathbf{a},\mathbf{f}|\mathbf{e})\sum_{j=1}^{m}\delta(f,f_j)\delta(e,e_{a_j})}{\prod_{j=1}^{m}\sum_{i=1}^{l}t(f_j|e_i)}$$

식 12)

위의 식을 이용하면, 식 8)은 보다 계산하기 쉬운 다음과 같은 식으로 변형된다.

$$c(f|e;\mathbf{f},\mathbf{e}) = \frac{t(f|e)}{t(f|e_1)+\cdots+t(f|e_l)}\sum_{j=1}^{m}\delta(f,f_j)\sum_{i=1}^{l}\delta(e,e_i)$$

식 13)

또한 이를 이용한 파라미터 t에 대한 학습 알고리즘은 다음과 같다.

파라미터 학습 알고리즘

1. $t(f \mid e)$를 위한 초기값들을 e를 기준으로 균일하게 배정한다.
2. 식 14)를 사용해 각 문장 쌍 ($f^{(s)}$, $e^{(s)}$)에 대해 c(f|e; $f^{(s)}$,$e^{(s)}$)를 계산한다.
3. $\mathbf{e}^{(s)}$ 상에 적어도 한번은 나타난 각각의 e에 대해 λ_e를 계산하고 $\mathbf{f}^{(s)}$ 상에 적어도 한번 이상 나타나는 각각의 f에 대해 식 11)을 이용하여 새로운 $t(f \mid e)$ 값을 구한다.

$$\lambda_e = \sum_f \sum_{s=1}^{S} c(f \mid e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})$$

4. $t(f \mid e)$가 특정 범위 내로 값이 수렴할 때까지 과정 2와 3을 반복한다.

이와 같이 불완전한 관찰 데이터에 대해 기댓값이 최대가 되도록 점진적 반복 계산에 의해 파라미터의 최우추정을 하는 방법을 EM 알고리즘이라 한다. 여기서 불완전한 데이터란, 어떤 데이터의 실제 파라미터 분포가 알려져 있지 않고 숨겨져 있음을 의미한다(Dempster, Laird and Rubin, 1976).

병렬 말뭉치의 어휘 정렬의 경우, 우선 영어 문장이 주어졌을 때 대응되는 한국어 문장의 기대값을 구하며, 이는 가능한 정렬의 연결 확률의 합으로 정의될 수 있다. 이를 E-Step(estimation step)이라 한다. 또한, M-Step(maximization step)에서는 각 파라미터의 최우추정을 하게 되는데 각 E-Step에서 주어진 파라미터 값과 기대값을 기반으로 하여 나타난 정렬의 횟수를 재계산함으로써 가능하다. 직관적 이해를 위해 영어-한국어 병렬 말뭉치를 구성하는 문장쌍이 〈"b c"-"x y"〉, 〈"b"-"y"〉의 두개로 주어진다고 가정해 보자. 여기에 어휘간 대응 관계를 만든다면 어떤 것이 가장 그럴 듯 할까? 관찰된 사실만으로 추정해본다면 가능한 번역 쌍은 {(b, y), (c, x)}가 될 것이다. EM에서는 부트스트래핑 개념을 이용하여, (b, y)와 같은 특정 대응 관계가 다른 관계보다 많이 나타나는 특성이 점진적으로 반영됨으로써 최우추정을 하게 된다.

EM의 방법론은 비록 국부적인 해를 얻기 쉽다는 단점이 있으나, SMT 시스템의 정렬 파라미터 예측을 비롯하여 HMM 모델의 파라미터 예측, finite mixture model의 파라미터 예측, 클러스터의 비교사 학습 등 레이블이 주어지지 않는 자연어 처리의 여러 불완전 데이터 문제에 적용되어 왔다.

2.4 IBM 모델2-5

IBM 모델 2에서는 j번째 불어 단어와 매핑될 영어 단어의 위치 aj를 위한 확률 $p(a_j \mid a_1^{j-1}, f_1^{j-1}, m, e)$를 구할 때 모델 1과는 달리 두 문장의 길이인 m과 l뿐만 아니라 불어 단어의 위치 j에 영향을 받는다고 하는 가정이 덧붙여진다. 즉, aj번째 영어 단어가 j번째 불어 단어와 정렬될 확률은 불어 단어의 위치 및 영어 문장의 길이에 의해 조건화된다고 가정하여 p(aj|j,m,l)로 근사화시켰다.

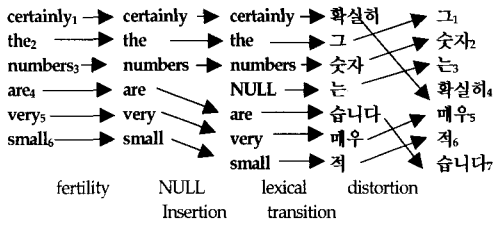
$$p(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j \mid e_{a_j}) a(a_j \mid j, m, l)$$

식 14)

따라서, 모델 2는 두 개의 파라미터, 번역 확률 t와 정렬 확률 a가 사용된다. 정렬 확률은 특정 위치의 불어 단어가 영어 문장의 어떤 위치와 잘 대응이 되는지를 반영하는 값이다.

모델 3, 4, 5는 해석을 달리하여 개념적으로 4가지 채널 오퍼레이션 즉, 확률 파라미터에 의해 대응 문장이 생성된다. 즉, 1) 단어 복사와 삭제를 위한 fertility 확률, 2) NULL 삽입을 위한 확률, 3) 위치의 재순서화를 위한 distortion 확률, 4) 단어 번역을 위한 translation 확률이 고려된다. 번역의 기본 전략은 각 원시 문장의 단어에 대해 목적 문

장의 단어(들)을 생성하고 그들을 재순서화 한다. 이를 도식화하면 (그림 1)과 같다.



(그림 1) 문장 번역 과정

또한 다음은 IBM 모델 3에 대한 수식이다.

$$p(\mathbf{f}\mid\mathbf{e}) = \sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\binom{m-\phi_0}{\phi_0}p_0^{m-2\phi_0}p_1^{\phi_0}\prod_{i=1}^{l}\phi_i!\,n(\phi_i\mid e_i)\times \prod_{j=1}^{m}t(f_j\mid e_{a_j})d(j\mid a_j,m,l) \quad \text{식 15)}$$

모델 3에서는 각 영어 단어에 대해 정렬되는 불어 단어의 개수를 고려하기 위해 fertility 파라미터 ϕ_i와 그 발생 확률인 n을 이용한다. 이때 ϕ_i가 0이 되는 경우는 단어의 삭제를 의미한다. 또한 특정 영어 위치와 정렬될 불어 단어의 위치와 관련된 distortion 확률 d, 그리고 해당 불어 단어로 번역될 translation 확률 t가 이용된다. 모델 3에서는 개별 영어 단어가 특정 개수의 불어 단어를 생성하는 fertility 확률이 이용되고 있으므로, 영어 단어와 결합되지 않는 불어 단어를 위해 가상적으로 설정된 영어의 NULL 단어의 fertility도 고려되어야 한다. 이를 위해, 문장이 길수록 부가적인 단어들이 많이 생긴다고 가정하였으며, 개별 불어 단어들이 잉여 불어 단어를 p1의 확률로 생산하는 다음과 같은 이항분포를 정의하였다.

$$p(\phi_0\mid\phi_1^l,e)=\binom{\phi_1+\ldots.+\phi_l}{\phi_0}p_0^{\phi_1+\ldots.+\phi_l-\phi_0}p_1^{\phi_0} \quad \text{식 16)}$$

한편, 모델 4는 단어들이 하나의 단위로 움직이는 구의 속성을 반영하기 위해 위치 결정 확률인 d를 다음 수식과 같이 변형하여 생각하였다.

$$p(\mathbf{f}\mid\mathbf{e}) = \sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\binom{m-\phi_0}{\phi_0}p_0^{m-2\phi_0}p_1^{\phi_0}\prod_{i=1}^{l}\phi_i!\,n(\phi_i\mid e_i)\times \prod_{j=1}^{m}t(f_j\mid e_{a_j})d_1(j-\odot_{i-1}\mid \mathbf{A}(e_{[i-1]}),\mathbf{B}(f_j))d_{>1}(j-\pi_{[i]k-1}\mid \mathbf{B}(f_j)) \quad \text{식 17)}$$

모델 4에서는 d 확률이 d_1과 $d_{>1}$로 나뉘어져 있다는 점이 모델 3과 다르다. 위의 식에서 A와 B는 클래스 함수에 해당되며 이를 위해 영어, 불어 단어에 50개의 클래스가 존재한다. d_1은 머리어를 위한 위치 변형 확률 $d_{>1}$는 나머지 단어를 위한 위치 변형 확률이다. 머리어는 불어 문장에서 가장 앞쪽 위치에 나타난 단어를 머리어로 보았다. $\odot_i$는 i번째 영어 단어가 생성해 내는 불어 단어들의 위치에 대한 평균(center)에 해당하며, $\pi_{[i]k-1}$는 i번째 영어 단어가 생성해 내는 불어 단어 리스트 중 k-1번째 불어 단어의 문장 내의 위치를 의미한다. 따라서 위치 변형 확률의 의미를 해석해 보면 현재 i번째 영어 단어가 j번째 머리어 불어 단어와 연결될 확률은 j번째 위치와 i-1번째 영어 단어가 생성해 낸 불어 단어들의 위치 중심과의 차이와 현 불어 단어의 클래스, 이전 영어 단어의 클래스에 의존한다고 보았다. 머리어가 아닌 경우에는 현재 리스트의 k번째 불어 단어의 실제 문장상의 위치를 i번째 단어가 생성해 내는 불어 리스트 중 이전 단어 k-1번째 불어 단어의 문장의 위치와 뺀 차이와 현재 불어 단어의 클래스에 의존한다고 보았다.

또한, 모델 5에서는 더 개선된 위치 변형 확률을 제안하고 있다.

다시 학습의 문제로 돌아가서 IBM 모델 2-5도 각 확률 파라미터를 병렬 말뭉치를 이용하여 추정하기 위해서 앞에서 설명한 대로 카운트 c를 정의하고 EM을 사용해 진행하여 나간다. 그러나 유의할 점은 모델 3부터는 모델 1과 모델 2에 적용할 수

있는 식 11)과 같은 계산상의 편의를 이용할 수 없으므로 카운트 c값을 모든 가능한 정렬에 대해 구한다는 것은 사실상 불가능하다. 따라서 더 가능성 있는 한 정렬들만을 대상으로 카운트 값을 구하는 방법이 사용된다. 이를 위해 IBM모델에서는 neighbor를 구해 pegged Viterbi 정렬을 시도하였다. 또, Knight (1999)는 모든 가능한 정렬을 다 고려한 것과 동일한 결과를 내는 계산상으로 훨씬 간단한 알고리즘을 제시하였으며, Yamada (2001)는 Inside-outside 알고리즘을 이용하여 학습의 효율성을 높이는 등, 학습의 효율화가 주요 문제로 다루어졌다.

3. IBM 모델의 문제점

앞 장에서는 SMT 연구의 수학적 기반이 되고 있는 IBM 모델 1~5와 EM 학습 방법에 대해서 간략하게 살펴보았다. 그러나 IBM 모델은 기본적으로 단어 대 단어 모델로서, 영어 대 불어에 대해 1:n 대응의 기본틀을 벗어나지 못하는 표현력의 한계를 가지고 있다. 실제로 단어대 단어 번역의 경우 어색한 언어 표현들이 많이 발생한다. 예를 들어, 숙어 표현 및 하나 이상의 단어들로 번역되어야 하는 복합어, 원거리 의존 관계가 존재하는 구나 문맥에 따라 다른 의미를 지니는 중의적 단어들의 경우는 구의 개념도 고려할 수 있는 정렬 모델 하에서 해석되어야 한다. 특히 영어-한국어 번역에서의 일대일 대응은 동종언어간의 번역에 비해 훨씬 낮은 수치를 보이는 것으로 드러난다.

또한 언어학적 지식이 모델의 파라미터로 거의 사용되고 있지 않기 때문에 복잡한 단어간의 대응 관계를 포착하기에는 부족한 모델 구조를 가지고 있다. 예를 들어, Wang (1998)에 따르면 IBM 모델 2의 경우 정렬 파라미터들의 수가 번역 파라미터들의 수보다 훨씬 작기 때문에 정렬 파라미터들이 상대적으로 과학습되어 결과적으로 어떤 단어가 등장했는지와 상관없이 비슷한 위치의 단어들끼리 정렬이 이루어지게 되는 문제점을 가지고 있음을 지적하기도 했다.

4. 그외의 연구

IBM 모델의 단점을 극복하기 위해 단어 대 단어 간 정렬이라는 기본 틀 안에서도 여러 가지 방법들이 시도되었다. Vogel(1996)의 연구에서는 원시 문장의 단어 위치와 번역된 문장에서 그 대응 위치에 대한 정보가 정렬 시 유리하지만 이 때 대응 위치 정보는 단어의 절대적 위치에 의존하는 것이 아니라 상대적 위치에 의존함을 파악하고 정렬에 있어서 위치 상의 지역성이 유지되는 점을 이용하였다. 또한, aj의 확률은 aj-1에 의존적이라는 마코프 가정을 적용함으로써 IBM 모델 2를 수정하였다. 수정된 정렬 확률 a는 다음의 식 18)에 의해 표현될 수 있다.

$$p(\mathbf{f}\mid\mathbf{e}) = \max_{a_i^j} \prod_{j=1}^{J} \mathbf{a}(a_j \mid a_{j-1}, I)\, p(f_j \mid e_{a_j})$$

$$\mathbf{a}(a_j = i \mid a_{j-1} = i', I) = \frac{s(i-i')}{\sum_{l=1}^{I} s(l-i')} \qquad \text{식 18)}$$

앞에서 언급한 대로 IBM 모델은 단어 간 독립 가정을 사용하고 구 대 구 대응을 고려하지 않기 때문에 문맥이 반영이 되지 않는다. 그러나 사실상 단어는 문맥에 따라 다르게 번역될 수 있다. Berger (1996)는 최대 엔트로피 모델을 사용하여 문맥이 반영되는 모델을 제안하였는데, 이 모델에서는 p(f|e) 대신 pe(y|x) 식이 정의되었다. 여기서 x는 영어 단어의 문맥, y는 x 문맥 하에서 영어 단어 e의 불어 번역이 y임을 의미한다. $p_e(y|x)$의 확률값을 구하기 위해 영어 단어 e가 주어졌을 때 주변 문맥 x가 나타나고 그 때 번역이 y가 되면 그 값이 1이 되는 이진 자질 함수 $f_e(x,y)$를 도입하여 p(f|e)를 구하는 문맥 의존적 모델을 제안하였다.

Nieben (2000)은 단어 대 단어 번역으로 설명할 수 없는 요소들을 위해 전처리 단계를 두어 입력 텍스트를 변형시켰다. 예를 들어, 독일어에서 특정 동사들은 중심이 되는 어간과 문장에서 분리될 수 있는 접두사(detachable prefix)를 가지고 있어 이들이 넓은 범위를 두고 떨어져 나타날 수 있다. 이 연구에서는 이러한 동사들은 '접두사+어간'의 형태로 붙여 전처리 단계에서 변형을 시킨다. 또한 복합어들의 경우는 두 단어로 쪼개고 문장에서 특별한 구문적 역할을 담당해 통째로 하나로 볼 수 있는 여러 단어로 구성된 구(multi-word phrase)들을 하나로 합쳐 단어 대 단어의 대응을 유지하면서 단어 대 단어 기반 모델의 단점을 극복하였다.

이 밖에도 단어 대 단어 정렬의 한계점을 극복하기 위해서 구나 템플릿, 청크, 구조적 정보를 고려한 모델들이 논의되었다. 구 단위 정보를 고려하게 되면 다대다 대응관계를 포착할 수 있고 지역 문맥을 반영할 수 있으며 숙어와 같이 비복합적 구(non-compositional phrase)들을 설명할 수 있게 된다.

Och and Ney (2000)와 Koehn(2003)의 경우는 휴리스틱을 사용하여 단어대 단어 정렬로부터 간단한 구 단위 정보를 유추하여 정렬의 질을 향상시켰다. IBM 모델은 1:m 대응을 허용하므로, 원시어에서 목적어로의 정렬 A1과 목적어에서 원시어로의 정렬 A2를 얻은 후, 두 정렬의 교집합 A 링크에 대해 휴리스틱을 이용해 정렬 포인트들을 추가적으로 확장함으로써 정렬의 질을 향상시켰다. 예를 들어, Och and Ney는 교집합 정렬의 각 포인트들을 A1이나 A2에 나타난 링크 (i,j)를 추가함으로써 확장해 나갔는데, 만약 (i,j)의 인접 링크들인 (i-1, j), (i, j-1), (i+1,j), (i,j+1) 포인트들이 이미 교집합 A에 나타났거나 영어 단어 e_i나 불어 단어 f_j가 둘 다 A에서 정렬되지 않았다면 추가되도록 하였다.

Koehn(2003)은 Och의 휴리스틱을 한층 더 확장하여 사용하였다. 그는 단어 정렬의 결과로부터 순차적 조합을 통해 가능한 모든 구들을 생성한다. 확장된 구들에 대해 상대적 빈도수를 이용하여 구 단위 번역확률을 계산해 내고 주어진 구에 대한 어휘 가중치를 고려하여 다음과 같은 수식을 제안하였다.

$$p(\bar{f}_1^I|\bar{e}_1^I)=\prod_{i=1}^{I}\phi(\bar{f}_i|\bar{e}_i)d(a_i-a_{i-1})p_w(\bar{f}_i|\bar{e}_i,a)$$

$$\phi(\bar{f}_i|\bar{e}_i)=\frac{count(\bar{f}_i,|\bar{e}_i)}{\sum_i count(\bar{f}_i,|\bar{e}_i)} \quad \text{식 19)}$$

$$p_w(\bar{f}_i|\bar{e}_i,a)=\prod_{i=1}^{n}\frac{1}{|\{j|(i,j)\in a\}|}\sum_{\forall(i,j)\in a}w(f_i|e_j)$$

Venugopal et al (2003)도 적절한 척도를 사용하여 IBM 모델의 정렬 결과로부터 구 단위를 추출하였다.

Marcu and Wong (2002)는 구 단위 결합 확률(joint probability) 모델을 구현하였다. 즉, 조건부 번역 확률 p(f|e)대신에 p(e,f)를 구하였는데, 두 문장의 생성확률 p(e,f)는 모든 구대구 번역 확률의 곱으로 나타내어지고 구는 모델 내부에서 은닉 파라미터로 작용하지 않는다. 다시 말해, 순차적 분할에 의해 구가 얻어지고 주어진 문장 쌍에 대해 구의 집합(bags of phrases)을 만들어 낸다. 예를 들어 "a b c"-"x y"의 문장쌍에서는 두 가지 형태의 구대구 번역집합이 만들어 진다. t("a b","y"), t("c","x")와 t("a b c", "x y")의 두 가지 가능성만이 존재하도록 하였다. 즉, t("a b","y"), t("c","y")나 t("a c","x"), t("b","y")는 허용되지 않는다. 또한, 학습에 모든 구를 다 반영할 수 없기 때문에 빈도수가 높은 n-gram만을 고려하도록 조정하였다.

이들뿐만 아니라 구조적 정보가 고려된 연구도 많이 이루어졌다. Yamamoto (2000)의 연구에서는 구 단위 대응 관계를 추출하는 방법을 제시하였다. 그들은 번역시 단어의 순서나 위치가 달라질

수 있지만 단어들 간의 의존 구조는 번역 후에도 유지된다고 가정하였다. 대역 말뭉치 쌍들은 각각의 언어에 대해 통계적 의존 관계 파서를 이용해 각각 의존 구조 후보들이 생성된다. 파서가 의존 구조 후보를 뽑을 때 여러 가지 형태의 후보를 생성하여 가중치 다이스 계수(weighted dice coefficient)를 사용해서 상대적 공기 빈도가 높은 두 구를 후보간의 해당 번역구로 선택하였다.

Ye-Yi Wang (1998)은 파서를 이용하지 않고 구조를 파악하기 위해 단어 클러스터링을 이용하여 구를 형성하였다. 먼저 비슷한 의미나 동일한 문법 기능을 가지는 단어들이 상호 정보(mutual information) 요구도에 의한 클러스터링 알고리즘을 이용하여 동일한 클래스로 묶인다. 예를 들면 [on, at, ...], [Sunday, Monday, ...], [afternoon, morning, ...], [I, We, ...], [meet, get, ...]과 같다. 이렇게 구해진 단어 클래스를 기반으로 각 단어를 클래스로 치환한 다음 식 20)의 조건을 만족하면 클래스들을 합쳐 구를 형성하였다.

$$p(c_1,c_2,...c_k)\log\frac{p(c_1,c_2,...,c_k)}{p(c_1)p(c_2)...p(c_k)}>\theta \qquad \text{식 20)}$$

결과적으로 문장은 단어 클래스의 열들로 이루어진 얕은 구문 구조(shallow phrase struct ure)로 표현된다. 예를 들어 "I could meet on Wednesday afternoon" 라는 문장에 대해서는 [I could meet] 와 [on Wednesday afternoon]의 두개의 구가 사용되고 이러한 구들을 기본 단위로 하되 내부적으로는 IBM 모델의 형식의 번역 방식을 취하였다. 해당 모델은 구대구 매핑을 시도하고 구의 내부에서 단어 매핑을 시도하는 chunk-to-string 모델이다.

Wu(1997)는 ITG(Inversion Transduction Grammar)를 제안하여 병렬 말뭉치 쌍을 파싱하였다. ITG는 두 언어가 동일한 하나의 문법 즉, 동일한 문법 구조를 공유하기 때문에 발생할 수 있는 어순의 차이를 고려하기 위해 CFG의 RHS를 구성하는 구성 요소들이 반대의 방향으로도 결합할 수 있도록 허용하였으며, A -> x/y 형태의 어휘 생성 규칙을 이용하였다. 여기서 x는 원시 언어의 단어이고 y는 이에 대응되는 목적 언어의 단어다. 이러한 ITG는 inside-outside 알고리즘을 통해 각 규칙에 확률값이 부여되는 Stochastic ITG(SITG)로 확장되고 SITG를 이용하여 각 문장 쌍에 대해 Earley 파싱에 의해서 가장 높은 확률의 파스 트리를 발견하는 알고리즘을 제안하였다.

Alshawi (2000)는 원시 언어의 문장을 목적 언어의 문장으로 변환하기 위한 WHT(Weighted Head Transducer) 알고리즘을 개발하여 대응 문장에 대해 동기화된 의존 트리를 생성하였다.

앞에서 설명한 Wu (1997)와 Alshawi (2000)의 연구는 두 언어를 동시에 생성하기 위한 모델을 만든 반면, Yamada (2001)의 연구에서는 이 방법을 번역 과정의 측면에서 고려하였다. Yamada (2001)는 구조적 특징을 정렬 모델에 첨가하기 위해서 SMT의 입력을 영어의 파스 트리로 주고 채널 오퍼레이션에서는자식 노드의 재순서화(reordering) r, 각 노드에 기능어 삽입(functional word insertion) n, 단말 단어의 번역(translation) t라고 하는 세 가지 변형 과정(transformation operation)을 거쳐 일본어 문장이 생성된다. 특정 영어 파스 트리가 주어졌을 때 특정 일본 문장이 나타난 확률은 그 문장을 만들 수 있는 가능한 모든 변형 확률의 합으로 표현할 수 있다. ε1,ε2,....,εn 노드로 구성된 영어 파스 트리 ε가 특정 불어 문장 f로 번역될 확률은 다음과 같이 나타낼 수 있다. 여기서 Str(θ(ε))는 ε가 θ에 의해 변형되었을 때 생기는 단말 노드의 단어열이 f와 같음을 의미한다. 각 확률파라미터 n, r, t는 EM 알고리즘을 적용하여 계산되는 tree-to-string모델이다.

$$p(\mathbf{f}\mid\boldsymbol{\varepsilon}) = \sum_{\theta:Str(\theta(\varepsilon))=\mathbf{f}} P(\boldsymbol{\theta}\mid\boldsymbol{\varepsilon}) = \sum_{\theta:Str(\theta(\varepsilon))=\mathbf{f}} \prod_{i=1}^{n} n(\nu_i \mid \mathrm{N}(\varepsilon_i))r(\rho_i \mid \mathrm{R}(\varepsilon_i))t(\tau_i \mid \mathrm{T}(\varepsilon_i)) \quad \text{식 21)}$$

또한, Charniak (2003)은 Yamada(2001)의 번역 모델에 통계적 언어 모델을 결합하여 번역의 질을 향상시켰으며, Yamamoto and Mat sumoto (2000)의 연구에서는 원시어 및 목적어를 모두 파싱해 놓고 그것을 학습 데이터로 삼는 parse-to-parse 모델을 제시하였다.

Watanabe (2003)의 연구에서는 청커(chunker)나 파서(parser)와 같은 외부 자원을 이용하지 않고 청크 정보가 모델 내부에 파라미터로 동작하여 EM 학습 시 자동으로 유도되어진다. 이 방법은 청크 층을 가진 stri ng-to-string 모델에 해당된다고 할 수 있다. 해당 모델에서는 번역의 과정을 단어를 머리어에 결합시키는 청킹, 삭제와 복사(fertility), NULL 단어 삽입, 어휘적 번역, 청크 내부의 재순서화 및 청크 재순서화 순으로 보았으며, 과도한 학습량을 줄이기 위해 Yamada의 Inside-Outside 기반 학습 방법을 변형하여 사용하였다.

그러나 구나 구조를 반영하기 위한 연구들도 다음과 같은 한계를 지니고 있다. 구문 분석을 하지 않고 단어 그룹을 구로 대체하는 연구에서는 문법적으로 올바르지 않은 구성요소가 정렬의 단위로 동작할 수 있다. 또한 구문 분석을 가정한 tree-to-tree나 tree-to-string 정렬의 경우 흔히 두 언어 간의 구조적 상이성 때문에 올바른 정렬 결과를 보이기 어렵고 복잡하고 계층적인 파스트리 상에서 정렬과 문장 번역을 설명하기 때문에 모델의 복잡도가 커지게 된다.

이러한 문제점을 해결하고자 Kim (2004)의 연구에서는 복잡도가 큰 파스트리 대신 더 평탄화된 구조인 청크 단위를 입력으로 하는 두 단계 청크 기반 정렬 모델을 제안하였다. Watanabe (2002)의 연구와 다른 점은 문장 번역 모델의 내부 구조에도 있지만 외부 자원인 청커를 이용한다는 점이다. 청크를 모델 내부에서 자동으로 유도되도록 모델링 할 수도 있겠지만 청커는 구현하기도 쉽고 그 정확성도 보장되기 때문에 내부 파라미터로 사용해 부정확성과 모델의 복잡성을 피하였다. 또한 단어와 구가 동시에 표현될 수 있는 한영 정렬 모델을 제안하고 있다. 한영의 경우 구조적 차이가 심해 구문 구조를 파스 트리로부터 정렬을 얻어내는 것은 좋지 않다. 또한 구조를 반영하기 위해 파스 트리의 비단말 노드를 사용하는 것은 지나친 단순화로 모호성을 심화시킨다. 이 연구에서는 품사 태그 순서열(POS tag sequence)을 NP나 VP대신 구를 나타내는 태그(phrasal tag)로 사용하였고 태그열의 매핑정보를 구문적 대응 정보로 사용하였다. 이에 따르면 "DT(determiner) +NN (noun)"의 영어 구태그는 한국어의 "NN (noun)+SUBJ(주격조사)", "NN(noun)+ OBJ(목적격 조사)"의 구태그에 대응될 가능성이 높다. 사용된 모델의 형태는 다음 수식과 같다.

$$p(\bar{\mathbf{k}}\mid\bar{\mathbf{e}}) \approx \sum_{\mathbf{a}} \prod_{i=1}^{q} n(T(\bar{k}_i)\mid T(\bar{e}_{h_i}))\times \prod_{j=1}^{\phi_i} s(T(k_{ij})\mid T(e_{h_i a_j}))w(k_{ij}\mid e_{h_i a_j}) \quad \text{식 21)}$$

여기서 n은 구조적 매핑을 위해 사용되는 파라미터, s는 단어의 품사 매핑을 위해, w는 어휘단위의 번역을 위해 고려된 파라미터다.

마지막으로 정렬 시스템의 평가를 위해 BLEU Score나 WER(word error rate), PIWER (position independent WER)등의 척도들이 제안되었는데, 이들은 객관적 평가 여부, 개별적 모델에 대한 평가 여부에 따라 선택적으로 이용되어 왔다.

5. 결 론

본 논문에서는 SMT 개요와 현재 기술수준을 논하였다. SMT는 아직까지는 시작단계이나 통계적 언어 처리의 가능성으로 볼 때 앞로의 번역시장을 주도할 만한 기계번역의 한 대안인 것은 분명하다. 많은 수식과 복잡한 모델링이 필요한 SMT의 이해를 돕고자 IBM 모델을 중심으로 번역 모델링을 살펴보았고 각 모델에 사용된 파라미터 학습을 위해 EM 알고리즘과 해당 카운트를 계산하는 방법을 설명하였다. 최근 연구동향으로는 과거의 언어학적 지식이 거의 부과되지 않았던 모델과는 달리 구조적 정보를 부과하여 정렬의 성능을 향상시키는 연구가 많이 진행되고 있다. 또한 구조적 정보를 EM 내부에서 구조를 자동적으로 유도해 나가는 방법은 과도한 학습 파라미터 수 때문에 실제적인 시스템에 응용이 어려운 점이 있는데 가능한 정렬만을 대상으로 학습하는 학습 알고리즘 상의 개선이 앞으로의 SMT 부분의 숙제다.

참고문헌

[1] Alshawi, Hiyan: Srinivas Bangalore and Shona Douglas, "Learning Dependency Translation Models as Collections of Finite-State Head Transducers", Computational Linguistics, 26(1): 45-60, 2000.

[2] Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra, "A maximum entropy approach to natural language processing", Computational Linguistics, 22(1):39-73, 199

[3] Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer, "The mathematics of statistical machine translation :parameter estimation", Computational Linguistics, 19(2):263-311, 1993.

[4] Charniak, Eugene, Kevin Knight and Kenji Yamada, "Syntax-based Language Models for Statistical Machine Translation", In Proceedings of MT Summit IX, 2003.

[5] Dempster, A. P., N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", The Royal Statistics Society, 39(B) 205-237, 1976.

[6] Knight, Kevin, "Statistical MT Tutorial Workbook", JHU Workshop, 1999

[7] Kim, Seonho, Juntae Yoon, Mansuk Song, "Improved Lexical Mapping Model of English-Korean Bitext Using Structural Feature", In Proceedings of EMNLP/NAACL 2001

[8] Kim, Seonho, Juntae Yoon, Dong-Yul Ra, "Two-Level Alignment by words and phrases Based on Syntactic Information", Computational Linguistics and Intelligent Text Processing (CICLing) 2004, LNCS 2945, 309-320.

[9] Koehn, Philipp, Franz Josef Och, and Daniel Marcu, "Statistical Phrase-Based Translation", In Proceedings of HLT/NAACL 2003.

[10] Marcu, Daniel and William Wong, "A phrase-based, joint probability model for statistical machine translation", In Proceedings of EMNLP 2002.

[11] Och, Franz Josef and Hermann Ney, "Improved statistical alignment models", In Proceedings of ACL, 440-447, 2000.

[12] Venugopal, Ashish, Vogel, Stephan, and

Waibel, Alex, "Effective Phrase Translation Exctraction from Alignment Models", In Proceedings of ACL 2003, 319-326.

[13] Wang, Ye-Yi and Alex Waibel, "Modeling with structures in machine translation", In Proceedings of ACL36/COLING, 1357-1363, 1998.

[14] Watanabe, Taro, Sumita, Eiichiro, and Okuno, G. Hiroshi, "Chunk-based Statistical Translation", In Proceedings of ACL 2003, 303-310.

[15] Wu, Dekai, "Stochastic Inversion Transduction Grammar and Bilingual Parsing of Parallel Corpora", Computational Linguistics, 23(3):377-403.

[16] Yamada, Kenji and Kevin Knight, "A Syntax-based statistical translation model", In Proceedings of ACL 2001, 523-530.

[17] Yamamoto, Kaoru and Yuji Matsumoto, "Acquisition of Phrase-level Bilingual Correspondence using Dependency Structure", In Proceedings of COLING 2000, 933-939

저자약력

김 선 호

2002 연세대학교 컴퓨터과학과박사
2002~2003 연세대학교 언어정보개발 연구원, 연구원
2003~현재 고려대학교 자연어처리연구실, 연구원
관심분야:자연어처리,기계번역,기계학습, 생물정보학
E-mail: shkim@nlp.korea.ac.k

윤 준 태

1998 연세대학교 컴퓨터과학과박사
1998~1999 한국과학기술연구원 ,인공지능센터, 연구원
2000 Univ. of Pennsylvania, IRCS, 연구원
2002~현재 Daumsoft. 자연어 처리 연구소 소장
관심분야 : 자연어처리,구문분석, 정보검색, 텍스트 마이닝, 기계학습
E-mail : jtyoon@daumsoft.com

임 해 창

990 Texas 주립대학 컴퓨터학과 박사
1991~현재 고려대학교 컴퓨터 학과 교수
1998.5~2000.5 정보과학회 한국어정보처리연구회 운영위원장
2001~현재 ACM Transaction on Asian Language Information Processing Associate Editor
관심분야 : 자연어처리, 정보검색, 생물정보학
E-mail : rim@nlp.korea.ac.kr