k-최근점 학습에 기반한 타동사-목적어 연어 사전의 최적화

(Optimization of Transitive Verb-Objective Collocation Dictionary based on k-nearest Neighbor Learning)

김 유 섭 † 장 병 탁 †† 김 영 택 ††
(Yuseop Kim) (Byoung-Tak Zhang) (Yung Taek Kim)

요 약 영한 기계번역에서 영어 문장의 동사구를 한국어로 정확하게 번역하기 위해서는 일반적으로 타동사와 목적어의 연어 관계를 이용한다. 본 논문에서는 k-최근점(k-nearest neighbor) 학습을 연어 관계에 적용하여 동사 번역을 선택하는 알고리즘을 제시하였는데 k-최근점 학습을 위해서 워드넷에서의 의미거리를 정의하여 사용하였다. 그리고 실시간 번역 시스템에 사용될 사전을 구성하기 위하여, 말뭉치로부터 타동사-목적어 쌍을 추출하여 학습예제를 구축하고, 이 예제의 크기를 번역률과 연관시켜 최적화시키는 알고리즘을 제시한다. 본 논문에서는 위의 알고리즘들을 사용하여 동사 "build"의 번역률을 약 90%로 유지하면서 사전의 크기를 최적화하였다.

Abstract In English-Korean machine translation, transitive verb-objective collocation is utilized for accurate translation of an English verbal phrase into Korean. This paper presents an algorithm for correct verb translation based on the k-nearest neighbor learning. The semantic distance is defined on the WordNet for the k-nearest neighbor learning. And we also present algorithms for automatic collocation dictionary optimization. The algorithms extract transitive verb-objective pairs as training examples from large corpora and minimize the examples, considering the tradeoff between translation accuracy and example size. Experiments show that these algorithms optimized collocation dictionary keeping about 90% accuracy for a verb "build".

1. 소 개

1.1 연구 배경 및 관련 연구

기계번역은 어휘분석, 구문분석, 대역어 선택, 목표문장 생성 과정 등에서 다양한 수준의 처리가 필요하나, 번역 선택을 올바로 하면 문장 분석이 완전하지 못한 경우에도 번역의 이해도를 높일 수 있다. 특히 문장 성분 중 동사는 전체 문장에서 가장 핵심적인 의미를 가지기 때문에 보다 정확한 번역의 선택이 필요하다. 또한 대역어의 선택에서는 원시언어의 어의중의성과 목표언어의 표현중의성을 동시에 해결해야 하므로 충분한 연어 정보를 가지고 있는 사전이 중요한 역할을 한다. 예를 들면, 동사 "have"는 일반적으로 "가지다"로 번역되지만, 그 목적어가 "fruit(과일)"인 경우에는 "먹다"로 번역되고 목적어가 "tour(관광)"인 경우는 "하다"로 번역되어야 한다. 이러한 정보는 동사-목적어 연어 사전에 구축된다. 본 연구를 이해하기 위하여 기계번역에서의 연어에 관련된 기법을 중심으로 관련 연구를 살펴보기로 한다.

Dagan et al.[1]은 히브리어-영어 기계번역을 위하여 원시언어 연어쌍에 대응하는 가능한 모든 목표언어 연어쌍을 구하고, 목표언어 말뭉치에서 빈도가 가장 높은 연어쌍을 대역어로 선택하였다. 그러나 이 방법은 목표언어 말뭉치에서 발생하는 데이타 희소성(data sparseness) 때문에 계산된 통계값을 신뢰하기 어려운 단점을 가지고 있다.

† 학생회원 : 서울대학교 컴퓨터공학과
yskim@nova.snu.ac.kr
†† 종신회원 : 서울대학교 컴퓨터공학과 교수
btzhang@comp.snu.ac.kr
ytkim@comp.snu.ac.kr
논문접수 : 1999년 5월 25일
심사완료 : 2000년 1월 11일

Kim et al.[2]은 한국어-영어 기계번역에서 동사의 번역을 선택하기 위하여 말뭉치로부터 추출한 동사-목적어 연어 리스트를 이용하였다. 그러나 이 방법 역시 한국어 말뭉치에서 발생하는 데이타 희소성 때문에 연어 리스트에 없는 목적어가 입력되었을 때는 정확한 의미를 찾을 수 없는 한계를 가지고 있다.

일반적으로 데이타 희소성을 해결하기 위해서는 말뭉치에서 유사한 다른 단어들을 찾아서 대신 사용하는 유사도(similarity) 기반 모델과 워드넷(WordNet) 등의 시소러스(thesaurus)에 기반한 부류(class) 기반 모델이 사용된다.

Dagan et al.[3], Dagan et al.[4] 그리고 Karov et al.[5]은 유사도 기반 모델을 설계하여 모든 단어마다 유사한 단어들의 리스트를 구성하였고, 학습예제에 나타나지 않은 연어쌍의 번역을 구할 때는 리스트의 단어들을 대신 사용하여 번역을 선택하였다. 이 모델은 모든 단어가 유사어를 그 가중치와 함께 리스트로 구성하여 개별 단어의 특성을 잘 반영하도록 했으나, 모든 단어들의 유사도를 미리 계산해야 하기 때문에 학습시 계산량이 너무 많고 말뭉치에서의 빈도가 작은 단어들은 유사도 계산이 어렵다. 또한 유사도를 계산할 때, 단어의 다의성을 고려하지 못하고 데이타의 크기가 반드시 커야 한다.

Resnik[6]은 워드넷 계층구조를 변형하여 다수의 부류(class)로 명사들을 구분하였고 Yarowsky[7]은 Roget's 시소러스로 부류를 정의하여, 입력된 단어와 동일한 부류에 속한 단어들을 대체 단어로 사용하여 번역을 선택하였다. 여기서는 말뭉치에서의 빈도가 작은 단어들도 비교적 정확하게 그 특성을 반영할 수 있으나, 동일 부류의 단어들을 차별화 시킬 수는 없다. 또한 이 방법 역시 통계적인 계산을 필요로 하므로 다량의 데이타가 필수적이다.

그리고 사전을 말뭉치로부터 구축하는 방법에 대해서도 연구가 진행되었는데, 특히 무감독 학습(unsupervised learning)을 이용하여 유사한 단어들을 집단화(clustering) 함으로써 사전을 구성하는 연구가 진행되었다[8, 9, 10]. 이 방법은 사람의 개입이 없이 사전을 자동으로 구성할 수 있다는 장점을 가지고 있으나, 이들은 원시언어 말뭉치만을 데이타로 사용했기 때문에, 목표언어 사용자들의 언어적 관습을 반영할 수 없다.

1.2 연구 개요 및 논문의 구성

본 논문은 영한 기계번역에서 연어에 기반하여 동사의 번역을 선택하는 방법과, 번역시 필요한 연어 사전을 구성하고 크기를 최적화하는 방법에 대하여 논한다. 번역 선택 시 발생하는 데이타 희소성 문제를 해결하기 위하여, 워드넷에서 유사어를 찾아서 사용하였다.

본 논문에서는 워드넷 상에서의 의미거리에 기반한 k-최근접 학습 알고리즘[11, 12, 13, 14]을 타동사와 목적어의 연어 관계에 적용하여 동사의 번역을 선택하였다. k-최근접 학습은 학습 데이타의 크기가 커야 하고 또한 데이타의 전체적인 구조보다는 지역적인 구조정보에 더욱 의존하는 한계를 가지고 있으나[15], 워드넷 계층구조 공간으로 전체 탐색공간을 제한하고 워드넷의 전체적인 구조를 사전지식으로 사용함으로써 이러한 한계를 극복하였다.

그리고 본 논문에서는 감독 학습(supervised learning)을 통하여 연어 사전을 구성하고 최적화하였다. 동사 "build"는 "make something by joining things together"라는 단일 의미를 가지나[16], 목적어가 "plant"이면 "건설하다"를, 목적어가 "car"이면 "제작하다"를 대역어로 선택하는 것이 더 자연스럽다. 이를 목표언어의 표현 중의성 이라 하는데, 이 문제를 해결하기 위한 사전은 원시언어 말뭉치만을 사용하는 무감독 학습 방법으로는 구성할 수 없다.

연어 사전은 대량의 말뭉치에서 추출된 학습예제로부터 구성되는데, 학습예제의 크기를 계속 확장한다면 번역은 더 정확해질 수 있지만 실시간 번역 시스템에서의 번역 속도는 예제의 수가 늘어날수록 느려지게 된다. 그러므로 학습예제 리스트를 핵심의미(prototype)를 중심으로 하여 최적화시키는 알고리즘이 필요하다. 본 논문에서는 워드넷의 특성과 새로이 정의된 동사 번역 유사도(similarity degree)를 사용하여 학습예제를 최적화하였다.

실험에서, "build" 동사를 90%에 가깝게 번역하였고 또한 번역률을 거의 희생하지 않으며 사전을 약 30% 수준으로 그 크기를 줄였는데, 이 과정에서 번역선택에 큰 영향을 미치지 못하는 예제들과 번역에 악영향을 미치는 예제들을 제거할 수 있었다.

2장에서는 연어를 정의하고 연어 사전을 생성하는 과정을 설명하였으며 3장에서는 워드넷의 구조를 설명하고 워드넷상에서의 의미거리를 정의하였다. 그리고 번역선택을 위한 k-최근접 학습에 대하여 설명하였다. 4장에서는 연어사전을 최적화하는 알고리즘과 동사번역 유사도에 대하여 설명하고 5장에서는 실험 결과와 그 평가를, 6장에서는 결론을 서술하였다.

2. 연어사전의 생성

2.1 연어

연어는 대량의 말뭉치에서 나타난 예제 문장으로부터 추출되고 다음과 같이 정의된다[17].

$$M(W_\alpha) = \begin{cases} M_\beta & \text{if } Col(W_\alpha, W_\beta) \\ M_\gamma & \text{if } Col(W_\alpha, W_\gamma) \\ \cdots & \\ M_\delta & \text{otherwise} \end{cases}$$

($M(k)$: 단어 k의 의미, $Col(W_i, W_j)$: W_i 와 W_j가 공기함, M_k : 의미)

여기서 단어는 공기(co-occurrence)하는 단어에 따라서 각기 다른 의미를 가질 수 있으며, 특히 통사적으로 밀접한 관계에 있는 단어에 따라 그 의미가 결정된다[18]는 것을 알 수 있다.

"build" 라는 동사에 대한 다음 예를 생각해 보자.

"build"
"You must *build* a ***plant***."
--> "당신은 ***공장***을 *건설해*야 합니다." (*build*, ***plant***)
"The line *builds* the ***car***."
--> "그 라인은 ***자동차***를 *제작합니다*." (*build*, ***car***)
"They *build* a ***company***."
--> "그들은 ***회사***를 *설립합니다*." (*build*, ***company***)

위에서 동사 *"build"*는 목적어가 ***"plant(공장)"***이면 "*건설하다*"로, ***"car(자동차)"***이면 "*제작하다*"로, 그리고 ***"company(회사)"***면 "*설립하다*"로 번역된다.

2.2 연어 사전의 생성

연어 정보는 기계번역 시스템에서 사전에 저장되는데 연어 사전 생성 과정은 그림 1에 나타나 있으며 다음과 같이 설명된다.

첫째, 대량의 말뭉치로부터 파서를 이용하여 원하는 동사와 그 목적어의 쌍을 자동으로 추출한다. 둘째, 추출된 쌍에서 목적어의 의미를 문장에서의 문맥을 보아 결정하는데 여기서의 의미는 워드넷 상에서의 의미를 말한다. 또한 동사 번역 역시 문맥을 통하여 결정하며, 동사의 번역 부류는 각 동사마다 미리 정의한다. 이 단계에서 명사 의미와 동사 번역의 결정은 수동으로 이루어진다. 위에서 결정된 목적어의 의미, 동사 번역, 그리고 빈도가 학습예제의 구성요소가 되어 초기 학습예제 리스트가 구성된다. 셋째, 초기 학습예제 리스트는 본 논문에서 제시된 사전 최적화 알고리즘을 통하여 최적화되고, 마지막으로 이 최적화된 학습예제 리스트는 연어 사전으로 변형된다. 다음 표 1은 위의 과정을 거쳐서 생성된 연어 사전의 한 예이다.

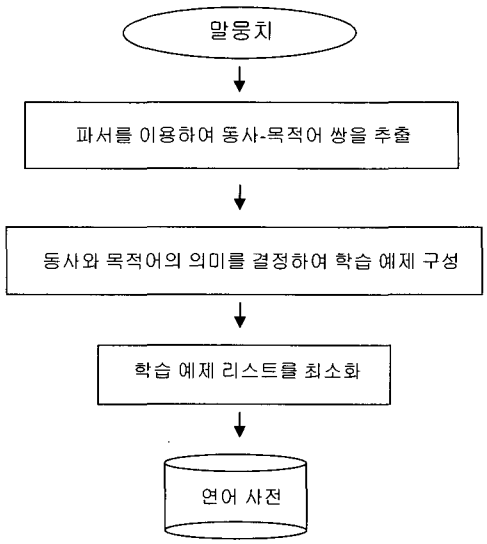

그림 1 사전 생성 과정

표 1 "build"의 연어 사전의 예

"build"				
"건축하다"	house 주택	center 센터	housing 주택	church 교회
"건설하다"	plant 공장	ship 선박	network 네트워크	park 공원
"제작하다"	car 자동차	ship 선박	model 모델	truck 트럭
"설립하다"	company 회사	market 시장	empire 복합기업	bank 은행
"구축하다"	system 시스템	stake 몫	relationship 관계	support 지원

그런데 입력된 문장의 목적어를 단순히 연어사전에서 찾아서 동사의 번역을 선택한다면, 연어 사전에 포함되어 있지 않은 목적어가 입력될 때는 동사의 번역을 선택할 수 없다. 예를 들어 목적어가 사전에 정의되지 않은 단어 "vehicle(차량)"일 때는 "build"의 번역을 선택할 수 없다. 이 문제를 해결하기 위하여 본 논문에서는 아래의 3장에서 설명되는 워드넷의 의미거리에 기반한 k-최근접 학습방법을 연어 사전에 적용시켰다.

3. 번역 선택 알고리즘

본 논문에서 동사 번역 선택을 위하여 사용된 방법은 사례기반 학습(instance-based learning) 방법중 하나인

k-최근접 학습 방법이다. 사례기반 학습은 학습예제가 주어졌을 때 일반적이고 명시적인 목표함수(target function)를 구성하는 다른 학습 방법들과는 달리 단순히 학습예제를 수집하며, 학습예제를 일반화하는 과정은 분류될 예제가 입력될 때까지 연기된다. 3.1에서는 워드넷의 구조와 특성에 대하여 설명하고 워드넷 상에서 두 의미간의 거리를 정의하였다. 그리고 3.2에서는 사례기반 학습 중 본 논문에서 사용된 k-최근접 학습에 대하여 설명하였다 .

3.1 워드넷과 의미거리

워드넷은 언어 심리학적인 이론에 근거하여 설계된 인간의 어휘 기억(lexical memory)의 온라인 어휘 참조 시스템으로, 영어의 명사, 동사, 형용사, 부사를 동의어(synonym) 집단으로 구성함으로써 어휘 개념(lexical concept)을 표현하였다[19]. 동의어 집단은 신셋(synset)이라 불리며 워드넷 구성의 기본이 되고, 각각의 신셋은 전체 워드넷 상에서 특정한 계층에 놓이게 된다. 본 논문에서 사용된 워드넷의 구조는 그림 2와 같다.

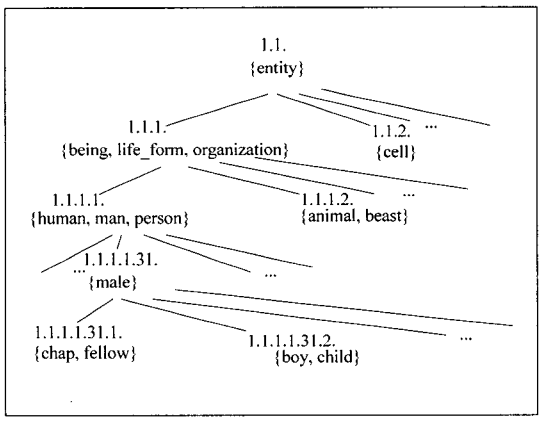

그림 2 워드넷의 구조

예를 들면, 명사 "boy"는 신셋 "1.1.1.1.31.2."에 동의어 "child"와 함께 포함되어 있으며 동시에 "entity", "human", "male" 등의 하위개념으로 정의되어 있다.

워드넷에서 의미 유사도 또는 의미간 거리를 측정하는 방법은 계속 연구되었다. 가장 간단한 방법으로는 두 의미 사이의 최단 경로에 있는 노드의 수를 세는 것이다[20]. 이 방법은 계산이 매우 간단하지만, 워드넷의 구조가 가지고 있는 다양한 특성을 잘 반영하지 못한다.

Resnik[6]은 하나의 의미와 그 하위 의미들을 모두 포함하여 그 의미의 클라스라 정의하였고 두 의미의 클라스를 동시에 포함시키는 가장 작은 상위 클라스를 두 의미의 슈퍼클라스라 정의하였다. 그리고 슈퍼클라스에 포함되는 모든 단어들의 말뭉치에서의 빈도를 구하여 두 의미의 유사도를 측정하였다. 이 방법은 실제 말뭉치에 나타나는 현상을 반영할 수 있으나 다의어나 동철이의어는 말뭉치에서의 빈도가 과장되는 문제를 가지고 있었다. 예를 들면, "bank" 는 "은행"과 "강둑" 두개의 의미를 가지고 있다. 이 중 "은행"의 의미와 다른 단어의 의미 유사도를 계산할 때, 말뭉치에서는 "은행"과 "강둑"의 두 의미의 구분이 없이 그냥 단어 "bank"의 빈도만을 고려하므로 의미 "은행"의 빈도는 실제보다 더 과장된다.

이밖에 Richardson[19]은 위의 두 방법을 혼합하여 워드넷의 링크에 가중치를 부여하는 방법을 연구하였다. 그러나 이 방법 역시 [6]의 방법에서 발생하는 정보의 과장현상을 처리하기 위하여 의미 중의성 해소기(sense disambiguator)가 필요하다는 문제를 가지고 있다. 또한 위의 통계정보를 이용한 방법은 계산시에 막대한 시간이 소비되기 때문에 두 의미간의 유사도를 번역 과정에서 실시간으로 활용하기에는 곤란하다.

본 논문에서는 의미간의 유사도를 측정할 때 실시간 번역 시간에 큰 영향을 미치지 않고 말뭉치에 포함된 과장된 정보의 영향을 받지 않는 방법을 제안하였다. 먼저 본 논문에서의 단어의 의미는 그 단어가 포함된 신셋이라 정의하고, 의미거리는 워드넷 상에서의 신셋간 거리라 정의한다. 본 논문에서 제안된 의미거리의 기본 속성은 다음과 같다:

두 신셋과 이들이 최초로 공유하는 상위 신셋을 세 꼭지점으로 하는 삼각형이

1) 워드넷의 하위부로 위치할수록 그리고

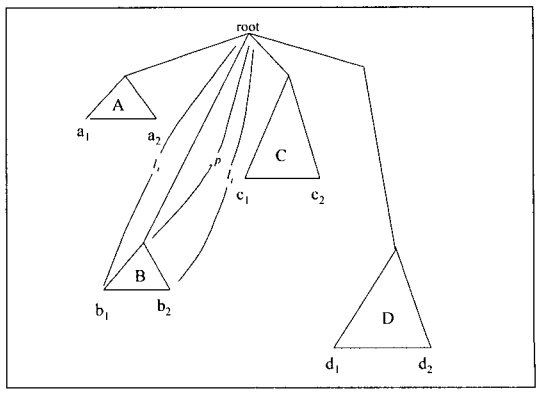

그림 3 의미거리 삼각형

2) 크기가 작을수록
두 신셋의 거리는 짧아진다

예를 들어, 그림 3에서 의미거리 삼각형 B 가 같은 크기의 삼각형 A보다 워드넷 상에서 하위에 위치하므로 신셋 b_1 과 b_2 의 의미거리는 a_1 과 a_2 의 의미거리보다 더 짧다. 또한 삼각형 B가 같은 계층의 삼각형 D 보다 크기가 더 작기 때문에 신셋 b_1 과 b_2 의 의미거리는 d_1 과 d_2 의 의미거리보다도 더 짧다.

그림 3에서 p는 두 신셋 b_1 과 b_2 가 최초로 공유하는 상위 계층의 루트로부터의 거리이고, l_i 와 l_j 는 각각 b_1 와 b_2 의 루트로부터의 거리라고 할 때, 두 신셋 b_1 와 b_2 간의 거리 $D(b_1, b_2)$를 다음과 같이 정의된다.

$$D(b_1,b_2)=\sum_{k=0}^{l_i-p}\frac{M}{sf^k}+\sum_{k=0}^{l_j-p}\frac{M}{sf^k} \quad (1)$$

여기서 $M = radix / sf^p$ 이고 $radix$ 와 sf 는 각각 초기값과 스케일 단위를 나타내는 상수값이다. $radix$ 는 그 값이 의미거리에 아무런 영향을 미치지 않는 단순한 초기 상수값이며, sf 는 의미 거리 삼각형의 크기와 위치 중 어디에 더 큰 비중을 줄 것인가를 결정하는 값이다. sf 의 값이 클수록 크기보다는 위치에 더 큰 비중을 두게 되며 작을수록 크기에 더 큰 비중을 둔다. 본 논문에서는 실험을 위하여 $radix$를 8.0으로 sf는 2.0으로 정하였다.

식(1)에서 M은 p에 반비례하는데, 이것은 두 신셋은 의미거리 삼각형이 워드넷 계층구조에서 하위에 위치할수록 더 가까운 의미거리를 가진다는 것을 의미한다. 그리고 식 (1)의 우변에 있는 두 항은 각 신셋과 공유계층과의 거리에 비례하는 값을 가지는데, 이는 두 신셋이 공유계층에서 멀수록 유사하지 않다는 것을 말한다.

위의 식을 이용하여 두 단어의 의미거리를 측정하는 예는 다음과 같다. 명사 "plant(4.1.4.8.1.10)"과 "home (4.1.4.8.2.1)"은 4번째 계층 "4.1.4.8"까지는 공유하고 있으므로 M의 값은 "$8.0 / 2.0^4 = 0.5$"가 된다. 그리고 두 단어 모두 그 아래로 2개의 계층이 더 존재하므로 전체 $D(b_1, b_2)$ 의 값은 "$(0.5 + 0.25 + 0.125) + (0.5 + 0.25 + 0.125) = 1.75$"가 된다.

3.2 k-최근접 학습 (k-nearest neighbor learning)

본 논문에서 사용된 학습 알고리즘은 k-최근접 알고리즘이다. 이 알고리즘은 모든 사례(instance) x가 n-차원 공간 R^n에 있는 한 점이라 가정하고, 그 공간에서 거리가 가장 가까운 점을 최근점이라 하는데 본 논문에서는 워드넷 상에서의 의미거리 $D(a, b)$를 두 사례 a와 b의 거리로 사용하였다. 이 학습에서는 $f : R^n \rightarrow V$ 의 형태를 가지는 이산값 목표 함수 (discrete-valued target function)를 학습하는 것이 주목적이며 V는 s개의 원소를 갖는 유한집합 $\{v_1,..., v_s\}$이다.

이 방법에서의 학습은 학습예제 $<x, f(x)>$를 학습예제 리스트에 단순히 추가하는 것으로 이루어지고, 분류될 질의사례(query instance) x_q 가 주어지면 x_q 와 거리가 가장 가까운 k개의 사례 $x_1,...,x_k$ 를 추출한 뒤 다음의 추정함수(estimating function) $\hat{f}(x_q)$의 값을 구하여 x_q의 부류 $f(x_q)$를 추정한다.

$$\hat{f}(x_q)\leftarrow \arg\max_{v\in V}\sum_{i=1}^{k} w_i\delta(v,f(x_i)) \quad (2)$$

여기서 $w_i \equiv 1/D(x_q, x_i)$ 이고 $\delta(a, b)$는 다음과 같이 정의된다.

$$\delta(a,b)=\begin{cases}1 & \text{if } a=b\\ 0 & \text{otherwise.}\end{cases}$$

본 논문에서는 식 (2)를 기반으로 하여 동사 번역을 선택하였다. (2)에서 x를 워드넷에서의 명사의미로, $f(x)$를 목적어의 의미가 x일 때의 동사 번역이라 하였다. 그리고 V는 미리 정의된 동사 번역 부류라 하였으며 k를 1로 정의하였다.

다음은 식 (2)를 이용하여 동사-목적어 쌍 "build-highway"에서의 동사 "build"의 번역을 선택한 예이다.

1) 목적어 "highway"의 워드넷으로부터 의미 "1.1.4.3.19.5.6."를 x_q로 얻는다.
2) 입력 질의 x_q에 대한 $f(x_q)$를 추정하기 위하여 x_q 와 가장 가까운 학습예제 x_i를 사전에서 찾는다. 예를 들면,
   x_q = "1.1.4.3.19.5.6."에 대하여 사전으로부터 가장 가까운 예제
   "1.1.4.3.19.5.(road)"를 사전으로부터 찾는다.
3) f("1.1.4.3.19.5.") "건설하다"를 f("1.1.4.3.19.5.6.") 로 추정하여 "highway"에 대한 동사의 번역으로 결정한다.

4. 연어사전의 최적화

이 장에서는 최적화된 연어 사전을 구성하기 위하여 학습예제 리스트를 최적화하는 과정에 대하여 설명한다. 최적화과정에서는 동사 번역 유사도(similarity degree)라는 새로운 개념을 사용하여 학습 예제 리스트에서 제

거될 예제를 선택한다.

4.1 학습예제의 구성

학습예제는 2장에서 설명된 바와 같이 말뭉치로부터 추출되어 구성된다. 다음 (3)은 학습예제의 여러 속성을 설명한 것이다.

$<o, s, fr, c>$, (3)

o : 목적어 단어,

s : o의 워드넷 상에서의 의미,

fr : o의 말뭉치에서의 빈도

c : 목적어가 o일때 동사 v의 번역,

표 2는 동사 "build"에 대하여 추출된 학습 예제의 일부이다.

표 2 말뭉치로부터 추출된 학습예제

목적어 단어 (o)	목적어의 의미 (s)	출현 빈도 (fr)	해당 동사번역 (c)
plant	4.1.4.8.1.10	67	건설하다
car	1.1.4.3.7.3.9.4.1	54	제작하다
home	4.1.4.8.2.1	28	건설하다
ship	1.1.4.3.7.3.9.11.5	26	제작하다
business	9.1.5.2.5.6.2	25	설립하다
house	1.1.4.3.7.9.12.23.6	20	건축하다
system	9.1.8	20	구축하다
facility	1.1.4.3.7.9.13.1.2	19	건설하다
company	9.1.5.2.5.6.2.6	18	설립하다

4.2 초기화

학습 예제 리스트가 구성되면 먼저 학습예제를 초기화시킨다. 본 논문에서는 학습예제를 최적화하기 위하여 벡터와 행렬로 예제 정보를 표현하였고 이 표현이 본 논문에서 제시된 최적화 알고리즘에 사용된다. 최적화 알고리즘의 초기화 과정은 다음 알고리즘 1과 같이 정의된다.

알고리즘 1 초기화

1) 4.1.2에서 구성된 학습 예제로부터 학습예제 벡터 $\mathbf{E}_{tr}$을 구성한다.

$$\mathbf{E}_{tr} =< \mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_i, ..., \mathbf{e}_{i'}, ..., \mathbf{e}_n >,$$
$$\mathbf{e}_i =< o_i, s_i, fr_i, c_i, nullptr_i >, \ (fr_i \geq fr_{i'}, i < i') \quad (4)$$

여기서 n은 전체 학습예제의 수이고 $nullptr_i$ 은 $\mathbf{e}_i$ 가 학습예제 리스트에서 제거되었으면 1을 그렇지 않으면 0의 값을 가지는데 초기값으로 0을 가진다. 그 밖의 $\mathbf{e}_i$의 원소들에 대한 정의는 (3)과 동일하며 빈도에 따른 내림차순으로 정리되어 있다.

3) $\mathbf{E}_{tr}$ 의 원소를 각각 식 (1)을 이용하여 의미거리를 측정하고 다음과 같은 의미거리 행렬 **DM**을 구성한다.

$$\mathbf{DM} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & & & \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{pmatrix} \quad (5)$$

여기서 d_{ij} 는 다음과 같이 정의된다.

$i, j \leq n$ 에서

$$d_{ij} = \begin{cases} D(s_i, s_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

알고리즘 1을 통하여 구성된 초기화 정보는 다음절에서 설명되는 방법으로 생성되는 유사도와 함께 예제 최적화 단계에서 계속적으로 사용된다.

4.3 유사도 (similarity degree)

기계번역에서 번역오류를 계산할 때는 읽는 사람의 이해도가 반영되어야 한다. 예를 들어 "house"가 "build"의 목적어일 때, "build"의 번역은 "건축하다"이다. 그러나 번역으로 "건설하다"를 선택할 경우와 "설립하다"를 선택할 경우에 오류 정도를 동일하게 부여할 수는 없다. 왜냐하면 동사 번역 "건설하다"와 "설립하다"는 "건축하다"와 각각 다른 정도로 유사하게 이해되기 때문이다. 본 논문에서는 이러한 이해도를 번역률에 반영하기 위하여 동사 번역 유사도라는 개념을 새로이 정의하였으며 이를 구하는 방법은 알고리즘 2에 설명된다.

알고리즘 2 유사도 계산 알고리즘

1) 각각의 동사번역 k = 1, ..., m 에 대하여, $c_i = k$ 인 학습예제 $\mathbf{e}_i$ 로 이루어진 벡터 $\mathbf{V}_k$ 를 정의한다.

$$\mathbf{V}_k =< \mathbf{e}_1, ..., \mathbf{e}_{j_k}, ..., \mathbf{e}_{l_k} > \ (l_k \leq n, c_{j_k} = k) \quad (1)$$

여기서 n은 전체 학습 예제의 크기이며 l_k 는 동사번

역 k의 부류에 속하는 학습예제 리스트의 크기이다.

2) 다음과 같은 유사도 행렬 **SM**을 정의한다.

$$\mathbf{SM} = (t_{kk'}) \ (k, k' = 1, \ldots, m)$$

행렬 **SM**의 각 구성원소 $t_{kk'}$ 는 동사번역 k 와 동사번역 k' 의 번역 유사도를 의미하며 $t_{kk'} = t_{k'k}$ 이고 t_{kk} 는 0이다. $k \neq k'$ 인 경우는 다음을 계산한다.

3) 두 동사번역 k 와 k' 에 대하여 두 번역간의 유사도는 다음과 같이 계산된다.

$$t_{kk'} = \frac{\sum_{e_i \in \mathbf{V}_k} \sum_{e_j \in \mathbf{V}_{k'}} D(s_j, s_{j'})}{l_{k'} * l_k * radix}$$

여기서 $\mathbf{V}_k$ 와 $\mathbf{V}_{k'}$은 동사번역이 각각 k 와 k'인 학습예제로 이루어진 벡터이고 l_k 와 $l_{k'}$ 은 각각 이들의 크기이다. 그리고 $radix$는 t_{kk}을 0과 1사이의 값으로 정규화 시키기 위한 상수이다.

위 과정은 모든 예제를 동사번역 부류에 따라서 분류하고, 유사도를 구할 두 부류에 속하는 모든 예제들간의 거리를 모두 합하여 평균을 계산하여 유사도를 구하는 방법을 보여준다. 표 3은 동사 "build"의 동사번역간의 번역 유사도를 보여준다.

표 3 "build" 의 동사 번역 유사도

	건축하다	건설하다	제작하다	설립하다	구축하다
건축하다	0.0	0.434733	0.150255	1.0	0.979905
건설하다	0.434733	0.0	0.486881	0.990209	0.985333
제작하다	0.150255	0.486881	0.0	1.0	0.974383
설립하다	1.0	0.990209	1.0	0.0	0.965919
구축하다	0.979905	0.985333	0.974383	0.965919	0.0

4.4 학습예제의 최적화

이 절은 학습예제를 최적화하는 알고리즘을 설명한다. 예제의 최적화는 크게 두 단계로 구성된다. 1차 최적화에서는 워드넷의 구조를 이용하여 중복예제와 동일 경로상의 비경계(non-boundary) 예제를 제거하며, 2차 최적화에서는 학습예제 중에서 제거 후 번역 오류의 증가가 가장 작은 예제부터 차례로 학습예제 리스트에서 제거한다.

4.4.1 동일 경로상의 비경계 예제의 제거

이 단계에서는 워드넷에서 중복 예제와 동일 경로 예제 중 비경계 예제를 제거한다. 워드넷의 기본 구성요소는 단어가 아닌 신셋이라는 동의어 집합이기 때문에 다른 학습예제(단어)가 동일한 신셋의 원소일 수 있으며 이 때 예제는 알고리즘 3에서 정의된 것과 같이 제거될 수 있다.

알고리즘 3 동일 의미 예제 제거 알고리즘

---

1) $s_i = s_j \ (i < j, fr_i \geq fr_j)$ 을 만족하는 d_{ij} 를 의미거리 행렬 **DM**에서 찾는다.

2) 제거될 예제의 빈도를 남을 예제의 빈도에 더하고 학습예제 리스트에서 제거한다.

$$fr_i = fr_i + fr_j, \ nullptr_j = 1,$$

3) 1)을 만족하는 d_{ij} 가 더 이상 없을 때까지 1), 2)를 반복한다.

알고리즘 3을 통하여 동일한 의미를 가지는 두 학습예제 중 빈도가 더 작은 예제가 학습예제 리스트에서 제거되고 그 예제의 빈도는 나머지 학습 예제의 빈도에 더해진다. 1차 최적화 과정에서 계산되는 예제의 빈도는 2차 최적화 과정에서 제거될 예제를 선택할 때 사용된다.

워드넷은 트리구조이다. 이 트리구조를 이용하여 동일 경로상의 비경계 예제들을 간단하게 구별할 수 있다. 아래 알고리즘 4에서는 동일 경로상의 예제 중 비경계 예제들을 선택하고 이들을 학습예제 리스트로부터 제거하는 과정을 보여준다. l_i 와 l_j 를 신셋 s_i 와 s_j 의 루트로부터의 거리라 하고 p를 s_i 와 s_j 가 최초로 공유하는 상위 계층의 루트로부터의 거리라 할 때, 비경계 예제들을 제거하는 알고리즘은 다음과 같다.

알고리즘 4 비경계 예제 제거 알고리즘

---

1) ($l_i = p$ 이고 $l_j \neq p$) 이거나 ($l_i \neq p$ 이고 $l_j = p$) 인 s_i와 s_j의 의미거리 d_{ij} $(i \neq j)$ 중 최소값을 가지는 d_{ij}를 의미거리 행렬 **DM**으로부터 찾는다.

2) $l_i = p$ 이고 $l_j \neq p$ 인 경우($l_i \neq p$ 이고 $l_j = p$ 인 경우는 아래의 식에서 i와 j를 서로 맞바꾼다),

만일 $c_i = c_j$ 이면, 다음과 같은 연산을 하여 비경계 예제를 학습예제 리스트에서 제거한다.

$$fr_i = fr_i + fr_j,\ nullptr_j = 1,$$

3) 1)의 조건을 만족하는 d_{ij} 가 없을 때까지 1), 2)를 반복한다.

그림 4)의 A)에서는 두개의 부류가 그들의 신셋과 함께 나타나 있다. 신셋 "1.1.4.3.7.9."와 신셋 "1.1.4.3.7.9.12."는 그들이 동일 경로상에서 서로 인접해 있는 동시에 다른 부류에 속해 있기 때문에 경계(boundary) 예제들이다. 그리고 이들 외의 비경계 예제들은 모두 제거되는데 이 결과가 그림 4)의 B)에 나타나 있다. 경계 예제들은 학습예제 리스트에 남아 있게 되며 "center", "hotel", "house", 그리고 "housing"과 같이 제거된 예제들의 빈도는 모두 가장 가까운 경계예제 "building"의 빈도에 더해진다.

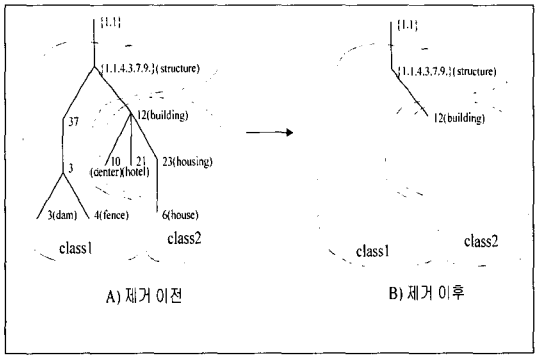

그림 4 비경계 예제들의 제거

4.4.2 최소 오류 예제 제거

1차 최적화가 끝난 후, 더 많은 학습예제를 제거하기 위하여 최소한의 오류를 유발하는 예제부터 학습예제 리스트로부터 제거한다. 2차 최적화 알고리즘은 알고리즘 5와 같이 정의된다.

알고리즘 5 최소 오류 예제 제거 알고리즘

---

1) 학습예제 리스트 $\mathbf{E}_{tr}$ 의 원소를 $\mathbf{e}_i$ ($nullptr_i = 0,\ 1 \le i \le n$) 라 하고 실험예제 리스트 $\mathbf{E}_{ts}$ 의 원소를 $\mathbf{e}_j$ 라고 한다.

1-1) 학습예제 리스트에서 $\mathbf{e}_i$ 가 제거되었다고 가정하고 $\mathbf{E}_{ts}$ 의 원소 $\mathbf{e}_j$ 의 동사번역을 식 (2)을 통해 구하여 k_j 라고 한다.

1-2) 이 때, c_j 와 k_j 간의 동사번역 유사도 t_{cjkj} 를 **SM** 으로부터 가져온다.

1-3) 1-1)와 1-2)을 반복하여 $\mathbf{E}_{ts}$의 모든 원소에 대하여 예제 $\mathbf{e}_i$ 가 제거되었을 때의 유사도의 총합을 다음과 같이 구한다.

$$ES(\mathbf{e}_i) = \sum_{\mathbf{e}_j \in \mathbf{E}_{ts}} t_{c_j k_j}$$

2) 다음을 만족하는 $\mathbf{e}_i$ 를 찾는다.

$$\arg\min_{\mathbf{e}_i}\left[ES(\mathbf{e}_i)\left(\frac{\gamma\, fr_i}{mval} + \beta\right)\right]$$

여기서 β는 베이스 값이고 γ는 값의 간격을 지정해주는 상수이다.
그리고

$$mval = \max_{\mathbf{e}_i} fr_i$$

이다.

3) $\mathbf{E}_{tr}$ 에서 $nullptr_i$ 를 0으로 만들어 학습예제 리스트에서 $\mathbf{e}_i$를 제거한다.

4) 학습예제의 크기가 실시간 기계번역에 적합할 때까지 1)-3)을 반복한다.

위의 과정에서는 학습예제를 하나씩 제거한다고 가정하고, 실험예제들을 번역하여 번역이 잘못되었을 경우, 결과로 나온 번역과 원래 그 예제가 취해야할 번역간의 유사도를 측정하였다. 이 때, 제거되었다고 가정한 각각의 학습예제에 대하여 모든 실험예제의 번역 유사도의 합계가 계산되는데, 이 합계가 가장 작은 학습예제를 선택하여 학습예제 리스트에서 제거한다.

예를 들면, 단어 "workshop(4.1.4.8.1.17)"을 제거하고 실험예제들의 동사번역을 구하면 그때 전체 유사도의 합이 가장 작다. 이런 경우 단어 "workshop"은 학습예제 리스트에서 제거되어 실험예제 리스트로 옮겨진다.

5. 실험 평가

본 논문에서는 타동사 "build"와 그 외의 5개 동사의 번역을 선택하고, 이들의 연어사전을 구축하고 최적화하는 것을 실험 대상으로 삼았다. 본 실험을 위하여 Wall

Street Journal 말뭉치 (220,047문장, 3,080,149 단어)와 경제관련 신문기사로 이루어진 말뭉치(41,750문장, 709,755단어)에서 908개의 build-목적어 쌍을 추출하였으며 그밖의 5개 동사에 대하여 1672개의 동사-목적어 쌍을 추출하였다. 그리고 5-집단 교차 검증(5-fold cross validation)[20]을 위하여 무작위로 전체 샘플을 5개의 샘플 집합으로 나누어 4개의 집합으로 학습 샘플을 구성하고 나머지 하나를 검증 샘플로 사용하였다. "build"의 경우, 샘플1, 샘플2, 샘플3은 182개의 쌍을 샘플4, 샘플5는 181개의 쌍을 가지고 있으며, 다음 표 4는 각 실험별로 샘플에서 추출된 학습예제의 상황을 한 예로 보여주고 있다.

표 4 학습예제의 상황

실 험	(1)	(2)	(3)	(4)	(5)	평 균
(학습 샘플)						
문장	726	726	726	727	727	726.4
학습 예제	241	248	241	246	246	243.2
(검증 샘플)						
문장	182	182	182	181	181	181.6
비워드넷 문장	21	15	16	15	23	18

표 4에서 각각의 실험 (k) 는 5개의 샘플 중 k번째 샘플을 제외한 나머지 샘플들로 학습 예제를 구성하였다. 실험 (1)에서는, 샘플1을 제외한 나머지 4개의 샘플들은 모두 726개의 문장을 가지고 있으며 이로부터 241개의 학습예제가 추출되었다. 그리고 샘플1에 나타난 182개의 예제 문장 중 21개는 목적어가 워드넷에 정의되지 않았다. 그리고 5번의 실험의 평균으로, 학습 샘플은 726.4개의 문장에서 243.2개의 학습예제를 추출하였고 검증 샘플은 181.6개의 문장 중 18개의 문장에 나타난 목적어가 워드넷에 정의되지 않았다.

5.1 동사 "build" 의 번역 선택과 사전 최적화 실험

표 5는 k-최근접 학습을 사용하지 않고 연어 관계만으로 번역을 선택[2]한 결과와 k-최근접 학습을 사용하여 번역을 선택한 결과를 보여주고 동시에 사전을 최적화하는 과정을 보여주고 있다.

표 5를 보면, 연어관계만을 가지고 번역을 선택할 때 (실험 1) 72.9%의 번역률과 0.026초의 번역시간이 나타났다. 그러나 k-최근접 학습을 여기에 추가하여 번역을 선택하면(실험 2) 87.7%의 번역률과 0.096초의 번역시간이 나타났다. 사전의 크기를 줄이는 과정에서 전체 사전의 크기를 초기 사전의 70%로 줄여도 번역률에는 아무런 변화가 보이지 않았으나 번역시간은 그때까지 계속 증가하였다.

표 5 동사 "build"의 번역 선택과 사전 최적화

사전 크기 (%)	실 험	결 과	비 율 (실험/실험1)
연어 관계만을 이용 (실험 1)			
100	번 역 률 (%)	72.9	1.0
	번 역 시 간 (초)	0.026	1.0
연어 관계와 k-최근접 학습을 이용 (실험 2)			
100	번 역 률 (%)	87.7	1.203
	번 역 시 간 (초)	0.096	3.692
90	번 역 률 (%)	87.7	1.203
	번 역 시 간 (초)	0.102	3.923
70	번 역 률 (%)	87.7	1.203
	번 역 시 간 (초)	0.118	4.538
50	번 역 률 (%)	88.4	1.213
	번 역 시 간 (초)	0.116	4.462
30	번 역 률 (%)	88.1	1.209
	번 역 시 간 (초)	0.094	3.615

그리고 사전의 크기를 초기 사전의 50%와 30%까지 줄이면 번역률은 오히려 더 좋아지는 현상을 보였으며, 30%부터는 사전의 크기가 100%일 때보다 번역시간이 감소하는 모습을 보여주었다.

표 6 5개 동사에 대한 번역 결과와 최적화 실험 결과

사전 크기 (%)	실 험	결 과	비 율 (실험/실험1)
연어 관계만을 이용 (실험 1)			
100	번 역 률 (%)	69.5	1.0
	번 역 시 간 (초)	0.030	1.0
연어 관계와 k-최근접 학습을 이용 (실험 2)			
100	번 역 률 (%)	82.1	1.181
	번 역 시 간 (초)	0.102	3.4
90	번 역 률 (%)	82.3	1.184
	번 역 시 간 (초)	0.096	3.2
70	번 역 률 (%)	82.1	1.181
	번 역 시 간 (초)	0.096	3.2
50	번 역 률 (%)	81.8	1.177
	번 역 시 간 (초)	0.087	2.9
30	번 역 률 (%)	79.8	1.148
	번 역 시 간 (초)	0.072	2.4

5.2 그밖의 동사들에 대한 실험

본 논문에서는 동사 "build"외에 말뭉치에서의 빈도가 80회 이상이며 번역 선택시 연어를 이용해야 하는 단어 5개에 대하여 위와 동일한 실험을 하였다. 표 6은 5개 동사에 대한 실험 결과이며, 이 표의 번역률과 번역시간은 5개 동사의 평균값을 나타낸다.

5.3 평가

위의 실험에서는 모두 k-집단 교차 알고리즘을 사용하여 그 평균값을 구하였다. 그리고 본 논문에서 제시된 k-최근접 학습 방법을 통하여 번역 선택을 보다 정확하게 하였다.

동사 "build"의 경우, 번역 선택에서 k-최근접 알고리즘을 사용하여 번역률을 72.92%에서 87.6%으로 향상시켰다(표 5). 그리고 표 6의 실험에서도 번역률은 69.5%에서 82.1%로 향상되었다. 이로써 k-최근접 학습은 번역률의 향상에 도움이 됨을 알 수 있다. 반면에 번역에 소비된 시간은 0.026초에서 0.096초로(표 5) 그리고 0.01초에서 0.034초로 증가하였는데(표 6), 이는 k-최근접 학습에 많은 시간이 소비되기 때문에 발생한 것이다. 그러므로 번역률에는 큰 영향을 미치지 않으면서 사전의 크기를 줄임으로써 번역에 필요한 시간을 감소시켜야 한다.

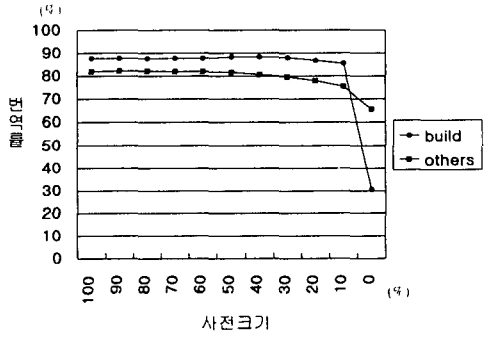

그림 5 사전의 크기와 번역률

그림 5는 사전의 크기와 번역률의 관계를 보여주고 있다. "build"의 경우 사전이 60% 수준을 유지할 때까지는 번역률에 변화가 없었다. 이는 1차 최적화 과정은 번역률에 아무런 영향을 미치지 않는다는 것을 보여주는 것이다. 다른 동사들의 경우에는 아주 미세한 변화를 보여주었는데 이것은 1차 최적화로 제거되는 예제의 비율이 각 동사마다 조금씩 다르기 때문이다. 또한 사전의 크기가 약 40% 가 될 때까지는 예제를 계속 제거하여도 번역률의 하락은 거의 보이지 않았다. 오히려 "build"는 번역률이 약간 좋아지는 모습을 보였는데 이는 2차 최적화로 인하여 번역에 악영향을 미치는 예제들이 제거되었기 때문이다. 다른 동사들 역시 번역률은 크게 떨어지지 않았다. 여기서 우리는 사전이 절반 이하의 크기로 줄어도 번역률은 크게 떨어지지 않는다는 것을 알 수 있었다. 즉 사전을 구성하는 예제들 중에서 실제 번역에 영향을 미치는 예제는 절반 이하인 것이다. 그러나 20% 정도까지 사전을 줄이면 그 후로는 번역률이 급격하게 떨어졌다. 이것은 실제 번역에 큰 영향을 미치는 예제들이 제거되기 때문으로 사전들의 크기를 20%에서 40% 사이로 줄일 때 번역에는 큰 영향을 미치지 않았다.

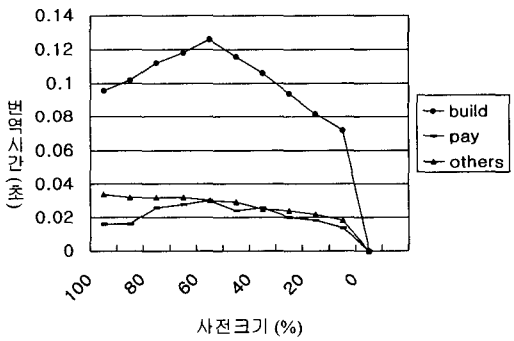

그림 6 학습예제의 크기와 번역시간

그림 6에서는 동사 "build"와 "pay" 그리고 그 외의 동사의 번역시간을 보여준다. "build"와 "pay"는 사전이 60% 가량 될 때까지는 시간이 오히려 증가하는 모습을 보여주었다. 이는 사전 최적화 과정을 계속하면서 사전의 크기가 줄어들어 입력된 목적어를 사전에서 바로 찾지 못하고 k-NN 알고리즘을 적용시켜야 하는 경우가 늘어났기 때문이다. 그러나 5 동사의 평균은 사전이 작아지면서 소비시간 역시 꾸준히 감소하는 모습을 보여주었는데 이는 동사별로 번역시간의 상승과 하강 지점이 모두 다르기 때문에 발생하는 것이다. 전체적으로 각각의 곡선은 서로 다른 위치를 가지는데, 이는 각 동사에 대해 구축된 예제의 수가 서로 다르기 때문이다. "build"와 "pay"동사는 사전의 크기가 30% 수준으로 줄었을 때 번역시간이 최적화 이전보다 감소하는 효과를 보여주었다.

최적화된 동사 연어 사전은 그림 5와 그림 6을 이용하여 번역률, 번역시간, 그리고 사전크기의 범위들이 서

로 만족하는 범위내에서 적정한 구성을 정하게된다.

6. 결 론

본 논문에서는 번역 정확도와 시간 효율을 고려하여 연어 사전을 구축하고 최적화시키는 알고리즘을 제시하였다. 그리고 동사 "build"를 위한 사전을 약 88%의 번역률을 유지시키며 초기 학습예제의 30% 수준으로 최적화시킬 수 있었고 그 외의 5개의 동사를 선택하여 "build"의 경우와 마찬가지로 연어 사전을 구축하고 최적화시켰다.

사전을 최적화함으로써 잡음 효과를 가진 예제들을 제거하여 번역률을 향상시키는 결과를 보였으며 동시에 번역 선택에 필요없는 많은 예제들을 줄임으로써 결과적으로는 번역시간을 줄일 수 있었다.

그러나 본 논문에서 제시된 번역 선택 알고리즘으로 번역률을 더욱 개선시키기 위해서는 몇 가지 문제점이 해결되어야 한다. 첫째, 워드넷은 말뭉치에 나타난 모든 단어와 의미를 포함하고 있지 못하다. 특히 새로운 영역에서 새롭게 사용되는 기술적인 용어들은 워드넷에서 그 의미를 찾을 수 없는 경우가 자주 발생한다. 이 문제를 말뭉치와 사전등을 사용하여 해결하는 알고리즘이 개발되어야 한다. 둘째, 의미거리를 측정할 때 워드넷이 가지고 있는 많은 논리적인 관계들을 통합해서 사용하여야 한다. 마지막으로, 현재의 의미거리 측정 방법은 워드넷 자체의 구조만으로 그 거리를 측정하게 되어 있는데, 이 방법에 말뭉치에 나타난 현상을 반영하는 방법을 연구하여야 한다.

그리고 본 논문에서 제시된 사전 구성 방법은 사람의 개입이 높은 비중을 차지한다는 문제를 가지고 있다. 이 문제를 해결하기 위하여 말뭉치, 전자사전, 시소러스 등의 여러 자원들을 활용하여 최대한 자동으로 연어 사전을 구성하는 방법에 대한 연구가 필요하다.

본 논문에서 제시된 최적화 알고리즘은 다른 여러 타동사와 자동사, 부사어와 전치사의 보다 정확한 번역과 좋은 효율을 위하여 응용될 수 있을 것이다.

참 고 문 헌

[1] Dagan I. and A. Itai, "Word Sense Disambiguation Using a Second Language Monolingual Corpus," *Association for Computational Linguistics*, Vol. 20, No. 4, pp. 563-596, 1994.

[2] Kim N. and Y. T. Kim, "Determining Target Expression Using Parameterized Collocations from Corpus in Korean-English Machine Translation," *Proc. of PRICAI-94*, pp. 732-736, 1994.

[3] Dagan I., F. C. N. Pereira and L. Lee, "Similarity-Based Estimation of Word Cooccurrence Probabilities," *32nd Annual Meeting of ACL*, 1994

[4] Dagan I., L. Lee, and F. C. N. Pereira, "Similarity-Based Models of Word Cooccurrence Probabilities," *Machine Learning*, Vol. 34, pp. 43-69, 1999.

[5] Karov, Y. and S. Edelman, "Similarity-based Word Sense Disambiguation," *Computational Linguistics*, Vol. 24, No. 1, pp. 41-59, 1998.

[6] Resnik, P., "Disambiguating noun groupings with respect to WordNet senses," *Proc. of the Third Workshop on Very Large Corpora*, pp. 54-68, 1995.

[7] Yarowsky D., "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora," *Proc. of COLING-92*, Nantes, Aug 23-28, pp.454-460, 1992

[8] Charniak, E., *Statistical Language Learning*, pp. 135-145, The MIT Press, 1993.

[9] 박성배, 장병탁, 김영택, "Self-Organizing Map을 이용한 한국어 동사 클러스터링," *98 가을 한국정보과학회 학술발표논문집(II)*, pp. 183-185, 1998.

[10] Brown P. F., V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-Based n-gram Models of Natural Language," *Association for Computational Linguistics*, Vol. 18, No. 4, pp. 467-479, 1992.

[11] Mitchell, T. M., *Machine Learning*, pp. 230-236, The McGraw-Hill Companies, Inc., 1997.

[12] Cover D.S. & P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, Vol. 13, pp21-27, 1967

[13] Duda R. & P. Hart, *Pattern Classification and scene analysis*, New York: John Wiley & Sons, 1973

[14] Bishop C. M., *Neural networks for pattern recognition*, Oxford, England: Oxford University Press., 1995

[15] Frey, B. J., *Graphical Models for Machine Learning and Digital Communication*, pp. 55-57, The MIT Press, 1998.

[16] *Collins Cobuild English Language Dictionary*, 1997.

[17] Kim Y. and Y. T. Kim, "Semantic Implementation based on Extended Idiom for English to Korean Machine Translation," *The Asia-Pacific Association for Machine Translation Journal*, No.21, pp. 23-39, 1998.

[18] 김유섭, 김영택, "영한 기계번역에서 관용구에 기반한 의미분석", *정보과학회논문지(B)*, 제25권, 제4호, pp. 609-617, 1998.

[19] Richardson R., A. F. Smeaton & J. Murphy, "Using WordNet as a Knowledge Base for Measuring Semantic Similarity between Words," *School of Computer Applications Working Paper*: CA-1294, 1994.

[20] Fellbaum, C., *Wordnet · An Electronic Lexical Database*, The MIT Press, 1998.
[21] Cherkassky V., and F. Mulier, *Learning from Data Concepts, Theory, and Methods*, pp78-80, John Wiley & Sons, Inc., 1998.

김 유 섭
1992년 서강대학교 전자계산학과 학사. 1994년 서울대학교 컴퓨터공학과 석사. 2000년 서울대학교 컴퓨터 공학과 박사. 2000년 ~ 현재 서울대학교 컴퓨터신기술연구소 연구원. 관심분야는 자연언어처리, 의미분석, 기계번역, 기계학습 등임.

장 병 탁
1986년 서울대학교 컴퓨터공학과 학사. 1988년 서울대학교 컴퓨터공학과 석사. 1992년 독일 Bonn 대학교 컴퓨터과학과 박사. 1988년 ~ 1992년 Bonn 대학교 AI Lab. 연구원. 1992년 ~ 1995년 독일 국립전산학연구소(GMD) 연구원. 1995년 ~ 1997년 건국대학교 컴퓨터공학과 조교수. 1997년 ~ 현재 서울대학교 컴퓨터공학과 조교수. 관심분야는 인공지능, 기계학습, 신경망, 진화 알고리즘

김 영 택
1963년 미국 Colorado대 전기과 석사학위 취득. 1968년 미국 Utah대 전산과공학 박사학위 취득. 1975년 서울대학교 전자계산소 설치(소장 역임). 알골 컴파일러 완성. 1979년 ~ 1981년 미 purdue 대, Yale대, Ilinois대 객원교수. 1981년 한국 정보과학회 회장, 1990년 한국 인지과학회 회장. 현재 서울대학교 컴퓨터공학과 교수로 재직중. 관심분야는 프로그래밍 언어, 자연언어 처리임