

문장구조 유사도와 단어 유사도를 이용한 클러스터링

기반의 통계기계번역

김한경⁰¹, 나휘동², 이금희², 이종혁²

포항공과대학교 정보통신대학원 정보처리학과¹, 포항공과대학교 컴퓨터공학과²

{arch, leona, ljj, jhlee}@postech.ac.kr

Integrated Clustering Method based on Syntactic Structure and Word

Similarity for Statistical Machine Translation

Hankyong kim⁰¹, Hwi-Dong Na², Jin-Ji Li², Jong-Hyeok Lee²

Dept. of Graduate School for Information Technology, POSTECH, Korea¹

Department of Computer Science and Engineering POSTECH, Korea²

요 약

통계기계번역에서 도메인에 특화된 번역을 시도하여 성능향상을 얻는 방법이 있다. 이를 위하여 문장의 유형이나 장르에 따라 클러스터링을 수행한다. 그러나 기존의 연구 중 문장의 유형 정보와 장르에 따른 정보를 동시에 사용한 경우는 없었다. 본 논문에서는 문장 사이의 문법적 구조 유사성으로 문장을 유형별로 분류하는 새로운 기법을 제시하였고, 단어 유사도 정보로 문서의 장르를 구분하여 기존의 두 기법을 통합하였다. 이렇게 분류된 말뭉치에서 추출한 모델과 전체 말뭉치에서 추출된 모델에서 보간법(interpolation)을 사용하여 통계기계번역의 성능을 향상하였다. 문장구조의 유사성과 단어 유사도 계산을 위하여 각각 커널과 코사인 유사도를 적용하였으며, 두 유사도를 적용하여 말뭉치를 분류하는 과정은 K-Means 알고리즘과 유사한 기계학습 기법을 사용하였다. 이를 일본어-영어의 특허문서에서 실험한 결과 최선의 경우 약 2.5%의 상대적인 성능 향상을 얻었다.

주제어: 통계기계번역, 클러스터링, 도메인 특화 모델, 문장구조 유사도, 단어 유사도

1. 서론

통계기계번역(statistical machine translation)은 언어모델, 번역모델의 확률모델을 사용하여 번역을 수행한다. 이 번역 성능을 높이기 위한 방법의 하나로 도메인에 특화된 모델을 추가로 사용하는 연구가 진행되어 왔다. 도메인에 특화된 모델의 추가사용은 보간법을 사용한다. 보간법을 모델에 적용하는 방법은 전체 말뭉치에서 획득한 모델의 각 확률을 특화된 모델의 해당 값을 참조하여 수정하는 것이다. 이를 사용한 기존연구들은 말뭉치를 분류하는 기준에 따라 문장의 유형에 따라 나누는 방법과 문서의 장르에 따라 나누는 방법의 두 가지로 정리할 수 있다. 본 논문에서는 각각 고유한 장점을 가진 두 방법을 통합하여 말뭉치를 분류하고 번역 성능을 높이는 실험을 수행하였다.

통계기계번역에서 흔히 사용하는 언어모델은 n-gram 어순에 따라 계산되며, 이는 의문문, 평서문 등 문장유형에 따라서 다른 결과를 가진다. 따라서 문장유형에 따라 말뭉치를 분류하게 되면 각 유형별 어순의 특성이 언어모델에 반영된다. 한편 문서의 장르에 따라서 원시언어와 대상언어 사이의 대역어가 달라질 수 있다 따라서 장르에 따라 분류된 말뭉치의 번역모델은 대역어의 확률이 도메인에 특화된 값을 가지게 된다.

본 논문에서는 문법적 구조의 유사도를 계산하여 문장

의 유형을 판단하는 방법을 제안하였고 단어 유사도를 계산하여 장르를 추정하는 방법을 통합하여 말뭉치를 분류하였다. 분류방법은 비지도 기계학습(unsupervised machine learning) 기법을 사용하였다. 따라서 문장유형의 구분에서 말뭉치의 특성에 관계없이 일반적으로 적용이 가능하며 그리고 장르별로 나누어져 있지 않은 말뭉치에도 적용이 가능하다.

본 논문에서는 말뭉치의 분류작업 후, 각 클러스터에서 얻어진 도메인 특화 모델을 통계기계번역 시스템에 추가로 사용하였다. 이 시스템에서는 선형보간법으로 전체 훈련 말뭉치에서 획득한 모델과 도메인 특화 모델을 통합하였다. 베이스라인 시스템은 전체 훈련 말뭉치에서 추출한 언어모델과 번역모델만을 사용하였다 제안한 방법의 성능을 확인하기 위해서 일본어-영어 특허번역문서에서 베이스라인의 번역결과와 도메인 특화 모델을 추가로 사용한 번역결과를 비교하였다

2. 기존연구

도메인에 특화된 번역을 시도한 기존연구들에서는 훈련 말뭉치를 분류하는 기준으로 의문문, 평서문 등의 문장유형을 이용하거나, 뉴스, 비평 등 장르를 사용하였다. 문장유형에 따라 분류한 기존연구에서는 문장유형을 특정하기 위해서 말뭉치의 특성에 의존적인 방법을 사용하

거나 특정 유형에만 적용이 가능한 단점이 있다 이러한 문제를 해결하기 위한 시도에서는 문장유형에 따른 분류 과정에서 문법적 정보가 사용되지 않았다 한편, 장르에 따라 훈련말뭉치를 분류한 기존연구에서는 훈련말뭉치가 사전에 장르별로 나누어져 있어야 적용 가능한 기법이거나, 번역할 원시언어 문장의 적합한 장르를 판별하는 도메인 예측과정에 문제가 있었다

말뭉치를 문장유형에 따라 분류한 기법의 연구들은 문장유형에 따른 어순의 특성을 언어모델에 반영하며 이를 번역 성능의 향상에 이용한다 Hasan등[1]은 훈련말뭉치를 정규식을 이용하여 의문문, 명령문, 단순 나열과 같은 10여개의 클러스터로 분류였다. 그리고 이들에서 도메인 특화된 언어모델을 획득하여 보간법을 적용한 결과 상당한 복잡도의 감소를 얻었으며, 번역 성능 또한 향상되었다. 하지만 이 실험에서 사용한 정규식은 훈련말뭉치의 일부에만 적용되며, 말뭉치의 특성에 따른 휴리스틱을 사용하기에 일반적으로 확장하기 곤란하다

Yamamoto등[2]은 간접적인 접근방법을 사용하여 이러한 문제를 해결하였다 이들은 훈련말뭉치를 임의로 나누고 각 유형별 말뭉치에 대하여 해당 언어모델로 계산한 엔트로피의 값이 낮아지도록 문장의 재배치를 반복하는 기계학습 기법을 사용하였다 이 방법은 정규식을 적용하는 것과는 달리 말뭉치의 특성에 대하여 독립적이며, 모든 문장을 분류할 수 있는 장점이 있다 하지만 분류 과정에서 어떠한 문법적인 정보도 사용되지 않기 때문에 정확히 문장구조에 따른 분류가 이루어진다고 확신할 수 없다.

한편 Finch등[3]은 확률적 분류기를 사용하여 훈련말뭉치를 의문문과 평서문 두 부류로 나누어 도메인 특화 모델을 획득하고, 분류기의 확률을 사용하는 동적 보간법을 제안하였다. 이는 번역할 대상 원시언어 문장이 각각의 도메인에 속할 확률을 해당 도메인 모델이 보간법에서 가지는 비율로 사용하는 기법이다 하지만 이 또한 분류기가 $n \leq 3$ 인 n-gram정보만을 사용하기 때문에 문장의 문법적 구조 정보를 사용하는 데 한계가 있다

번역 성능의 향상을 위해서 도메인 특화 모델을 사용한 연구의 다른 방향으로는 훈련말뭉치를 문서의 장르에 따라 나누는 기법이 있다. 이는 주로 훈련말뭉치가 여러 장르의 문서로 이루어진 경우, 각각에 특화된 번역을 수행하여 그 성능을 향상시키기 위해서 사용한다 Yasuda등[4]은 특허번역 말뭉치를 국제특허분류(International patent classification, IPC)에 따라 분류하고 각 부영역에서 획득한 도메인 특화 모델을 보간법에 사용하였다

Ito등[5]은 동일한 특허번역 말뭉치에서 정보검색 기법을 사용하는 분류기(CLUTO, Karypis[6])를 사용하였다. 이 분류기는 코사인 계수를 이용하여 단어 유사도를 측정하여 분류하였다. 이들은 도메인 예측 과정에서 어려움을 겪었지만, 각 문장에 가장 알맞은 도메인 특화 모델을 사람이 직접 결정하여 적용한 실험에서는 상당한 번역 성능의 향상을 얻었다.

Foster등[7]은 비평, 연설, 방송, 뉴스 등 여러 장르로 이루어진 말뭉치에서 다양한 유사도 측정기법을 사용하였으며 선형보간법과 지수선형보간법의 결과를 비교하였다. 이들은 다양한 방법을 사용하여 각 장르별 모델을

조합하여 훈련말뭉치에 해당하지 않는 장르의 문서를 번역하는 효과적인 방법을 제안하였다 그러나 이러한 방법을 사용하기 위해서는 훈련말뭉치가 사전에 여러 장르로 나누어져 있는 상태여야 하는 단점이 있다.

3. 방법론 개요

본 논문에서 제안하는 방법은 크게 두 단계로 이루어진다. 하나는 훈련말뭉치를 유형별로 분류하는 단계이다. 이를 위하여 문장의 문법적 구조 유사도를 의존관계 트리에서 커널 함수를 사용하여 계산하였고, 단어 유사도를 코사인 유사도로 계산하였다. 훈련말뭉치의 분류는 K-Means와 유사한 비지도 기계학습 알고리즘으로 수행한다. 이 알고리즘에서 사용하는 거리함수는 문장구조 유사도와 단어유사도를 일정한 비율로 합산하여 계산한다.

다음은 유형별로 특화된 번역을 수행하는 단계이다 이는 각각의 도메인 특화 모델의 획득과 이를 추가로 사용하는 통계기계번역 시스템의 구축 그리고 번역할 원시언어 문장의 도메인을 예측하여 해당 도메인에 특화된 시스템에서 번역을 수행하는 것으로 이루어진다

아래에서는 각 단계별로 착안점과 수행 방안을 설명하였다.

3.1 문장구조 유사도의 계산

두 원시언어 문장 사이의 문장구조 유사도는 일본어의 의존관계 트리에서 커널 함수를 사용하여 계산한 값을 사용하였다. 본 논문에서 사용한 커널 함수는 Collins등[8]이 제안한 의존관계 트리 사이에서 유사도를 측정하는 기법으로, 이를 사용하여 문장구조 유사도를 계산하는 과정에서 일본어 의존관계 트리의 특성에 의해 일부 수정된 사항이 있어 다음과 같이 정리하였다

3.1.1 일본어 의존관계 트리

본 논문에서는 일본어 의존관계 파서인 CaboCha[9]를 사용하여 의존관계 트리를 획득하였다 이 의존관계 트리는 일본어의 의미단위인 문절(bunsetsu)을 단위로 사용한다. 또한 트리의 각 문절 사이에 지배소의존소 관계만이 표시되고 관계표지가 나타나지 않는다 그리고 각 문절은 형태소분석 결과와 형태소의 품사가 포함되어 있으며 어간형태소도 표시되어 있다 별도로 계산할 단어 유사도와 중복을 피하고 문절에서 문장의 구조를 나타내는 정보만을 추출하기 위해서 첫 어간형태소의 품사와 이후의 형태소를 연결한 것을 문절의 품사로 정의하였고, 그 예는 그림 1에 정리한 바와 같다

문절의 품사를 이와 같이 정의하였을 때 첫 번째 어간형태소의 품사는 문절의 문법적 특성을 추정하는 데 사용된다. 이후의 형태소들은 지배소의존소 사이의 관계표지를 대신하여 해당 문절과 다른 문절이 서로 어떤 관계로 연결되었는지 추론하기 위한 자료이다

이후의 실험에서는 일본어 의존관계 트리에서 지배소의존소 관계와 본 논문에서 정의한 문절의 품사를 문장구조의 유사도를 계산할 정보로 사용한다

3.1.2 의존관계 트리에서 커널 함수의 계산

상기한 바와 같이, 본 논문에서는 Collins등[8]이 제안한 “A Kernel for Dependency Structure”를 적용하여 문장구조의 유사도를 계산하였다 그러나 커널 함수를 계산할 때 사용할 정보가 Collins등[9]의 논문에서 예제로 사용된 영어의 의존관계 트리와는 다르기에 관계표지 정보의 사용을 삭제하였고 각 노드의 형태소를 사용한 부분을 폼사를 사용하도록 수정하였다

또한 이들이 제안한 커널 함수는 두 트리에서 공통으로 가지는 서브트리의 수를 이용하여 계산한다 그러나 공통의존소가 없을 때 공통서브트리의 수가 -1로 계산되는 오류가 있어 그림 2와 같이 정상적인 값을 가지도록 수정하였다.

형태소	POS(형태소)	POS(문절)
その他	名詞-代名詞-一般 記号-読点	'名詞-代名詞-一般'+','
ウレタン 樹脂 等 が	名詞-一般 名詞-一般 名詞-接尾-一般 助詞-格助詞-一般	'名詞-接尾-一般'+ 'が'
保護 層 15 の	名詞-サ変接続 名詞-接尾-一般 名詞-数 助詞-連体化	'名詞-数'+ 'の'
材質 として	名詞-一般 助詞-格助詞-連語	'名詞-一般'+ 'として'
用い られる 。	動詞-自立 動詞-接尾 記号-句点	'動詞-自立'+ 'られる'

그림 1. 일본어 문절의 폼사 추출 예

두 의존관계 트리 d_1, d_2 의 커널 함수 $K(d_1, d_2)$ 는 각 트리에서 노드의 집합이 N_1, N_2 이고, 그 원소인 문절 n_1, n_2 가 가지는 공통서브트리의 수를 $C_m(n_1, n_2)$ 라 정의할 때 아래와 같이 계산된다.

$$K(d_1, d_2) = \sum_{n_1 \in N_1, n_2 \in N_2} C_m(n_1, n_2) \quad (1)$$

이 때 $C_m(n_1, n_2)$ 은 다음의 식을 만족하는 n_1, n_2 의 공통 의존소 $\text{sim}(n_1, n_2)$ 을 사용하여 계산할 수 있다.

$$\text{sim}(n_1, n_2) = \left\{ (x, y) \mid \begin{array}{l} x \in \text{children}(n_1), \\ y \in \text{children}(n_2), \\ \text{POS}(x) = \text{POS}(y) \end{array} \right\} \quad (2)$$

이 $\text{sim}(n_1, n_2)$ 에서 $C_m(n_1, n_2)$ 은 조건에 따라서

if $\text{POS}(n_1) \neq \text{POS}(n_2)$ or $\text{children}(n_1) = \emptyset$ or $\text{children}(n_2) = \emptyset$ or $\text{sim}(n_1, n_2) = \emptyset$

then $C_m(n_1, n_2) = 0$

$$\text{else } C_m(n_1, n_2) = \prod_{(x,y) \in \text{sim}(n_1, n_2)} (C_m(n_1, n_2) + 2) - 1 \quad (3)$$

위의 식으로 계산된 값을 가진다.

식 (2)에서 계산한 커널 함수의 값은 두 의존관계 트리에서 공통으로 가지는 서브트리의 수가 일치하면 각 트리에서 일치하지 않는 부분의 크기와 관계없이 일정한 값을 가진다. 따라서 문장구조의 유사도를 올바르게 나타내기 위해서 이 값을 두 트리의 크기로 정규화 하여 사용하며, 이 문장구조 유사도의 값은 다음과 같다

$$\text{Structure}(d_1, d_2) = \frac{K(d_1, d_2)}{|N_1| + |N_2|} \quad (4)$$

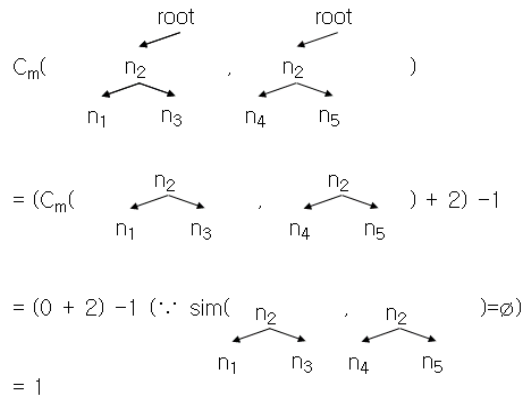


그림 2. 수정된 $C_m(n_1, n_2)$ 이 적용되는 예

3.2 단어 유사도의 계산

두 원시언어 문장 사이의 단어 유사도를 계산하기 위해서 CaboCha[8]의 일본어 문장 구문분석 결과에서 형태소 분석 정보를 추출하여 사용하였다 계산 방법으로는 코사인 유사도의 계산 기법을 적용하였다 코사인 유사도는 두 벡터의 내적 값으로 유사도를 계산하므로 각 문장의 형태소 분석 정보를 가공하여 1-gram 단어 빈도수의 벡터로 변환하여 코사인 유사도를 계산하였다

즉 두 문장 S_1, S_2 에서 이들에 포함된 형태소를 bag of word모델로 사용하여 벡터의 차원을 결정하고 각 문장에서 형태소의 빈도수를 해당 차원에 표시하여 벡터로 변환한다. 이러한 방법을 사용하여 두 문장 S_1, S_2 은 각각 단어 빈도수의 벡터 V_1, V_2 로 표현할 수 있다. 따라서 두 문장 S_1, S_2 에서 코사인 유사도의 값은 아래의 식에 대입하여 얻을 수 있다.

$$\text{Word}(S_1, S_2) = \cos(\theta) = \frac{V_1 \cdot V_2}{|V_1| \times |V_2|} \quad (5)$$

이때 각 벡터를 구성하는 단어 빈도수는 형태소의 출현 빈도를 나타내는 0 이상의 정수이므로 $\text{Word}(S_1, S_2)$ 의 값은 항상 $0 \leq \text{Word}(S_1, S_2) \leq 1$ 을 만족한다.

3.3 분류 알고리즘

분류 알고리즘에서 사용할 거리함수(distance function)는 위에서 정의한 $Structure(d_1, d_2)$ 와 $Word(S_1, S_2)$ 를 다음과 같이 일정한 비율로 더하여 계산한다

$$D(S_1, S_2) = \alpha Structure(d_1, d_2) + (1 - \alpha) Word(S_1, S_2) \quad (6)$$

따라서 이 거리함수의 값이 클수록 두 문장은 서로 유사도가 높다. 아래에서는 이 거리함수를 사용한 기본 분류 알고리즘과 분류기, 분류기를 추가하여 처리속도를 향상한 개선된 알고리즘을 제안하였다

3.3.1 기본 분류 알고리즘

기본 분류 알고리즘은 Yamamoto등[2]이 사용한 방법이나 K-Means 알고리즘과 유사하며 훈련말뭉치를 사용자가 지정한 수만큼의 클러스터로 분류한다 이는 비지도 기계학습으로, 분류할 클러스터의 수와 거리함수에서 사용할 유사도 간의 비중을 제외하면 모든 정보를 말뭉치에서 학습한다. 이때 거리함수는 두 문장 사이의 쌍대 비교(pairwise comparison)만이 가능하므로 문장과 클러스터 간의 거리는 아래의 식과 같이 대상 문장과 해당 도메인에 속한 모든 문장과의 거리의 평균을 사용한다

$$D(S, C) = \frac{\sum_{S_i \in C} D(S, S_i)}{|C|} \quad (7)$$

기본 분류 알고리즘의 동작은 다음과 같다

단계 1: 훈련말뭉치의 모든 문장 쌍에 대하여 거리함수를 계산하고 그 값을 저장한다

단계 2: 훈련말뭉치의 문장을 사용자에게 의해 정해진 수의 클러스터로 임의로 배정한다

단계 3: 훈련말뭉치의 각 문장에 대하여, 해당 문장과 모든 클러스터 사이의 거리함수 값을 계산한다

단계 4: 각 문장을 $c^* = \operatorname{argmax}_{c \in C} D(S, c)$ 와 같이 해당 문장과 거리함수의 값이 가장 큰 클러스터로 재배치한다.

단계 5: 단계 4에서 기존 소속과 다른 클러스터로 재배치되는 문장의 수가 일정 이하가 될 때 까지 단계 3과 단계 4를 반복한다.

3.3.2 분류기

기본 분류 알고리즘에서 사용한 거리함수를 응용하여 대상 원시언어 문장이 훈련말뭉치의 클러스터 중 어느 클러스터와 가장 유사도가 높은지 판별하는 분류기를 만들 수 있다. 이는 대상 원시언어 문장에 대하여 기본 분류 알고리즘의 단계 3과 단계 4를 한 번씩 수행하는 것으로 구현한다.

3.3.3 개선된 분류 알고리즘

기본 분류 알고리즘에서는 훈련말뭉치에 속한 모든 문장 쌍에 대한 거리함수를 계산한다 복잡도로 표현하면 문장수 n 에 대하여 $O(n^2)$ 과 같다. 본 논문에서는 기존 분류 알고리즘의 시간 복잡도를 개선하기 위해 분류기를 이용한 개선된 분류 알고리즘을 제안한다 이 알고리즘에서는 일정 이상의 문장이 확보되면 분류 알고리즘의 분류 정확도를 신뢰할 수 있다고 가정하고, 훈련말뭉치에서 최소한의 문장들을 시드로 선택하여 기본 분류 알고리즘을 수행하고 나머지 문장들은 분류기를 이용하여 각 문장에 가장 알맞은 클러스터로 배치한다. 이 개선된 분류 알고리즘의 동작은 다음과 같이 정리할 수 있다

단계 1: 훈련말뭉치에서 충분한 양의 문장을 시드 말뭉치로 선택하여 기본 분류 알고리즘을 수행한다

단계 2: 훈련말뭉치에서 시드로 선택되지 않은 여분의 문장들과 선택된 문장들 간의 거리함수를 계산한다

단계 3: 단계 1에서 실행한 기본 분류 알고리즘이 완료되면 분류되지 않은 여분의 문장들을 분류기를 사용하여 적합한 클러스터에 배치한다

단계 4: 기본 분류 알고리즘으로 나누어진 클러스터와 단계 3에서 배치된 문장들을 통합하여 클러스터를 재구성한다.

이 개선된 분류 알고리즘의 복잡도는 시드로 선택된 부분의 문장을 x 라 하면 기본 알고리즘 부분에서 $O(x^2)$, 나머지 문장을 분류하는 과정에서 $O(xn)$ 으로 알고리즘 전체에서는 $O(x^2) + O(xn)$ 이 된다. 따라서 훈련말뭉치에서 시드로 선택하는 부분이 $x \ll n$ 일 경우 실질적인 복잡도는 $O(n)$ 으로 감소된다.

3.4 도메인 특화 번역

훈련말뭉치의 분류가 완료되면 각 클러스터에서 추출된 도메인 특화 모델을 추가로 사용하는 번역시스템을 구축한다. 이 시스템은 그림 3과 같이 선형보간법을 적용하

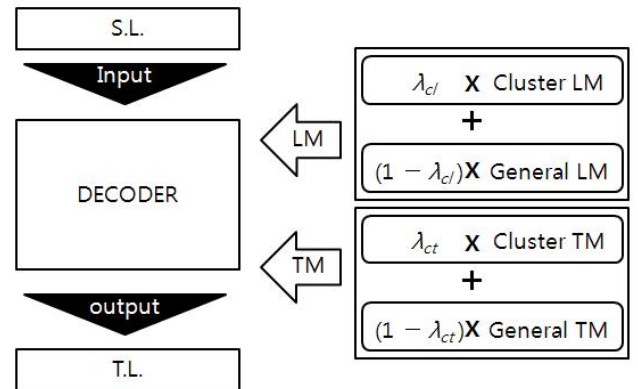


그림 3. 도메인 특화 모델을 사용하는 시스템 개괄

여 도메인에 특화된 모델과 전체 모델을 함께 사용한다. 이때 보간법의 적용에서 각 모델에 사용한 비율은 최선의 성능을 발휘하도록 임의로 조정한다.

다음으로는 번역할 원시언어 문장마다 각각에 적합한 도메인을 예측하는 작업을 수행한다. 이는 분류기를 사용하여 해당 문장이 훈련말뭉치의 클러스터 중 어디에 가장 유사한지 판단하는 것으로 수행한다.

4. 실험

본 논문에서는 NTCIR-7 특허번역 말뭉치[10]에서 일본어-영어의 번역 방향을 선택하여 실험을 수행하였다. 언어모델과 번역모델은 SRILM[11]과 GIZA++[12]를 이용하여 획득하였다. 베이스라인 번역시스템의 구축은 통계기계번역 도구 Moses[13]를 Och등[14]이 제안한 방법에 따라 환경변수를 보정하여 사용하였다. 번역성능은 BLEU[15]를 사용하여 비교하였다.

4.1 말뭉치

실험에 사용한 말뭉치는 NTCIR-7 특허번역작업[10] 말뭉치이다. 이는 실질적으로 평서문만으로 구성되어 있으며, 문장의 길이가 긴 편이다.

이 말뭉치에서 편의상 10만 문장을 임의로 선택하여 훈련말뭉치로 사용하고 번역도구의 환경변수를 보정하기 위해서 609문장을, 실제 번역할 대상으로는 1381개 문장을 사용하였다. 각 말뭉치의 자세한 정보는 표 1에 나타난 바와 같다.

표 1. 말뭉치 정보

말뭉치	크기		
	문장	단어	
		일본어	영어
Training	100,000	2,660,659	2,445,265
Dev.	609	16,230	14,818
Test	1,381	48,930	44,910

표 2. 분류 알고리즘 실험 결과

비율	기본 알고리즘 분류결과			
	1	2	3	4
0:10	34,736	31,480	22,675	11,109
3:7	33,382	26,585	26,367	13,666
5:5	31,830	30,048	21,658	16,464
7:3	40,161	23,152	19,352	17,335
10:0	47,377	20,794	18,763	13,066
개선된 알고리즘 분류결과				
0:10	41,531	29,043	22,651	6,775
3:7	33,098	25,091	23,815	17,996
5:5	29,574	25,663	24,872	19,891
7:3	32,772	27,300	23,376	16,552
10:0	40,795	23,549	19,175	16,481

4.2 분류 및 번역 성능 비교

본 논문에서는 훈련말뭉치의 분류 방법으로 기본 분류 알고리즘과 개선된 분류 알고리즘의 두 가지 방법을 제안하고 있다. 이들의 성능확인을 위하여 훈련말뭉치를 4개의 클러스터로 분류하고, 그 결과에서 도메인 특화 번역을 실행하여 번역 성능을 비교하였다. 기본 알고리즘에서는 훈련말뭉치 전체를 분류하였고, 개선된 알고리즘에서는 전체의 20%를 시드로 사용하여 분류하고 나머지는 분류기로 배치하였다. 또한 두 알고리즘 모두 거리함수의 문장구조 유사도와 단어 유사도간 비율을 0:10에서 10:0까지 5가지 경우로 달리하여 실험하였으며 이 결과를 표 2에서 정리하였다.

두 알고리즘에서 분류된 훈련말뭉치를 클러스터로 사용하여 도메인에 특화된 번역시스템을 구축하고 분류기를 사용한 도메인 예측으로 테스트말뭉치의 문장들을 각각 최적의 시스템에서 번역하였다. 각 알고리즘을 적용하였을 때의 성능은 기본 분류 알고리즘을 사용한 결과를 표 3에, 개선된 분류 알고리즘을 사용한 결과를 표 4에 정리하였다.

훈련말뭉치의 분류 결과에서는 직접적인 번역 성능을 알 수 없으므로 표 3과 표 4에서 번역 성능을 비교하면 기본 알고리즘을 사용한 경우는 최대 약 0.5포인트, 개선된 알고리즘에서는 약 0.6포인트의 성능향상이 있었다. 베이스라인의 번역 결과와 비교할 때 상대적으로 2%, 2.5%가량의 성능 개선으로, 각각 신뢰도 95%에서 유의성이 검증된 값이다. 이는 분류과정에 사용한 알고리즘에 따른 번역 성능의 차이가 크지 않으며 오히려 개선된 알고리즘을 사용하였을 때의 성능 향상이 더 우수할 수도 있다는 것을 보여준다. 따라서 알고리즘의 분류 성능을 번역 성능과 분류속도를 기준으로 판단하면 개선된 알고리즘이 더 뛰어난 것을 알 수 있다.

표 3. 기본 알고리즘의 번역 성능

비율	보간 대상		
	LM	TM	Both
베이스라인	24.17		
0:10	24.25	24.46	24.54
3:7	24.36	24.23	24.60
5:5	24.44	24.46	24.62
7:3	24.44	24.44	24.67
10:0	24.46	24.39	24.58

표 4. 개선된 알고리즘의 번역 성능

비율	보간 대상		
	LM	TM	Both
베이스라인	24.17		
0:10	24.34	24.39	24.60
3:7	24.38	24.50	24.76
5:5	24.35	24.55	24.62
7:3	24.48	24.45	24.63
10:0	24.34	24.28	24.47

분류 알고리즘의 거리함수에서 문장구조 유사도와 단어 유사도의 비율을 달리한 결과를 확인하면 본 논문에서 제안한 방법을 기존연구들과 비교할 수 있다 하나의 유사도만을 사용한 결과들을 기존연구와 동일하다고 고려할 때 두 유사도를 조합하여 말뭉치를 분류한 경우의 번역 성능이 더 뛰어나다. 즉 본 논문에서 제안하는 두 유사도를 동시에 사용하는 분류기법이 기존연구에 비하여 번역 성능 향상에 우수하다고 할 수 있다.

또한 보간법을 사용한 대상 모델에 따른 변화를 관찰하면, 표 3의 번역 성능에서 언어모델만을 보간법의 대상으로 사용하였을 때의 번역 성능이 거리함수에서 문장구조 유사도의 비율이 높아질수록 향상되는 것을 볼 수 있다. 이것은 문장구조의 유사도가 언어모델의 성능 향상에 기여하는 것의 증거로 볼 수 있다. 거리함수에서 두 유사도의 비중을 5:5로 동일하게 사용하였을 때는 번역 모델을 사용한 성능이 언어모델을 사용한 성능보다 우수하다. 이것은 말뭉치에서 문장유형의 변동이 거의 없기 때문이라 볼 수 있다. 그리고 모든 실험 결과에서 언어 모델이나 번역모델 단독으로 보간법을 사용한 것 보다는 둘 다 사용한 성능이 우수하였다.

5. 결론

본 논문에서는 문장구조와 단어의 유사도를 사용하여 말뭉치를 분류하고, 각 클러스터에 대하여 특화된 번역을 수행하는 방안을 제시하였다 이는 말뭉치의 특성에 의존적이지 않은 비지도 기계학습으로 실험결과 그 성능 향상을 확인하였다. 또한 문장유형의 구분을 말뭉치의 특성에 의존적이지 않게 수행할 수 있었던 점과 번역할 문장과 가장 가까운 클러스터를 확인하는 도메인 예측에서 각각 기존의 연구에서 보였던 단점을 해결하였다 그리고 분류과정에서 계산시간의 소요를 경감하기 위해서 개선된 분류 알고리즘을 제안하고 그 성능 또한 확인하였다.

향후 연구 사항으로는 분류과정에서 유사도 계산법을 개선하여 쌍대비교를 회피하는 방안, 훈련말뭉치의 크기에 따른 차이의 확인, 다른 언어 쌍에 본 방법론을 적용하였을 때 결과의 확인 등이 있다. 번역과정에서는 보간법 적용시 각 모델간 비율을 학습하여 최선의 성능을 보이도록 조절하는 방안의 모색이 있겠다

감사의 글

본 논문은 한국과학재단 기초연구사업과 2009년도 두뇌한국21사업의 지원을 받아 수행되었습니다

참고문헌

[1] Sa`sa Hasan and Hermann Ney : "Clustered Language Models based on Regular Expressions for SMT" 10th EAMT conference "Practical applications of machine translation", pp. 119-125, 30-31 May 2005.
 [2] Hirofumi Yamamoto and Eiichiro Sumita : "Bilingual

cluster based models for statistical machine translation" Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 514-523, June 28-30, 2007.

[3] Andrew Finch and Eiichiro Sumita : "Dynamic Model Interpolation for Statistical Machine Translation" ACL 2008 THIRD WORKSHOP ON STATISTICAL MACHINE TRANSLATION, pp. 208-215, June 19, 2008.

[4] Keiji Yasuda, Andrew Finch and Hideo Okuma : "System Description of NiCT-ATR SMT for NTCIR-7" Proceedings of NTCIR-7 Workshop Meeting, pp. 415-419, December 16-19, 2008.

[5] Takeshi Ito, Tomoyosi Akiba and Katunobu Itou : "Effect of the Topic Dependent Translation Models for Patent Translation - Experiment at NTCIR-7" Proceedings of NTCIR-7 Workshop Meeting, pp. 425-429, December 16-19, 2008.

[6] George Karypis : "CLUTO - A Clustering Tool kit" Technical Report 02-017, Dept. of Computer Science, University of Minnesota

[7] George Foster and Roland Kuhn : "Mixture- model adaptation for SMT" In Proceedings of the Second Workshop on Statistical Machine Translation ACL, pp. 128-135, Prague, Czech Republic, 2007.

[8] Michael Collins and Nigel Duffy 2001 Parsing with a Single Neuron: Covolution Kernels for Natural Language Problems. technical report UCSC-CRL-01 -01, university of california at santa cruz

[9] Taku Kudo and Yuji Matsumoto : "Fast Methods for Kernel-based Text Analysis" Proceedings of ACL-2003, pp. 24-31, 7-12 July 2003

[10] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto and Takehito Utsuro : "Overview of the Patent Translation Task at the ntcir-7 Workshop" Proceedings of NTCIR-7 Workshop Meeting, pp. 389-400, December 16-19, 2008.

[11] Andreas Stolcke. : "SRILM - an extensible language modeling toolkit" In Proc. of the 7th International Conference on Spoken Language Processing (ICSLP). pp. 693-696, 2002.

[12] Franz Josef Och and Hermann Ney : "A Systematic Comparison of Various Statistical Alignment Models" Computational Linguistics, 29(1):19-51, 2003.

[13] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst : "Moses: Open Source Toolkit for Statistical Machine Translation" Annual Meeting of the Association for Computational Linguistics (ACL), 2007.

[14] Franz Josef Och : "Minimum Error Rate Training for Statistical Machine Translation" Proc. of the 41st Annual Meeting of the Association for Computational Linguistics, pages 160-167, 2003.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. : "BLEU: A method for automatic evaluation of Machine Translation" Technical Report RC22176, IBM. 2001.