논문 2006-43CI-6-6

문맥가중치가 반영된 문장 유사 척도

(Context-Weighted Metrics for Example Matching)

김 동 주*, 김 한 우**

(Dong-Joo Kim and Han-Woo Kim)

요 약

본 논문은 영한 기계번역을 위한 예제기반 기계번역에서 예제 문장의 비교를 위한 척도에 관한 것으로 주어진 질의 문장과 가장 유사한 예제 문장을 찾아내는데 사용되는 유사성 척도를 제안한다. 제안하는 척도는 편집거리 알고리즘에 기반을 둔 것으로 표면어가 일치하지 않는 단어에 대해 기본적으로 단어의 표제어 정보와 품사 정보를 이용하여 유사도를 계산한다. 편집거리 척도는 비교 단위의 순서에 의존적이기는 하지만 순서만 일치하면 동일한 유사성 기여도를 갖는 것으로 판단하기 때문에 완전 문맥을 반영하지는 못한다. 따라서 본 논문에서는 완전 문맥 반영을 위해 추가적으로 이들 정보 외에 일치하는 단위 정보를 갖는 연속된 단어들에 대해 연속 정보를 반영한 문맥 가중치를 제안한다. 또한 비유사성 정도를 의미하는 척도인 편집거리 척도를 유사성 척도로 변경하고, 문맥 가중치가 적용된 척도를 문장 비교에 적용하기 위하여 정규화를 수행하며, 이를 통하여 유사도에 따른 순위를 결정한다. 또한 언어적 정보를 이용한 기존 방법류들에 대한 일반화를 시도하였으며, 문맥 가중치가 적용된 척도의 우수성을 증명하기 위해 일반화된 기존 방법류들과의 비교 실험을 수행하였다.

Abstract

This paper proposes a metrics for example matching under the example-based machine translation for English-Korean machine translation. Our metrics served as similarity measure is based on edit-distance algorithm, and it is employed to retrieve the most similar example sentences to a given query. Basically it makes use of simple information such as lemma and part-of-speech information of typographically mismatched words. Edit-distance algorithm cannot fully reflect the context of matched word units. In other words, only if matched word units are ordered, it is considered that the contribution of full matching context to similarity is identical to that of partial matching context for the sequence of words in which mismatching word units are intervened. To overcome this drawback, we propose the context-weighting scheme that uses the contiguity information of matched word units to catch the full context. To change the edit-distance metrics representing dissimilarity to similarity metrics, to apply this context-weighted metrics to the example matching problem, and also to rank by similarity, we normalize it. In addition, we generalize previous methods using some linguistic information to one representative system. In order to verify the correctness of the proposed context-weighted metrics, we carry out the experiment to compare it with generalized previous methods.

Keywords: 예문 일치(example matching), 문맥가중치 척도(context-weighted metrics), 편집거리(edit-distance), 번역 메모리(translation memory), 예제기반 기계번역(example-based machine translation)

Ⅰ. 서 론

말뭉치기반(corpus-based) 기계번역(machine translation)[1]은 기계번역 분야에서 가장 주목할 만한 분야 중에 하나이다. 통계기반(statistical) 기계번역, 예제기반(example-based) 기계번역, 사례기반(analogy-based, 혹은 case-based) 기계번역, 기억기반(memory-based) 기계번역 등과 같은 말뭉치기반 기계번역은 말뭉치를 사용하는 모든 종류들의 기계번역 방식들을 의미한다. 이들 말뭉치기반 기계번역들은 모두 이전 번역 예문들을 사용한다는 점에서 동일하지만 구현방식은 매우 상이하다. 이들 중 예제기반(example-based) 기계번역[2]과 더불어 번역 메모리(translation memory)[3]는 가장

* 학생회원, ** 정회원, 한양대학교 컴퓨터공학과
(Dept. of Com. Sci. & Eng., Hanyang University)
접수일자: 2006년10월9일, 수정완료일: 2006년10월30일

활발히 연구되는 분야 중 하나이다. 예제기반 기계번역은 원언어(source language) 문장을 이전에 번역된 예제를 이용하여 대상언어(target language)로의 번역을 수행한다. 번역 메모리는 이미 번역된 예문들을 재사용한다는 관점에서 예제기반 기계번역과 유사하다. 그러나 예제기반 기계번역은 근본적으로 자동번역을 지원하지만 기억 메모리는 인간 번역을 위한 대화형 도구로 번역을 수행하는 주체가 인간이라는 점에서 차이가 있다. 예제기반 기계번역과 번역 메모리의 두 가지 필수 요소는 대상언어와 원언어 문장 간의 대응관계(correspondence relation)를 파악하기 위한 정렬(alignment)과, 주어진 원언어 문장에 대해 가장 유사한 원언어 예문을 찾기 위한 유사문장 검색(혹은 예문 일치; example matching)이다. 정렬과 유사문장 검색의 단위는 일반적으로 문단, 문장, 절, 구, 혹은 단어가 될 수 있다.

본 논문은 예제기반 기계번역이나 번역 메모리 분야에서 사용되는 영한 기계번역을 위한 유사 문장 검색에 관한 것이다. 따라서 비교 대상이 되는 문장은 영어 문장이며, 본 논문에서는 영어 문장들 간의 단어 단위 비교를 위한 척도를 제안한다.

1. 유사 문장 검색

유사문장 검색 문제는 두 가지 쟁점으로 나뉘어 지는데, 하나는 예문 말뭉치의 검색 시간과 연관된 효율성 문제이고, 나머지 하나는 검색된 예문들의 질(quality)과 연관된 정확성 문제이다. 효율성에 관한 최근 접근 방법들은 여과(filtering)[4]나 군집(clustering)[5,6]을 통한 공간 축소 방식과 중복계산의 회피 방식[7]으로 나뉘어 진다. 공간 축소 방식은 검색 속도가 매우 빠른 반면 적합한 문장을 발견하지 못할 가능성이 존재하지만, 중복계산 회피 방식은 검색 속도는 공간 축소 방식에 비해 느리지만 적합한 문장을 발견하지 못할 가능성을 적다. 정확성과 관련한 대부분의 연구에서는 통사 정보나 의미 정보와 같은 언어적 정보를 사용한다. 본 논문은 정확성에 관련된 주제를 다루고 있으며 몇 가지 언어적 정보를 사용한다.

주어진 질의문장과 가장 유사한 예제 문장을 검색하기 위한 가장 일반적인 예제기반 기계번역 시스템이나 번역 메모리 시스템은 유사성, 혹은 거리 척도를 갖는다. 다양한 성공적인 척도[2, 5, 8-11]들이 제안되었지만, 예제기반 기계번역이나 번역 메모리 분야에서 가장 훌륭한 척도를 정의하는 문제는 여전히 해결되지 않은 문제이다[8].

본 논문에서는 표면어가 일치하지 않는 영어 단어에 대해 형태론적 표제어(lemma)와 품사 정보를 사용하여 주어진 질의 문장에 대해 예제 문장으로부터 가장 유사한 문장을 검색하기 위한 단어 수준의 예문 일치 척도를 제시한다. 제시하는 척도는 편집거리(edit-distance)[12] 알고리즘에 기반을 둔 방법으로 일반적인 편집거리 척도는 두 문장에 나열된 단어의 순서에 의존적이어서 전반적인 문맥은 반영된다. 그러나 단순히 순서에만 의존적인 척도는 단어 사이에 일치하지 않는 단어가 끼어드는 문맥의 문장 유사성 기여 정도와 연속적으로 일치하는 문맥의 문장 유사성 기여 정도를 구분하지 못하는 문제점을 가지고 있다. 이러한 단점을 극복하기 위해 본 논문에서는 연속하여 일치하는 단어 단위의 정보에 대한 가중치를 적용하여 완전 문맥을 반영하는 척도를 제안한다.

2. 관련연구

문장의 비교 문제에 있어서 단순히 표면 정보만을 반영할 경우 단어의 굴절요소(inflectional component)에 민감하여 단복수나 시제의 변화와 같은 어형변화에 따라 유사성 정도가 크게 변하게 된다. 이를 해결하기 위해 Nirenburg[9]는 형태론적 어형과 유의어, 하위어(혹은 상위어)를 사용한 일치 완화(matching relaxation)와 공유하지 않는 단어수를 이용한 문자열 구성 불일치(string composition discrepancies)에 기반을 둔 일치 척도를 제안하였다. 이 방법은 소규모 예문 말뭉치에서 매우 적응적으로 동작하지만 문장을 구성하고 있는 단어들의 연관성이나 의존성을 전혀 고려하지 않아 분별력은 매우 떨어진다.

어형 변화에 따른 영향을 완화하고 의미적인 유사성 정보를 반영할 뿐만 아니라 문장 상의 단어 순서에 따른 의존성을 반영하기 위해 Sumita[8]는 최대구체 공통추상(MSCA: Most Specific Common Abstraction)이라는 일치 척도를 도입하고 편집거리에 기반을 둔 방법을 제시하였다. 최대구체 공통추상 정보는 표면적으로 일치하지 않는 단어일지라도 의미적으로 유사한 단어에 대해 시소러스를 이용하여 비교하는 두 단어의 가장 구체적이면서 공통으로 갖는 의미 범주의 수준에 따른 일치 가중치를 주어 의미적 유사성을 반영하는 정보이다. 또한 이 방법은 편집 거리를 근간으로 하기 때문에 비록 제한적이고 부분적일지라도 문맥 의존성을 반영하고

있다. 그러나 문장 비교 문제에서 동적 프로그래밍 기법(dynamic technique)을 이용하여 유사성 정도를 계산하는 편집거리 기반 방식은 일치 단어에 대한 두 문장 상의 순서만 동일하면 같은 정도의 유사성 기여도를 부여하기 때문에 완전한 문맥을 반영한다고 볼 수 없다. 즉, 일치 단어들 사이에 일치하지 않는 단어가 끼어 있는 경우나 끼어있지 않는 경우나 모두 동일한 유사도를 갖는 것으로 간주하기 때문에 문장 전반적인 의존성만을 반영한다고 볼 수 있다.

Cranias[5]는 Sumita의 문제를 어느 정도 극복하였다. 즉, Cranias는 간접적이기는 하지만 편집거리에 기반을 둔 2단계 동적프로그래밍 기법(two-level dynamic programming technique)을 통하여 기능어(functional word)들 사이에 놓인 내용어(content word)들 간의 의존성을 반영하고 있다. 2단계 동적프로그래밍 기법은 첫 번째 단계에서 기능어가 일치하는 경우에만 두 기능어 사이에 놓인 내용어들에 대해 간단한 몇 가지 언어 정보를 사용하는 편집거리 기반의 두 번째 단계를 적용한다. 그러나 이 방법은 기능어로 둘러싸인 부분 문장이 청크(chunk)와 같은 의미적으로, 그리고 통사적으로 고립된 단위와 항상 일치하지는 않기 때문에 낮은 처리 범위를 갖는다는 문제점이 존재한다. 무엇보다 근본적인 문제점은 여전히 의존성이 반영되지 않는 부분들이 남는다는 것이다. 즉, 기능어와 이웃하는 내용어들 간에는 완전 의존성이 반영되지만 기능어들 간의 의존성이나 내용어들 간의 의존성은 Sumita의 방법과 동일하다.

본 논문에서는 이러한 기존 방법들에 대한 단점을 극복하기 위해 연속으로 일치하는 단위에 대해 문맥 가중치를 적용하여 완전 문맥을 반영하는 편집거리에 기반한 척도를 제안한다. 또한 편집거리에 기반한 기존 방법들에 대한 일반화를 시도하여 제안하는 방법과 정확성 비교를 수행한다.

Ⅱ. 문맥 가중치가 반영된 편집거리

1. 편집거리 알고리즘

두 문자열의 유사도를 계산하기 위한 편집거리 알고리즘은 패턴 인식, 생물정보학, 철자오류 교정, 음성인식 등 다양한 분야에서 사용된다. 이 알고리즘은 개념적으로 한 문자열을 비교 대상이 되는 다른 문자열로 변환하는데 필요한 삽입, 삭제 대치 연산의 최소 횟수를 계산하는 알고리즘으로 값이 크면 클수록 비유사성 정도가 커지는 문자열 비교를 위한 비유사성 척도이다.

이 알고리즘을 문장 비교 문제에 적용하기 위해 비교 단위를 문자가 아닌 단어로 한다. 즉, 질의 문장을 $X = x_1 x_2 \cdots x_m$, 예제 문장을 $Y = y_1 y_2 \cdots y_n$이라고 하자. 여기서 x_i, y_i는 각각 X, Y의 i번째 단어이고, 문장의 길이는 각각 $|X| = m$, $|Y| = n$이다. 또한 x_i에서 시작하여 x_j에 이르는 단어열을 X의 부분 문장 $x_{i\cdots j}$라고 표기한다($1 \le i,j \le m$). 이때, $D_{i,j}$를 $x_{1\cdots j}$를 $y_{1\cdots j}$로 변환하는데 필요한 최소 연산의 수라고 하면, 전통적인 편집거리 알고리즘에서 두 문장 X, Y 사이의 편집거리 $D_{m,n}$는 식 (1)과 같은 재귀에 의한 동적프로그래밍 알고리즘으로 계산된다. 이 알고리즘에서의 각 연산 삽입, 삭제, 대치, 일치에 대한 가중치 값들은 벌점(penalty score)으로 작용하여 한 번의 삽입 연산은 $D_{i,j} = D_{i,j-1} + 1$로 계산되고, 한 번의 삭제 연산은 $D_{i,j} = D_{i-1,j} + 1$로 계산된다. 또한 대치 연산은 $x_i \neq y_j$일 경우 $D_{i,j} = D_{i-1,j-1} + 1$로 계산된다. 일치 연산은 $x_i = y_j$의 경우 벌점이 없는 $D_{i,j} = D_{i-1,j-1}$로 계산된다.

$$
\begin{aligned}
&D_{0,0} = 0 \\
&D_{i,0} = i, \quad D_{0,j} = j \\
&D_{i,j} = \min\begin{cases} D_{i,j-1} + 1 \\ D_{i-1,j} + 1 \\ D_{i-1,j-1} + t_{i,j} \end{cases} \qquad (1) \\
&\text{where } t_{i,j} = \begin{cases} s, & \text{if } x_i = y_i \ \text{(match)} \\ r, & \text{if } x_i \neq y_j \ \text{(mismatch)} \end{cases} \\
&\text{Edit weight set: } W = \{s,r\} = \{0,1\}
\end{aligned}
$$

이 전통의 편집거리 알고리즘은 응용의 목적에 따라 다양하게 변경되기도 하고, 유사한 많은 종류들이 존재한다. Damerau[13]는 철자오류 교정을 위해 키보드 거리상의 근접성을 일치의 가중치로 도입하였다. Editex 거리[14]는 음성 일치를 위해 삽입과 삭제 연산만을 허용하였으며 또한 삽입과 삭제 연산이 적용되는 문자에 따라 서로 다른 가중치를 할당하였다. 더 나아가 연속된 삽입, 삭제 연산들의 길이에 따른 급격한 비유사성의 변화의 완화를 위해 연속 연산의 개수를 매개 변수로 하는 선형함수를 고안하여 가중치를 동적으로 결정하는데 사용하기도 하였다[15]. Needleman-Wunsch[16]의 알고리즘은 두 단백질에서의 아미노산열의 유사성을 판단하기 위해 실수 값을 가질 수 있도록 가중치 값을 일반화하였으며 동시에 유사성 정도에 기여하는 부분을 판단하

기 위해 역추적 정보 $P_{i,j}$를 유지하였다.

2. 언어정보를 반영한 편집거리

본 논문에서는 먼저 기존 방법들을 대신하고 제안하는 척도와 비교 대상이 되는 척도를 제시하기 위해 몇 가지 언어정보만을 사용하는 일반적인 척도를 제시한다. 이를 기반으로 문맥가중치가 반영된 편집거리 척도를 고안한다. 따라서 기존 방법들을 대표하는 보편적인 척도를 위해 우선적으로 요구되는 것은 몇몇 언어적 정보를 이용하여 X를 Y로 변환하는데 필요한 연산에 대한 가중치(혹은 벌점) 집합을 고안하는 일이다.

삽입과 삭제에 대한 가중치를 w_4이라고 하고, 식 (1)의 편집거리와 마찬가지로 최대값인 1로 설정한다. 이외의 나머지 일치와 불일치(대치) 연산에 대한 가중치 집합을 변경하게 된다.

우선 x_i가 y_j와 표면(철자)적으로 완전히 일치하는 내용어일 경우 연산에 대한 가중치 값으로 최소인 $w_0 = 0$을 설정한다. 또한 x_i가 y_j와 표면적으로 완전히 일치하는 기능어(관사, 전치사, 접속사)일 경우, 일치하는 기능어는 일치하는 내용어에 비해 의미적 유사성에 기여하는 정도가 더 작기 때문에 w_0보다 더 큰 값 w_1를 설정한다. 더 큰 값을 설정하는 이유는 (1)의 편집거리 척도와 동일하게 이 절에서 제시하는 척도는 비유사성 척도로 가중치는 벌점으로 작용하기 때문이다. 이와 같이 비교 대상이 되는 보편적 척도에서 의미적 유사성을 반영하기 위해 사용된 정보는 일치하는 단어가 내용어인지 기능어인지에 관한 정보뿐이다. 또한 굴절요소에 따른 유사성의 변화를 완화하기 위해 단어의 표제어(lemma) 정보를 사용한다. 다시 말해, x_i와 y_j가 표면적으로는 일치하지 않지만 x_i의 표제어와 y_j의 표제어가 서로 일치하는 경우, 표면적으로 완전히 일치하는 경우에 비해 유사성에 기여하는 정도가 낮기 때문에 w_1보다 더 큰 값 w_2를 설정한다. 마지막으로 표면적으로 완전히 다를 뿐만 아니라 표제어조차 다른 경우는 품사 정보를 사용한다. 즉, x_i와 y_j의 표면정보 뿐만 아니라 표제어조차도 일치하지 않는 경우, x_i의 품사와 y_j의 품사가 서로 일치한다면 w_2보다 더 큰 가중치 w_3를 설정한다. w_3는 최대 가중치인 $w_4 = 1$보다 더 작은 값이어야만 한다.

이렇게 정의된 새로운 전체 다섯 가지 가중치 집합 $W = \{w_0, w_1, w_2, w_3, w_4\}$는 다음과 같다.

- w_0: $x_i = y_j$이고, x_i와 y_j가 내용어 ($w_0 = 0$)
- w_1: $x_i = y_j$이고, x_i와 y_j가 기능어 ($w_0 \leq w_1$)
- w_2: $x_i \neq y_j$이고, x_i의 표제어가 y_j의 표제어와 같을 때 ($w_1 < w_2$)
- w_3: $x_i \neq y_j$이고, x_i의 표제어와 y_j의 표제어가 같지 않고, x_i의 품사가 y_j의 품사와 같을 때 ($w_2 \leq w_3 \leq w_4$)
- w_4: 위의 어느 경우도 아닐 때 ($w_4 = 1$)

이렇게 몇 가지 언어정보를 이용하여 정의된 가중치 집합은 식 (1)의 알고리즘에서의 W를 대신하게 된다. 본 논문에서 비록 위 가중치 집합 W를 사용하는 편집거리 척도를 새롭게 기술하였지만, 이 척도는 기존 방법류와 동일하게 몇 가지 언어정보만을 사용하고 문맥이 반영되지 않은 것이다. 다만 사용되는 정보의 종류와 정보를 이용한 가중치 융합 방식에서만 차이가 있을 뿐이다.

3. 문맥 가중치

이 절에서는 본 논문에서 제안하고 있는 문맥 가중치를 소개하고 2.2절에서 기술한 척도에 적용한 새로운 문맥 가중치 척도를 설명한다. 이 문맥 가중치는 2.2절의 척도뿐만 아니라 약간의 변형만 가한다면 편집 거리에 기반을 둔 어떤 종류의 척도에도 적용 가능하다.

완전 문맥을 반영하기 위한 핵심적인 개념은 이전에 일치한 단어의 정보를 참조하여 현재 다시 일치한다면 이전에 일치의 기여도 수준을 분류하여 현재 계산된 가중치의 기여도 수준과 비교해 차등 감소하는 것이다. 이를 위해 먼저 단어 u와 v의 각 단어가 포함된 전체 문장 X와 Y의 유사성 정도에 기여하는 기여도 수준(class)을 의미하는 $c(w_k)$를 식 (2)와 같이 가중치 수준에 따라 정의한다.

$$c(w_k) = \{k : \text{단어 } u\text{와 } v\text{가 } w_k\text{로 일치할 때 } k\} \quad (2)$$

이때 가중치는 유사성 기여정도에 따라서 내림차순, 비유사성에 기여하는 정도에 따라서 오름차순으로 정렬되어 있어야 한다. 즉, 2.2절의 가중치가 벌점 역할을 하는 척도에서는 임의의 정수 k, l에 대해서 집합 W의 가중치들은 다음의 식 (3)을 만족하여야만 한다.

$$W=\{w_k \mid 0 \le k, 0 < l, w_k \le w_{k+l}\} \quad (3)$$

2.2절의 가중치 수준에 따른다면 $c(w_k)$의 값은 $0 \le c(w_k) \le 4$ 범위를 갖는 정수이다. $c(w_k)$는 $C_{i,j}$에 저장되고 $C_{i,j}$의 값이 의미하는 바는 단어 x_i와 y_j의 각 단어가 포함된 전체 문장 X와 Y의 유사성 정도에 기여하는 기여도 수준이다. 현재 비교 대상이 되는 x_i와 y_j에 대한 $C_{i,j}$ 값과 이전 x_{i-1}와 y_{j-1}의 일치 연산에 대한 정보 $C_{i-1,j-1}$를 이용하여 문맥 가중치를 계산하기 위한 함수는 $x = C_{i-1,j-1} + C_{i,j}$라고 했을 때, 식 (4)와 같다.

$$CW(x) = (1-\lambda)\frac{x}{2\max(c(w_k))} + \lambda \quad (4)$$

여기서 문맥 가중치율 λ는 $0 \le \lambda \le 1$의 범위를 가지며, 2.2절의 언어정보에 따른 가중치를 갖는 2.1절의 편집거리 척도 (1)에서 일치 및 대치에 관한 $D_{i,j}$를 식 (4)를 이용하여 재정의 하면 식 (5)와 같다.

$$D_{i,j} = (D_{i-1,j-1} + w_k) \times CW(x) \quad (5)$$

이때 $C_{i,j}$의 값은 w_k에 의하여 k가 되고, 앞서 언급한 대로 x는 $C_{i-1,j-1} + C_{i,j}$가 된다. 식 (4)에서는 x의 값이 커지면 커질수록 $CW(x)$의 값은 커지는데, 그 범위는 $\lambda \le CW(x) \le 1$가 된다. x의 값이 크다는 의미는 이전 일치 연산과 현재의 일치 연산에서 문장 유사성에 대한 기여도가 낮다(벌점에 의한 가중치 값은 높다)는 의미이기 때문에 문맥 가중치로 1에 가까운 큰 값이 곱해져 $D_{i-1,j-1}$의 벌점값 수준의 유지하는 성향을 보이게 된다. 반대로 x의 값이 작다는 의미는 이전 일치 연산과 현재 일치 연산에서 문자 유사성에 대한 기여도가 높다(벌점에 의한 가중치 값은 낮다)는 의미이기 때문에 문맥 가중치로 작은 값이 곱해져 $D_{i-1,j-1}$의 벌점값 수준의 떨어뜨리는 성향을 보이게 된다. 또한 x로 가능한 값의 종류는 전체 일치에 따른 가중치의 개수의 제곱이 존재하며, 2.2절의 가중치에 따른다면 총 25가지가 존재하게 된다. 따라서 (4)는 일대일 관계의 함수이므로 문맥 가중치 $CW(x)$값 역시 가중치 개수의 제곱 가지가 존재한다. λ는 문맥 가중치의 최저 경계값으로 이 값이 작아진다면 x가 가질 수 있는 값들에 대한 $CW(x)$의 간격이 커져 문장 유사도에 대한 문맥 기여도의 폭 또한 커지므로 문맥에 대한 영향력이 커진다. 극단적으로 λ가 0이라면 두 문장 중에 일치하는 내용어가 나타나게 되면 그 때의 $D_{i,j}$의 값은 0이 되어버리고, 문장 전체의 유사도는 일치하는 내용어나 기능어의 발생 빈도에 의해 결정될 가능성이 높아진다. 반대로 λ가 1이라면 $CW(x) = 1$이 되고, 이 경우 문맥이 전혀 반영이 되지 않게 된다. 따라서 적합한 λ의 값을 결정하는 일은 매우 중요하다.

2.2절의 가중치 집합을 이용하여 본 절에서 설명한 문맥 가중치 $CW(x)$를 반영한 전체 편집거리 척도는 알고리즘 (6)과 같다 알고리즘 (6)에서 W의 값 case 3~case7은 2.2절에서 설명한 바와 같다.

$$\begin{aligned}
&D_{0,0} = 0 \\
&D_{i,0} = i, \quad D_{0,j} = j \\
&C_{0,0} = C_{i,0} = C_{0,j} = 4 \\
&D_{i,j} = \min\begin{cases} D_{i,j-1} + 1 & [\text{case1}] \\ D_{i-1,j} + 1 & [\text{case2}] \\ (D_{i-1,j-1} + W) \times CW(x) \end{cases} \\
&\text{where} \quad W = \begin{cases} w_0 = 0 & [\text{case3}] \\ w_1 & [\text{case4}] \\ w_2 & [\text{case5}] \\ w_3 & [\text{case6}] \\ w_4 = 1 & [\text{case7}] \end{cases} \\
&\qquad x = C_{i-1,j-1} + C_{i,j} \\
&C_{i,j} = \begin{cases} 0 & [\text{case3}] \\ 1 & [\text{case4}] \\ 2 & [\text{case5}] \\ 3 & [\text{case6}] \\ 4 & [\text{case1, case2, case7}] \end{cases}
\end{aligned} \quad (6)$$

그림 1은 문맥 가중치를 적용한 편집거리 척도 (6)을 이용하여 질의 문장 "There are a few limits to the reflector."에 대한 예제 문장 "But there is a potential drawback to the volume."의 비유사성 정도 $D_{9,10}$를 계산하는 과정의 중간결과인 $D_{5,9}$에 대한 계산이다. 이때, 문맥 가중치율은 $\lambda = 0.7$, 가중치는 각각 $w_0 = 0$, $w_1 = 0.1$, $w_2 = 0.2$, $w_3 = 0.4$, $w_4 = 1.0$로 설정하였으며, 식 (5)와 기여도 수준 $C_{i,j} = c(w_k)$는 (7)과 같은 형태로 표현되어 있고, 실선은 최소값으로 선택되었다는 것을 의미하고 있다.

$$\begin{matrix} D_{i-1,j-1} \\ C_{i-1,j-1} \end{matrix} \xrightarrow{+ w_k) \times CW(x)} \begin{matrix} D_{i,j} \\ C_{i,j} \end{matrix} \quad (7)$$

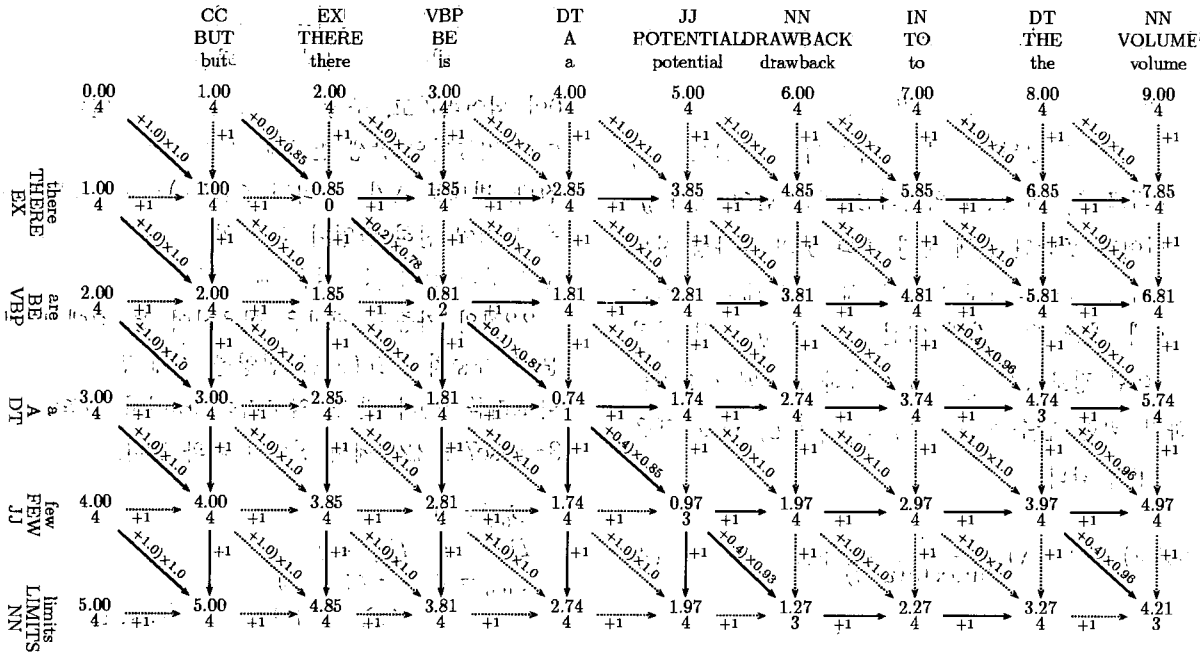

그림 1. 문맥 가중치가 반영된 편집거리 척도의 계산 예

Fig. 1. A calculation example for context-weighted edit-distance metric.

3. 정규화

질의 문장과 가장 유사한 예제 문장을 선택하기 위해서는 예문 말뭉치의 모든 예문에 대해 유사도에 따른 순위를 매겨야만 한다. 2.3절의 알고리즘 (6)의 척도는 문장의 길이에 의존적인 값이므로 순위를 매길 수 없다. 따라서 식 (8)과 같이 정규화를 수행해야만 한다.

$$SIM(X,Y)=\frac{D_{m,n}-\max(m,n)}{D_{m,m}-\max(m,n)} \tag{8}$$

식 (8)의 $SIM(X,Y)$은 두 문장 X와 Y의 유사도 값으로 $0 \le SIM(X,Y) \le 1$의 범위를 가지며, $\max(m,n)$은 X의 길이 m과 Y의 길이 n 중 더 큰 값을 의미한다. $D_{m,n}$은 문장 X와 Y의 비유사성 정도를 의미하지만 $SIM(X,Y)$는 유사성 정도를 나타낸다. $D_{m,m}$은 질의 예문의 자기유사도(self-similarity)를 의미하는데, 이를 계산하는 간단한 방법은 식 (6)의 알고리즘을 이용하여 문장 X와 X에 대해 직접적으로 계산하는 것이다. 그러나 $D_{m,m}=d_m$이라고 하고, 더 간단하게 다음 알고리즘 (9)과 같이 재귀적으로 계산할 수 있다.

$$\begin{aligned} &d_0=0\\ &d_i=(d_{i-1}+M)\times\alpha\\ &\text{where}\\ &M=\begin{cases}w_0 & [\text{case1}]\\ w_1 & [\text{case2}]\end{cases} \quad \alpha=\begin{cases}\lambda & [\text{case3}]\\ (1+7\lambda)/8 & [\text{case4}]\\ (1+3\lambda)/4 & [\text{case5}]\end{cases} \end{aligned} \tag{9}$$

알고리즘 (9)에서 각 경우(case1~case5)에 대한 설명은 다음과 같다.

- case1: x_i가 내용어일 경우
- case2: x_i가 기능어일 경우
- case3: x_{i-1}, x_i 모두 내용어일 경우
- case4: x_{i-1}, x_i 둘 중 어느 하나가 기능어일 경우
- case5: x_{i-1}, x_i 모두 기능어일 경우

d_m의 최소값은 질의 문장 X를 구성하는 모든 단어가 내용어일 경우이고 최대값은 모든 단어가 기능어일 경우로 값의 범위는 식 (10)과 같다. 2.2절의 가중치 값과 같이 $w_0=0$이라면 최소값은 0이 된다.

$$w_0\sum_{i=1}^{m}\lambda^i \le d_m \le w_1\sum_{i=1}^{m}\left(\frac{1+3\lambda}{4}\right)^i \tag{10}$$

2.1절에 기술한 전통의 편집거리 척도에서는 $\lambda=1$이고 $w_0=w_1=0$이기 때문에 자기유사도 d_m의 값은 항상 0이 된다. 또한 2.2절에 기술한 언어정보를 사용하는 편집거리 척도에서는 문맥 가중치가 사용되지 않기 때문에 $\lambda=1$이고, 이때 최소값은 $w_0 \cdot m$이 되고 최대값은 $w_1 \cdot m$이 된다. 그림 1의 질의 문장 일부 $x_{1\ldots5}$와 예제 문장 일부 $y_{1\ldots9}$대한 계산 예를 식 (8)의 정규화

를 통한 유사도 값 $SIM(x_{1\ldots5}, y_{1\ldots9})$을 계산하면 $(4.207+9)/(0.038+9)=0.535$가 된다.

Ⅲ. 실 험

실험을 위해 Penn-Tree Bank 말뭉치 버전 2로부터 품사 태깅된 "The Wall Street Journal(WSJ)"의 사설을 사용하였다. 태깅된 WSJ의 전체 문장 수는 약 53,797이고, 전체 토큰 수는 약 1,288,775이며, 평균 문장 길이는 23.96이고 문장 길이는 1에서부터 283개까지 매우 다양하다. 헤드라인이나 날짜, 표의 내용 등과 같이 완전한 문장이 아니거나 비정상적으로 긴 문장의 제거를 통하여 잡음 문장을 최소화하기 위해 문장의 길이가 10이상이고 23보다 작은 문장들 23,720개만 선택하여 사용하였다. 선택된 문장들은 WordSmith 도구로 표제어를 추출하였으며, 선택된 문장의 통계적 특성은 표 1과 같다. 표 1에서 서로 다른 표제어의 수에서 기호, 구두점 및 숫자는 제외했다.

실험을 위해 경험적으로 문맥 가중치율 $\lambda_1 = 0.7$과, 가중치 집합 $W_1 = \{w_0 = 0.0,\ w_1 = 0.1;\ w_2 = 0.2,\ w_3 = 0.4,\ w_4 = 1.0\}$를 설정하였다. 또한 비교 실험을 위해 또 다른 문맥 가중치율 $\lambda_0 = 1.0$을 설정하고, 또 다른 가중치 집합 $W_0 = \{w_0 = 0.0,\ w_1 = 0.0,\ w_2 = 1.0,\ w_3 = 1.0,\ w_4 = 1.0\}$를 설정하였다.

λ_0와 W_0를 적용한 척도를 $\lambda_0 W_0$라고 표기하였을 때, 이 척도는 2.1절에서 설명한 표면정보만을 사용하는 전통의 편집거리 척도를 정규화한 것과 완전히 동일하게 된다. 또한 λ_0와 W_1을 적용한 척도를 $\lambda_0 W_1$이라 표기했을 때, 이 척도는 2.2절에서 제시한 문맥 가중치를 사용하지 않고 언어정보만을 사용하는 편집거리에 기반을 둔 기존의 여러 척도들과 등가이다. 물론 기존의 대부분 척도들은 2.2절에서 제시한 언어정보보다는 더 풍부한 정보를 사용하기 때문에 명백히 더 우수할 것이지만, 문맥을 반영하지 못한다는 점에서 $\lambda_0 W_1$ 척도와 동등하다고 볼 수 있다. 본 논문에서는 언어정보를 사용하는 편집거리 척도에 문맥 가중치를 적용한 제안하는 $\lambda_1 W_1$ 척도와 언어정보를 사용하지만 문맥 가중치가 반영이 되지 않은 $\lambda_0 W_1$ 척도를 비교함으로써 문맥 가중치에 의한 순수한 정확률 향상을 조사하였다.

표 1. 실험코퍼스의 통계적 특성
Table 1. Statistics of experiment corpus.

문장 수	23,720
토큰 수	405,721
서로 다른 토큰 수	29,129
서로 다른 표제어 수	17,609
문장당 평균 토큰 수	17.10

먼저 개략적인 동작 성향을 파악하기 위해 유사문장의 발견 성공 여부를 의미하는 경계값인 임계치를 설정하였다. 이를 위해 실험 코퍼스로부터 임의로 100개의 문장을 선택하여 질의 문장으로 사용하고 나머지 문장들을 예제 문장으로 사용하였다. 각 질의 문장에 대해 가장 유사도가 높게 계산되는 상위 세 개의 문장을 조사하였다. 질의 문장에 대해 발견된 상위 세 개의 문장들 간의 유사도의 정확성을 평가하기 위해 투표 방식을 사용하였다. 즉, 특정 질의문장과 검색된 문장 한 개에 대해 다섯 명의 투표자 중 최소한 세 명이 유사하다고 판정을 한다면, 척도는 해당 질의 문장과 검색 문장에 대해 정확한 것으로 간주하였다. 이렇게 하여 주어진 질의 문장의 대해 검색된 상위 세 개의 문장의 전반적인 유사성 정도에 따라 다음과 같은 세 가지 등급을 매겼다.

- Class A: 세 문장 모두가 유사한 것으로 판정
- Class B: 한 문장, 두 문장만 유사한 것으로 판정
- Class C: 세 문장 모두 유사하지 않은 것으로 판정

이렇게 하여 $SIM(X, Y)$의 값을 0.2에서 0.95로 변화시켜가면서 각 등급에 속하는 질의문의 수를 조사하였다. 그림 2는 $SIM(X, Y)$의 변화에 따라 각 등급에

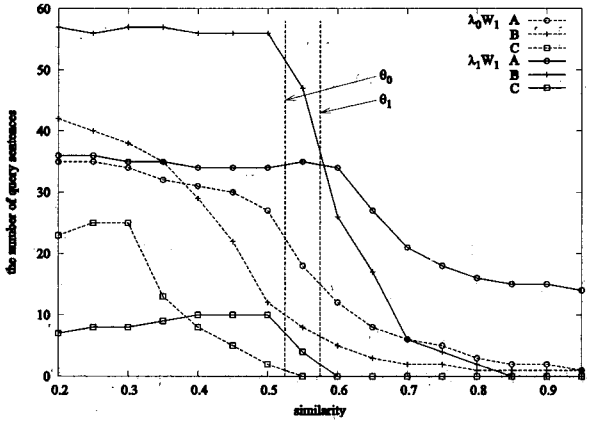

그림 2. 임계치를 결정하기 위한 실험 결과
Fig. 2. Results of experiment to determine thresholds.

표 2. 정확률 평가결과
Table 2. Experimental results for correctness.

척도		$\lambda_0 W_1$	$\lambda_1 W_1$
발견된 전체 예제 문장 수		4,519	7,861
발견된 평균 예제 문장 수		15.06	26.20
질의 문장 수	발견 문장수	198 (66.0%)	239 (79.7%)
	제1종의 오류	37 (12.3%)	14 (4.7%)
	미발견 문장수	102 (34.0%)	61 (20.3%)
	제2종의 오류	13 (4.3%)	11 (3.7%)

속하는 질의문의 수를 나타내고 있다.

유사문장 검색 성공과 실패 사이의 최적의 임계치 조건은 C등급의 수가 0개이어야 하고 A등급과 B등급의 수는 최대가 되어야만 한다. 따라서 그림 2에서 보는 바와 같이 $\lambda_1 W_1$ 척도의 경우 $SIM(X, Y)$의 0.55일 때 C등급의 잘의 문장 수가 4이고 $SIM(X, Y)$이 0.6일 때 0이다. 따라서 $\lambda_1 W_1$ 척도에 대한 최적의 임계치는 중간값인 $\theta_1 = 0.575$로 설정하였다. 같은 방식으로 $\lambda_0 W_1$ 척도에 대한 임계치는 $\theta_0 = 0.525$로 설정했다.

임계치 설정을 위해 말뭉치로부터 선택했던 100개의 문장을 제외한 나머지 문장 중에서 제안된 척도 $\lambda_1 W_1$과 $\lambda_0 W_1$의 비교를 위해 임의로 300개의 문장을 선택하였고, 나머지 문장을 예제 문장으로 사용하였다. 표 2는 실험 결과로, 두 척도에 대한 각각의 임계치 θ_0과 θ_1보다 큰 유사도 값을 갖는 발견된 문장들의 수와 오류율을 보이고 있다. 검색 성공과 실패에 대한 $\lambda_1 W_1$의 오류율은 제1종의 오류(type I; 잘못된 긍정)와 제2종의 오류(type II; 잘못된 부정)을 합한 약 8.4%였다. 반면에 $\lambda_0 W_1$의 오류율은 16.6%나 되었다. 따라서 제안하는 문맥 가중치가 반영된 척도 $\lambda_1 W_1$는 문맥 가중치가 반영되지 않고 단순 언어정보만을 사용하는 척도 $\lambda_0 W_1$보다 매우 우수함을 알 수 있었다.

IV. 결 론

본 논문에서는 영한 기계번역을 위한 예제기반 기계번역이나 번역 메모리에서 질의 문장과 예제 문장들 간의 유사성을 측정하기 위한 새로운 문맥 가중치 척도를 제안하였다. 제안하는 척도는 언어적 정보를 사용하는 편집거리 기반의 기존 척도들과는 달리 완전 문맥의 반영을 시도한 것이다. 또한 제안하는 척도의 우수성을 입증하기 위해 기존 척도들을 포괄하는 일반화를 수행하였다. 비록 제안하는 척도의 핵심인 문맥 가중치를 일반화된 척도에 적용하였지만, 약간의 변화만 준다면 편집거리 척도에 기반하고 있는 기존의 어떠한 척도들에도 응용이 가능할 것이다.

문맥 가중치 척도에서 문맥 유사성에 대한 기여도를 높이는 역할을 수행하는 매개변수는 λ이다. λ를 크게 설정하면 할수록 문맥의 역할이 약화되며, 반대로 λ를 작게 설정하면 할수록 문장간의 유사성 정도는 비교 단위가 얼마나 연속적으로 길게 일치하여 왔는가에 결정되는 문맥의 역할이 강화되게 된다. 그러나 λ가 너무 작으면 유사한 문장들간의 분별력이 떨어지는 문제점을 낳게 된다. 따라서 남은 과제는 λ의 값을 자동적이면서 합리적으로 설정하는 방안을 연구하는 일이다.

참 고 문 헌

[1] H. L. Somers, ""New Paradigms" in MT: the State of the Play now that the Dust has Settled," In *10th European Summer School in Logic, Language and Information, Workshop on Machine Translation*, pp.22-33, 1998.

[2] M. Nagao, "A Framework of a Mechanical Translation between Japanese and English by Analogy Principle," In *Artificial and human intelligence*, A. Elithorn and R. Banerji (Eds.), Amsterdam: North-Holland, pp.173-180, 1984.

[3] M. Kay, "The Proper Place of Men and Machines in Language Translation, Research Report CSL-80-11; Xerox Palo Alto Research Center. Palo Alto. Calif., Reprinted in *Machine Translatin* vol.12, pp.3-23 (1997), 1980.

[4] F. Mandreoli, R. Martoglia and P. Tiberio, "Searching similar (sub)sentences for example-based machine translation," In *Atti del Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD)*, pp.208-221, Isola d'Elba, Italy, 2002.

[5] L. Cranias, H. Papageorgiou and S. Piperidis, "A matching technique in example-based machine translation," In *Proc. 15th Int. Conf. on Computational Linguistics*, pp.100-104, 1994.

[6] L. Cranias, H. Papageorgiou and S. Piperidis, "Clustering: A technique for search space reduction in example-based machine translation," In *Proc. Int. Conf. on Systems, Man, and Cybernetics*, pp.1-6, 1994.

[7] T. Doi, H. Yamamoto and E. Sumita, "Graph-based retrieval for example-based

machine translation using edit-distance," In *Proc. Workshop Example-Based Machine Translation at MT Submmit X*, pp.51-58, 2005.
[8] E. Sumita and H. Iida, "Experiments and Prospects of Example-Based Machine Translation," In *Proc. of the 29th Annual Meeting of the ACL*, pp.185-192, 1991.
[9] S. Nirenburg, C. Domashnev and D. Grannes, "Two approaches to matching in example-based machine translation," In *Proc. 5th Int. Conf. on Theoretical and Methodological Issues in Machine Translation*, pp.47-57, 1993.
[10] T. Baldwin and H. Tanaka, "The Effects of Word Order and Segmentation on Translation Retrieval Performance," In *Proc. of the 18th Int. Conf. on Computational Linguistics*, pp.35-41, 2000.
[11] E. Sumita, "Example-based machine translation using DP-matching between word sequences," In *Proc. of the ACL Workshop on Data-Driven Methods in MT.*, pp.1-8, 2001.
[12] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics-Doklady*, vol.10 no.8 pp.707-710, 1996, Translated from *Doklady Akademii Nauk SSSR*, vol.163, no.4 pp.845-848, 1965.
[13] F. J. Damerau, "A Technique for Computer Detection and Correction of Spelling Errors," *Communications of the ACM*, vol.7, no.3, pp.171-176, 1964.
[14] J. Zobel and P. Dart, "Phonetic String Matching : Lessons from Information Retrieval," In *Proc. of the 19th Annual International ACM SIGIR Conf.*, pp.166-172, 1996.
[15] E. W. Meyers and W. Miller, "Row replacement algorithms for screen editors," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol.11, no.1, pp.33-56, 1989.
[16] S. Needleman and D. Wunsch, "A General Method Applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol.48, no.3, pp.443-453, 1970.

저 자 소 개

김 동 준(학생회원)
1996년 한양대학교 전자계산학과 학사 졸업.
1998년 한양대학교 전자계산학과 석사 졸업
1998년~현재 한양대학교 컴퓨터공학과 박사과정 재학중.
<주관심분야 : 한국어 형태소 및 구문분석, 기계번역, 맞춤법검사, 정보검색>

김 한 우(정회원)
1975년 한양대학교 전자공학과 학사 졸업.
1978년 한양대학교 전자공학과 석사 졸업.
1980년 일본 동경대학 정보공학과 연구원.
1981년~현재 한양대학교 컴퓨터공학과 교수, 정보통신부 문자방송기술협회 이사.
<주관심분야 : 기계번역, 한국어정보처리, 자연언어질의응답>