

## 채팅 시스템 구현을 위한 3단계 문장 검색 방법

전원표<sup>1</sup> · 송영길<sup>2</sup> · 김학수<sup>†</sup>

(원고접수일 : 2012년 12월 27일, 원고수정일 : 2013년 1월 10일, 심사완료일 : 2013년 2월 25일)

### A three-step sentence searching method for implementing a chatting system

Won-Pyo Jeon<sup>1</sup> · Yoeng-Kil Song<sup>2</sup> · Hark-Soo Kim<sup>†</sup>

**요약:** 기존 채팅 시스템은 일반적으로 사용자 입력 문장과 데이터베이스 내 목표 문장들 사이의 어휘 일치도에 기반을 둔 방법을 사용한다. 그러나 이러한 방법은 어휘 불일치 문제를 자주 일으킨다. 이러한 문제를 해결하기 위해 순차적으로 적용되는 3단계 문장 검색 방법을 제안한다. 첫 번째 단계는 어휘 수준에서 사용자 입력 문장과 목표 문장들 사이의 공통 키워드 열을 비교하는 것이다. 두 번째 단계는 의미 수준에서 사용자 입력 문장과 데이터베이스 내 문장들 사이의 문장 유형과 의미 표지를 비교하는 것이다. 마지막 단계는 미리 정의된 어휘-구문 패턴을 사용자 입력 문장과 매칭하는 것이다. 실험에서 제안된 방법은 단순 키워드 매칭 방법 보다 더 나은 응답 정확도와 사용자 만족도를 보였다.

**주제어:** 채팅, 어휘 수준 검색, 의미 수준 검색, 어휘-구문 패턴 매칭

**Abstract:** The previous chatting systems have generally used methods based on lexical agreement between users' input sentences and target sentences in a database. However, these methods often raise well-known lexical disagreement problems. To resolve some of lexical disagreement problems, we propose a three-step sentence searching method that is sequentially applied when the previous step is failed. The first step is to compare common keyword sequences between users' inputs and target sentences in the lexical level. The second step is to compare sentence types and semantic markers between users' input and target sentences in the semantic level. The last step is to match users' inputs against predefined lexico-syntactic patterns. In the experiments, the proposed method showed better response precision and user satisfaction rate than simple keyword matching methods.

**Keywords:** Chatting, Lexical-level searching, Semantic-level searching, Lexico-syntactic pattern matching

## 1. 서론

최근 해양 정보 관련 비정형 데이터의 양이 많이 증가하고 있다[1]. 이러한 데이터를 사용자 친화적으로 사용하기 위한 많은 연구가 진행되고 있다 [2][3]. 사용자 친화적인 해양 정보 시스템을 구축하기 위한 방법으로 애플(Apple)사의 시리(Siri)와 같은 가상 비서 소프트웨어가 각광받고 있다. 본 논문은 이러한 시스템을 구축에 기반이 되는 채팅 시스템

구현을 위한 문장 검색 방법을 제안한다.

채팅 시스템은 인간에게 익숙한 자연어를 사용한다는 점에서 사용자 친화적인 인터페이스이며, 목적 지향 대화 시스템을 보완하여 시스템 환경을 인간 중심적으로 만든다는 점에서 중요한 의미를 가진다. 이러한 채팅 시스템은 그 목적에 따라 크게 목적 지향 채팅 시스템과 말 잇기 채팅 시스템으로 나눌 수 있다. 목적 지향 채팅 시스템은 사용자로부터 질의

† 교신저자: (200-701) 강원도 춘천시 강원대학길1,

강원대학교 컴퓨터정보통신공학전공, E-mail: nlpdrkim@kangwon.ac.kr, Tel: 033-250-6388

1 강원대학교 컴퓨터정보통신공학전공, E-mail: nlpjwp@kangwon.ac.kr, Tel: 033-251-6388

2 강원대학교 컴퓨터정보통신공학전공, E-mail: nlpyksong@kangwon.ac.kr, Tel: 033-251-6388

를 입력받아 길 안내나 조작 방법 등을 알려주는 것을 목적으로 한다. 대화 형식의 비행기, 기차표 예매 시스템 등이 여기에 해당한다. 말 잇기 채팅 시스템은 특별한 목적 없이 흐름을 유지한 채 대화를 진행해 나가는 것을 목적으로 한다. 최초의 채팅 시스템인 'ELIZA'가 여기에 해당한다[4]. 목적에 따라 구분을 하였지만 궁극적으로는 사용자로 하여금 마치 사람과 채팅하는 것 같은 느낌이 들 수 있도록 자연스러운 응답을 생성하는 것을 목적으로 한다. 그러나 기존의 채팅 시스템들은 대부분 단순한 패턴 매칭 기법을 사용하기 때문에 특정 영역에 한정되거나 인간의 복잡하고 다양한 대화 현상을 설명하기에는 부족한 성능을 보여준다. 사용자에게 사람과 대화하듯이 자연스러움을 느끼게 하기 위해서는 사용자가 입력한 어휘 정보만으로는 한계가 있다. 어휘 정보에만 의존하여 대화를 한다면 비슷하거나 동일한 대화가 계속되어 채팅의 유연성이 부족해진다. 본 논문에서는 이러한 문제를 해결하기 위해 사용자 발화의 의미를 고려하고 말뭉치 활용도를 극대화하여 응답할 수 있는 3단계 문장 검색 기법과 그것을 위한 색인 방법을 제안한다.

## 2. 관련 연구

기존의 채팅 시스템에 관한 연구로는 패턴 매칭(pattern matching) 방법, 키워드 인식(keyword recognition), 마르코프 모델(Markov model), 유전 알고리즘(genetic algorithm) 등을 이용한 방법이 있다[5].

최초의 채팅 시스템인 ELIZA는 사용자가 입력한 문장에 포함된 키워드로부터 미리 정의된 문장으로 응답하는 간단한 패턴 매칭 방법과 어휘를 치환하는 등의 방법을 사용한다. 이와 유사한 시스템으로 Julia[6], Alice[7] 등이 있으며, 다양한 기능과 패턴 추가를 통해 지속적인 성능 향상을 가져왔다. 패턴 매칭 방법은 구현이 매우 쉬운 장점이 있지만 비슷하거나 동일한 대화가 계속되어 채팅의 유연성이 부족하고, 채팅의 수준을 향상시키기 위해서는 대량의 말뭉치가 필요하다는 단점이 있다.

마르코프 모델과 유전 알고리즘을 이용한 방법은 검색이 아닌 새로운 문장을 생성하는 방법이다. 마르코프 모델을 이용한 방법은 인간의 대화 내용을

마르코프 모형으로 저장한 후 사용자의 입력에 대해 데이터를 조회하여 가장 적절한 문장을 생성하는 문장 생성 기법 중 하나이다[8]. 유전 알고리즘은 기존 데이터베이스에 저장되어 있는 데이터로부터 새로운 데이터를 생성하는 방법이다. Vrajitoru [8][10]는 유전 알고리즘을 활용하여 기존 문장으로부터 더 나은 응답 문장을 생성하는 방법을 제안하였다. 하지만 이 방법은 문법이 잘못된 문장을 생성할 수 있기 때문에 적절한 필터링이 필요하다.

본 논문에서는 어휘 정보에 의존하여 검색할 경우 발생하는 문제를 해결하기 위해 3단계 문장 검색 기법과 그것을 위한 색인 방법을 제안한다. 제안 방법은 문장의 의미를 고려하여 데이터를 검색하기 때문에 기본 채팅 말뭉치의 활용도를 극대화할 수 있다.

## 3. 3단계 색인 및 검색 기법

제안 방법은 어휘 수준, 의미 수준, 패턴 기반 말 잇기 모두 3단계로 구성되어 있다. 각 단계 별로 필요한 자질을 추출하여 색인 데이터베이스를 생성하고 검색한다. 제안 시스템의 구조는 Figure 1과 같다.

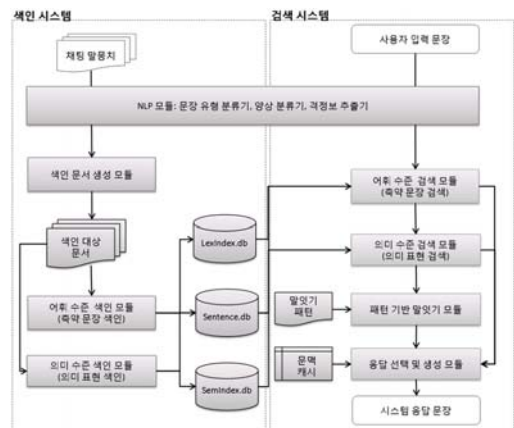


Figure 1: Overview of the proposed system

제안 시스템은 크게 두 부분으로 구성되어 있다. 먼저 사용자 발화와 시스템 발화의 쌍으로 구성된 말뭉치의 사용자 발화 문장에서 각 단계 별로 필요한 자질을 추출하여 데이터베이스를 생성하는 색인 시스템과 사용자 입력 문장과 유사한 사용자 발화

문장의 검색을 수행할 검색 시스템으로 나뉘어 있다. 각 단계에 대한 자세한 내용은 다음 절에서 설명한다.

### 3.1 어휘 수준 색인 및 검색

본 절에서는 1단계 어휘 수준 색인 및 검색에 대하여 설명한다. 어휘 수준 검색은 3단계 문장 축약 및 매칭 기법을 사용한다. **Table 1**은 어휘 수준 검색에서 각 단계별 문장 축약 수준을 나타낸다.

**Table 1:** Three steps of the sentence abstraction

		1단계	2단계	3단계
색인 대상 형태소 태그		ALL	명사류, 동사, 형용사, 부사류	명사류
필수 태그		명사류, 동사, 형용사	명사류, 동사, 형용사	명사류
형태소 휴리스틱	STOPWORD	NO	NO	YES
	대표어 정보	YES	YES	YES
	색인등록대상	ALL	YES	YES
문장 휴리스틱	중복기호처리	NO	NO	YES
	분용언 추출	NO	YES	YES
사전	문장정보추출	NO	YES	YES
	불용어 사전	YES	YES	YES
	대표어 사전	YES	YES	YES

어휘 수준 검색은 **Table 1**에서 보는 것처럼 각 단계 별로 색인 대상이 되는 형태소 태그와 사용되는 휴리스틱 기법이 다르다. 1단계에서는 사용자 발화의 형태소 분석 결과를 모두 색인한다. 이를 기반으로 1단계에서 매칭될 경우 가장 정확한 응답이 선택되어 발화된다. 2단계에서는 사용자 발화의 형태소 분석 결과를 모두 색인 하지 않고 명사류, 동사, 형용사, 부사류를 대상으로 색인한다. 그리고 분용언과 문장 정보를 휴리스틱하게 추출하여 사용한다. 문장 정보는 문형, 형식, 시제가 있다. 3단계에서는 사용자 발화에서 명사류만 추출하여 색인한다. **Table 2**는 “난 병원 냄새가 너무 싫어”를 각 단계별로 색인한 예를 보여준다. **Table 2**에서 보는 바와 같이 입력 문장 “난 병원 냄새가 너무 싫어”에 대하여 위와 같이 축약되어 색인되기 때문에 이와 유사한 문장 “병원냄새는 너무 싫어”를 입력하여도 검색이

가능하다. 그러나 이 방법의 경우 축약의 단계가 낮은 경우에 검색된 결과가 거의 정확하다는 장점이 있지만, 어휘가 정확히 동일해야만 검색이 된다는 단점이 있다. 또한 축약의 단계가 높은 경우에는 입력과 너무 동떨어진 검색 결과가 나올 수 있다. 그래서 제안 방법에서는 어휘 수준 검색 1~3단계를 순서대로 진행하지 않고, 어휘 수준 검색 1, 2단계 수행 후 2절의 의미 수준 검색을 수행한 뒤 어휘 수준 검색 3단계를 수행하도록 하였다.

**Table 2:** An Example of lexical-level indexing

	색인 키
1단계	난, 병원, 냄새, 가, 너무, 싫어
2단계	난, 병원, 냄새, 너무, MV: 싫
3단계	병원, 냄새, MV: 싫

### 3.2 의미 수준 색인 및 검색

사용자 발화를 의미 수준에서 색인 및 검색하기 위해서는 사용자 발화에서 의미 정보를 추출해야 한다. 이를 기반으로 사용자 발화를 준 의미 표현으로 변환하여 색인하고, 의미 표현 매칭 및 순위화 한다.

#### 3.2.1 의미 정보 추출 및 의미 표현 색인

본 절에서는 의미 정보를 추출하고, 이를 기반으로 의미 표현 색인하는 방법에 대하여 설명한다. 본 논문에서 사용한 의미 정보는 **Table 3, 4, 5**와 같이 정의한다.

**Table 5**에서 격 정보를 다섯 가지만 사용한 이유는 그 이상 세분화하여 사용할 경우 정확한 격 정보 추출이 어렵기 때문이다. 또한 잘못된 격 정보를 이용하여 검색할 경우 의미 수준 색인 및 검색에 미치는 영향이 크기 때문에 대부분류로 묶어서 사용한다.

의미정보 유형 및 양상은 Transformation Based Learning(TBL)[11]을 이용하여 추출하고, 격 정보는 휴리스틱 규칙 기법을 이용하여 추출한다. TBL 학습을 위해서 유형 및 양상 정보가 태깅된 채팅 말뭉치 49,312 문장을 사용하였다. 학습 자질은 첫 번째 어절과 두 번째 어절, 마지막 어절과 마지막 이전 어절, 형용사 어절과 주 용언 어절의 6개 어절로부

터 각각 내용어와 형식어의 어휘와 품사의 4가지를 사용하였다. Figure 2는 TBL 학습 자질의 예이다.

**Table 3:** Sentence types in the semantic information

태그	분류	태그	분류
Yn_question	Y/N 의문문	Request	요청
How_question		Assert	명사나열
What_question	WH 의문문	Agree	긍정
When_question		Reject	부정
Where_question		Describe	서술
Who_question		Admire	감탄
Which_question		Greeting	인사
Why_question		Other	기타

**Table 4:** Modalities in the semantic information

태그	분류	태그	분류
Feel	감정	Behave	행동
Experience	경험	Prefer	선호
Guess	추측	Enjoy	즐김
Hope	희망	Opinion	의견
Confirm	확인	Other	기타

**Table 5:** Cases in the semantic information

태그	분류	태그	분류
MainV	주 용언	Agt	주격
Obj	목적격	Bnf	수혜격
Etc	기타		

20=있 21=PA 22=니 23=EF sp=Describe => sp=Yn\_question  
 16=먹 17=PV 18=자 19=ECS sp=Describe => sp=Other  
 0=응 1= 2=e 3=e 5=e 7=e sp=Yn\_question =>sp=Agree  
 16=어떻하 17=PV 18=지 19=EF sp=Describe =>sp=How\_question

**Figure 2:** Examples of TBL features

격 정보 추출을 위해 사용된 패턴은 한국어 문장의 특성을 적용하여 정규 표현식을 이용하여 작성하였다. Table 6은 격 정보 추출 패턴의 일부이다.

어휘 수준 색인 방법은 어휘 정보를 기반으로 저장하기 때문에 문장의 의미를 고려할 수 없다. 이러한 문제를 해결하기 위해 추출한 의미 정보를 기반으로 색인한다. 의미 정보 추출이 끝나면 MainV, Obj, Agt, Bnf, Etc 순으로 추출된 상위 2개의 격 정보를 사용하여 [유형, 양상, 긍/부정, 격정보 의미코드1, 격정보 의미코드2]의 형태로 키를 만들고, [발화번호, 어휘/의미코드/격정보, ...] 형태의 데이터를 만들어 색인 데이터베이스에 저장한다. ‘발화 번호’

는 응답 문장을 검색하기 위해 사용되고, ‘어휘/의미코드/격정보’는 3.2.2의 순위화에서 사용한다. Figure 3은 의미 정보 추출의 예이다.

**Table 6:** Examples of lexico-syntactic patterns for case information extraction

격 정보 추출 패턴	
([ ^ ]+?):[A-Z]##+[ 를을 ]+:JCO	Obj
([ ^ ]+?):[A-Z]##+[은는]:JX	Agt
([ ^ ]+?):[A-Z]##+에게:JCA	Bnf
([ ^ ]+):XPV	MainVerb

문장 : 난 세상에서 우리 학교가 제일 좋아  
 결과 : [decl, none, yes, 좋/%001, 나/@010, 학교/@1101, none/#0, {우리/@010, 세상/#0}]  
 [유형, 양상, 긍/부정, 주용언, 주격, 목적격, 수혜격, {기타}]

**Figure 3:** An example of the semantic information extraction

### 3.2.2 의미 표현 검색 및 후보의 순위화

의미 표현 검색을 위하여 사용자 발화를 2.1의 색인 키와 같은 형태로 변형하여 색인 데이터베이스로부터 후보 문장을 검색한다. 후보 문장이 검색되면 각 후보들의 순위화를 위해 입력 문장과의 유사도 계산을 수행한다. 유사도 점수는 각각의 격 정보에 대하여 최대 1점씩 부여하여 5점 만점으로 계산한다. 기타를 제외한 나머지 격 정보는 ‘어휘/의미코드’ 쌍이 모두 일치 할 경우 1점을 부여하고, 의미코드만 동일할 경우는 1/(의미코드 내 어휘 수+1)점을 부여한다. 기타는 집합이므로 사용자 발화의 의미 표현과 색인 문장의 의미 표현 사이의 Dice’s Coefficient로 계산한다. 계산 결과를 정규화하기 위해 계산에 사용된 격 정보 수로 나누어 준다. 예를 들어 한 후보에 대하여 MainV가 1점, Obj가 0.1점, Etc가 0.2점이 나왔다면 최종 점수는 위 점수를 모두 더한 1.3점을 격 정보의 수 3으로 나눈 0.43점이다.

이와 같이 계산된 유사도 값을 이용하여 순위화를 하고, 가장 유사도가 높은 문장의 응답을 시스템 발화로 사용한다. 이때, 유사도가 임계치(threshold, 본 논문에서는 실험적으로 0.51을 사용. 4.3절에서 설명) 이하일 경우 발화하지 않고, 말 잇기 단계로

넘어간다.

### 3.2.3 패턴 기반 말 잇기

어휘 수준 검색과 의미 수준 검색에서 응답이 생성되지 않을 경우 마지막 패턴 기반 말 잇기에서 처리한다. 매칭 방법은 키워드 매칭과 정규 표현식을 이용한 매칭 두 가지 방법을 사용한다.

키워드 매칭은 동일한 범주 내의 키워드들을 수집하여 다수의 키워드와 다수의 응답 쌍으로 구성된다. **Table 7**은 키워드 매칭 패턴의 예이다.

**Table 7:** Examples of keyword-matching patterns

키워드 매칭 패턴	키워드 매칭 패턴
K슬퍼_	K_하늘이
K슬프다_	K_날씨가
K우울해_	R그래 *
K우울하다	R맞다 *
K울었어	R정말 그렇게
R무슨일이야?	K내 이름은
R힘내 팬찮아질거야	R너가 *?
R문제 있어?	R만나서 반가워

K는 사용자 발화에서 매칭 시킬 키워드를 의미하고, R은 시스템 발화를 의미한다. ‘\_’가 붙어있는 K는 키워드 뒤에 다른 단어가 오면 안 되는 것을 뜻하고, ‘\*’는 키워드가 매칭 되고, 나머지 부분을 응답에 붙여서 발화로 사용하는 것을 의미한다.

정규 표현식을 이용한 매칭은 사용자 발화에서 키워드가 되는 단어를 이용하여 말 잇기를 수행한다. **Table 8**은 말 잇기 정규표현식의 예이다. **Table 7, 8**과 같은 말 잇기 패턴을 이용하여 시스템 발화를 생성한다.

**Table 8:** Examples of lexico-syntactic patterns for story continuation

말 잇기 정규표현식
.* ([^ ]+)?(을 를) 했.+
(1)도 하는구나.
^[ (^ ]+)?는 ([^ ]+)? 먹.+
(1)가 (2) 먹었구나.
어제.* ([^ ]+)?에 ([^ ]+)?[ ]?있더라
(2)이 (1)에는 왜 갔대?
(너 넌) ([^ ]+)?(.)+ 좋.+
나는 (3) 좋아하지

## 4. 실험 및 평가

### 4.1 실험 데이터 및 실험 환경

실험에 사용된 채팅 말뭉치는 실제 모바일 메신저에서 수집된 사용자 발화와 시스템 발화로 구성된 총 49,313개 대화 쌍으로 사용자 발화에 대하여 문장유형, 양상정보가 태깅되어 있다. 평균 어절 수는 4.37어절이고, 주제별로 구분되어 있다.

실험은 색인된 채팅 대화 쌍 중 임의의 사용자 발화 100문장을 선택하여 의미는 동일하지만 문장의 모양을 바꾸어 제안 시스템의 입력으로 넣어 실험한다. 문장의 모양을 바꾸기 위해 선택한 100문장을 컴퓨터 전공자 5명에게 각각 배포하여 변형하도록 하였다. 전체 49,313개로 이루어진 말뭉치를 100문장으로 성능의 객관성을 판단하기는 어렵지만, 5명의 실험자가 수동으로 문장을 변형하고 직접 입력하여 성능을 측정해야 하기 때문에 많은 문장을 입력으로 하기엔 어려움이 있다. **Table 9**는 원본 사용자 발화와 변형된 사용자 발화의 일부를 나타낸 예이다.

**Table 9:** Examples of the modified users' utterances

원본 사용자 발화	변형된 사용자 발화
그래도 난 미니스커트가 더 좋아	그래도 난 치마가 더 예뻐
난 병원 냄새가 너무 싫어	병원에서 나는 향은 너무 안 좋아
얼마 전에 너 친구들이랑 낚시 갔다면?	며칠 전에 친구들이랑 낚시 갔다면?
꼭 비가 올 것 같은 날씨가	곧 비가 오겠는걸

### 4.2 실험 척도 및 평가 방법

제안한 방법의 성능 평가를 위한 실험 척도는 응답 정확률과 사용자 만족도 두 가지로 평가한다. 응답 정확률(Response Precision)은 식 (1)과 같이 계산한다.

$$\text{응답정확률} = \frac{\text{정답으로 간주할 수 있는 응답 수}}{\text{시스템의 응답 수}} \quad (1)$$

사용자 만족도는 **Table 9**와 같이 변형된 100문장을 입력하여 출력되는 시스템 발화의 점수를 측정하였다. 배점은 3점 만점으로 2, 3점을 정답으로 가정하였다. 채점 기준은 시스템 발화가 되지 않는 경우와 시스템 발화는 되나 사용자 발화와 관련 없는

답변일 경우 1점, 자연스럽게 못하나 말을 이어갈 수 있는 경우 2점, 사용자 발화에 대해 자연스럽게 의미가 통하는 시스템 발화가 출력될 경우 3점을 주었다. 채점은 5명의 실험자가 각각 채점하여 합산하여 평균을 구하였다.

### 4.3 실험 결과

Figure 4는 변형된 사용자 발화 100문장을 입력했을 때 정답이라 판단된 발화의 임계치 분포이다.

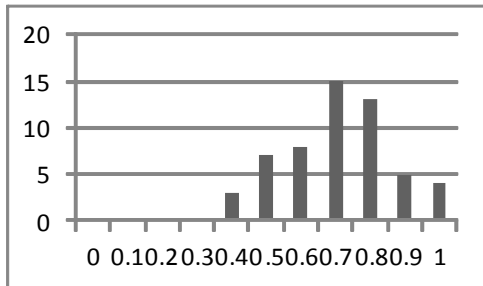


Figure 4: Distribution of the threshold values

Figure 4의 x축은 임계치, y축은 정답 수를 나타낸다. 분포를 살펴보면 0.4 ~ 1.0 사이에 값이 분포되어 있다. 본 논문에서 0.51을 사용한 이유는 0.4의 경우 정답이라 판단된 발화가 있지만 오답 발화를 하는 경우로 인하여 제안 시스템의 성능이 떨어진다. 이를 방지하여 말 잇기 단계에서 처리하기 위해 임계치 0.51을 사용한다.

Table 10은 각 단계별로 실험자들의 사용자 만족도를 합산하여 평균을 구한 결과이다.

Table 10의 결과와 같이 각 단계를 독립적으로 사용할 경우 평균 사용자 만족도가 2점미만으로 기대하는 성능에 미치지 못하였다. Table 11은 실험에 대한 사용자 만족도에 따른 발화의 예이다.

Table 10: User satisfaction scores of the proposed model

실험 방법	전체 실험자 평균	
어휘 수준 검색	1단계	1.07
	2단계	1.29
	3단계	1.61
의미 수준 검색	1.75	
말 잇기	1.41	

Table 11: Chatting examples according to user satisfaction scores

사용자 만족도	발화 예문
1	사용자 : 요즘엔 바깥 날씨가 너무 추워 시스템 : 오늘도 안 지났는데 내일 날씨는 왜? 사용자 : 시간이 지날수록 연락이 안되 시스템 : 응답 없음
2	사용자 : 집에 밥이 없어서 라면 먹을거야 시스템 : 맛있는거 사줘~ 아직 안먹었는데 사용자 : 테니스는 잘하는 편이니? 시스템 : 테니스는 둘이 호흡 맞춰가며 해야되
3	사용자 : 내 동생이 너무 아파서 병원에 갔어 시스템 : 어디가 아픈데? 사용자 : 짜게 먹어서 고혈압 환자들이 많데 시스템 : 짠 음식은 어디든 안좋을 거 같아

각 단계를 독립적으로 사용하지 않고, 조합하여 실험한 결과는 Table 12와 같다. 제안한 방법으로 실험했을 경우 실험 방법 2, 4, 5에서 평균 사용자 만족도가 2점 이상으로 각 단계를 독립적으로 사용했을 때 보다 높은 성능 향상이 있음을 알 수 있다.

Table 12: User satisfaction scores of the combined sub-models

번호	실험 방법	전체 실험자 평균
1	어휘 수준 검색	1.73
2	어휘 수준 검색 + 의미 수준 검색	2.22
3	어휘 수준 검색 + 말 잇기	1.9
4	의미 수준 검색 + 말 잇기	2.05
5	어휘 수준 검색 + 의미 수준 검색 + 말 잇기	2.36

제안 시스템의 응답 정확률을 측정하기 위해 실험자의 사용자 만족도가 2, 3점인 경우를 정답이라 가정한다. 원본 사용자 발화와 변형된 사용자 발화를 입력했을 경우 제안 시스템의 응답 정확률은 Table 13, 14와 같다.

Table 13: Precisions of the proposed system when the as-is utterances are input

실험 방법	응답 정확률
어휘 수준 검색	100%
의미 수준 검색	82%
말 잇기	21%

**Table 14:** Precisions of the proposed system when the modified utterances are input

실험 방법	응답 정확률
어휘 수준 검색	36%
의미 수준 검색	55%
말 잇기	19%
어휘 수준 검색 + 의미 수준 검색	71%
어휘 수준 검색 + 의미 수준 검색 + 말 잇기	78%

실험 결과 원본 사용자 발화의 경우 어휘 수준 검색의 성능이 월등함을 보였다. 그 이유는 의미 수준 검색의 경우 문장에서 의미 정보를 추출하여 색인할 때, 말뭉치 내에서 같은 의미 정보를 가지는 문장들 때문에 나타나는 오류로 확인 되었고, 말 잇기의 경우 패턴 수집 시 어휘, 의미 수준 검색에서 발화가 안 되는 경우를 증점적으로 수집하였기 때문이다. 의미는 동일 하지만 문장의 모양을 변형한 경우 의미 수준 검색의 정확률이 55%로 가장 높았다. 이러한 결과를 보이는 이유는 어휘 수준 검색의 어휘 불일치 문제로 인하여 발생하는 것으로 확인했다. 제한한 3단계 문장 검색 기법의 경우 78%로 제안한 방법의 타당함을 확인 할 수 있는 결과이다.

### 5. 결론 및 향후과제

본 논문에서는 어휘 수준 색인 및 검색 기법에서 생기는 어휘 불일치 문제를 해결하기 위해 3단계 문장 검색 기법과 그것을 위한 색인 방법을 제안했다. 실험 결과 채팅 말뭉치 내에 존재하는 문장의 경우 어휘 수준 색인 및 검색 기법의 성능이 높았다. 하지만 의미는 동일하지만 문장의 모양이 변형된 사용자 발화의 경우는 제안한 기법이 정확률 78%로 월등히 성능이 높음을 확인할 수 있었다. 이는 상대적으로 적은 양의 말뭉치를 이용하여 많은 사용자 발화를 인식할 수 있음을 나타낸다. 향후과제로 실험에 필요한 변형된 사용자 발화를 자동으로 구축할 수 있는 방법을 연구할 예정이다.

### 후 기

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041678, 다중영역 정보서비스

를 위한 대화형 개인 비서 소프트웨어 원천 기술 개발] 또한 2012년도 정부(교육과학기술부)의 지원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2010-0009875).

### 참고문헌

- [1] J.-M. Seo and H.-K. Hwang, "A design and implementation of database for integrated data management on shipboard" Journal of the Korean Society of Marine Engineering, vol. 34, no. 34, pp. 1188-1194, 2010 (in Korean).
- [2] Y.-S. Ha, "The prospects of intelligent robots for marine pollution response", Journal of the Korean Society of Marine Engineering, vol. 26, no. 4, pp. 408-414, 2002 (in Korean).
- [3] Y.-S. Ha and S.-J. Oh, "Multi-sensor fusion as a sensing technique for system intelligence", Journal of the Korean Society of Marine Engineering, vol. 24, no. 2, pp. 151-157, 2000 (in Korean).
- [4] J. Weizenbaum "ELIZA-A computer program for the study of natural language communication between man can machine", Communications of the Association for Computing Machinery, vol. 9, no. 1, pp. 36-46, 1996.
- [5] J.-H. Kim, D.-S Jang, and H.-S Kim, "Statistical generation of korean chatting sentences using multiple feature information", Korean Journal of Cognitive Science, vol. 20, no. 4, pp. 421-437, 2009 (in Korean).
- [6] M. L. Mauldin, "ChatterBots, TinyMuds, and the turing test: entering the Loebner prize competition", Proceedings of the Twelfth National Conference on Artificial Intelligence, vol. 1, pp. 16-21. 1994.
- [7] B. P. Schumaker, Ying Liu, Mark Ginsburg, and Hsinchun Chen, "Evaluating mass knowledge acquisition using the ALICE chatterbot: the AZ-ALICE dialog system", International Journal of Human-Computer Studies, vol. 64, no 11, pp.

1132-1140. 2006.

- [8] Jason L. Hutchens, Michael D. Alder, "Introducing MegaHAL", NeMLaP3 / CoNLL98 Workshop on Human-Computer Conversation, Association for Computational Linguistics, pp. 271-274, 1998.
- [9] D. Vrajitoru, "Evolutionary sentence building for chatterbots", Genetic and Evolutionary Computation Conference, pp. 315-321, 2003.
- [10] D. Vrajitoru, J. Ratkiewicz, "Evolutionary sentence combination for chatterbots", The IASTED International Conference on Artificial Intelligence and Applications, pp. 287-292, 2004.
- [11] J. R. Curran, R. K. Wong, "Formalization of transformation-based learning", Australasian Computer Science Conference, pp. 51-57, 2000.