이전 문장 자질과 다음 발화의 후보 화행을 이용한 한국어 화행 분석

김세종[O] 이용훈 이종혁
포항공과대학교 전자컴퓨터공학부 컴퓨터공학과, 첨단정보기술 연구센터
{sejong[O], yhlee95, jhlee}@postech.ac.kr

Korean Speech Act Tagging using Previous Sentence Features and Following Candidate Speech Acts

Se-Jong Kim[O] Yong-Hun Lee Jong-Hyeok Lee
Dept. of Computer Science and Engineering, Division of Electrical and Computer Engineering, POSTECH
Advanced Information Technology Research(AITrc)

요 약

화행 분석이란 자연언어로 된 발화를 통해서 나타나는 화자의 의도를 인식하는 것으로 대화를 처리하는 여러 응용 분야에서 중요하게 요구되는 과정이다. 기존의 연구에서는 이전 발화의 화행과 현재 발화의 문장 자질을 사용하여 규칙 기반 및 통계 기반의 연구가 진행되었다. 하지만 본 논문에서는 현재 발화 이후의 후보 화행을 추정하여 이를 현재 발화의 화행을 결정하는데 사용함으로써 기존의 연구와 차별화를 두었으며, 실제로 기존 방법보다 약 4%의 성능향상을 보인 96.08%의 정확도(accuracy)를 보였다. 또한 이전 발화의 화행 정보 대신 이전 화행을 결정하는데 사용했던 문장 자질을 현재 발화의 화행 결정에 직접 사용함으로써 실제 이전 화행을 적용했을 때보다 향상된 성능을 가져왔으며, 이전 문장 자질과 후보 화행을 함께 사용하여 화행을 결정했을 경우 96.96%의 정확도를 보였다.

1. 서론

화행이란 화자의 발화를 통해서 나타나는 화자의 의도를 가리키는 것으로 이를 올바르게 인식하는 과정을 화행 분석이라 한다. 화행 분석은 대화를 처리하는 여러 응용 분야에서 매우 중요하게 요구되는 과정으로 특히 목적 지향적 에이전트를 구현하는데 있어서 없어서는 안 되는 요소이다. 현재 목적 지향적 에이전트로 활용되고 있는 분야는 기차 예약 에이전트, 항공 예약 에이전트, 인터넷 쇼핑 도우미 등이 있으며 이러한 대화 에이전트는 사용자가 입력한 발화로부터 해당 도메인에 필요한 정보를 추출하고 화행 분석을 통한 화자의 의도를 파악한 뒤, 시스템 상에서 정의된 대화 전략에 따라 응답을 생성한다[1, 2, 3].

그러나 화자는 자신의 의도가 담긴 발화를 다양한 표현으로 상대방에게 전달할 수 있으며 이러한 특성은 시스템이 화자의 의도를 완벽하게 파악한다는 것이 무척 어렵다는 것을 보여준다[4]. 실제 연구 분야에서도 특정 도메인에 한정된 화행 분석은 비교적 높은 성능을 보인 반면, 도메인에 독립적인 화행 분석은 상당히 낮은 성능을 보였다[5, 6, 7]. 이는 화행 분석의 한계성을 나타냄과 동시에 이에 대한 더 많은 연구가 요구됨을 뜻한다. 화행의 종류를 결정하는 것도 화행 분석의 성능에 영향을 미치지만 도메인에 독립적인 화행을 결정하는 것 또한 대화 처리 분야의 큰 이슈로서 연구되고 있으므로 따로 언급하지 않겠다[8].

화행 분석은 '대화를 처음 처리하는 단계'라는 입장에서 볼 때 매우 높은 성능을 보여야한다. 자연언어 처리의 중요 분야인 형태소 분석이 그러하듯이 초기 단계의 성능이 높을수록 이후 단계의 성능이 향상될 수 있기 때문이다. 기존의 연구에서는 이전 발화의 화행과 현재 발화의 문장 자질을 사용하여 규칙 기반 및 통계 기반의 연구가 진행되었다. 하지만 본 논문에서는 현재 발화 이후의 발화에 대한 후보 화행을 추정하여 이를 현재 발화의 화행을 결정하는데 사용함으로써 기존 연구 결과에 대한 성능향상을 꾀했다. 또한 이전 발화의 화행 정보 대신 이전 화행을 결정하는데 사용했던 문장 자질을 현재 발화의 화행 결정에 직접 사용함으로써 이전 발화에서 추정된 정보의 사용을 최소화하였다.

본 논문의 2장에서는 화행 분석에 관련된 기존의 연구들을 살펴보고 3장에서는 제안하고자 하는 두 가지 방법에 대해서 자세하게 설명한다. 4장에서는 실제 사용자와 항공예약전문가 간의 대화내용을 기록한 SRI ATIS Collection[9]의 일부를 번역 및 정제한 코퍼스를

사용하여 실험결과를 제시하고 5장에서 결론을 맺는다.

2. 관련연구

2.1 규칙 기반 모델

규칙 기반 모델이란 시스템 설계자가 화행을 결정하기 위한 규칙들을 직접 작성하는 것으로, 해당 도메인의 지식을 포함하는 언어 정보 규칙과 문맥 규칙을 사용하여 화행을 결정하는 방법이다. 대표적인 방법으로는 화행 분석 단계를 표층적 화행 분석과 심층적 화행 분석으로 나누어 해당 발화의 후보 화행을 점차 줄여가는 것이다[10]. 여기서 표층적 화행 분석이란 문맥을 고려하지 않고 표층적으로 나타난 정보만을 가지고 모든 후보 화행을 결정하는 것을 말하며, 심층적 화행 분석은 대화의 흐름을 고려한 문맥 정보를 사용하여 표층적 화행 분석에서 결정된 화행들 중 하나의 화행만을 선택하는 과정이다. 심층적 화행 분석에서 사용된 문맥 정보는 담화 지식만으로 구현된 재귀적 대화 전이망(RDTN, Recursive Dialogue Transition Network)과 심층적 화행 결정 규칙으로 이루어진다. 이러한 규칙 기반 모델은 화행 결정에 있어서 매우 높은 성능을 가져왔으나 도메인 지식을 획득하는데 많은 시간이 소모되고, 다른 도메인으로 확장할 경우 해당 도메인에 대한 규칙을 다시 구축해야 하는 단점을 가지고 있다.

2.2 통계 기반 모델

통계 기반 모델은 규칙 기반 모델의 단점을 보완한 것으로, 화행 분석을 위한 규칙들을 직접 작성하는 대신 대량의 대화 코퍼스를 사용하여 기계학습을 하고 학습된 결과를 바탕으로 각 발화의 화행을 결정하는 방법이다. 본 모델은 기본적으로 다음과 같은 식(1)을 토대로 설계되었으며 S_i는 i번째 화행을, U_i는 i번째 발화를 의미한다[5, 11].

$$S(U_{1\cdots n}) = \underset{S_{1\cdots n}}{\operatorname{argmax}} P(S_{1\cdots n}|U_{1\cdots n}) \approx \underset{S_{1\cdots n}}{\operatorname{argmax}} \prod_{i=1}^{n} P(S_i|S_{1\cdots i-1}) P(U_i|S_i) \quad (1)$$

통계 기반 모델은 식(1)을 이루는 각각의 확률값을 다른 형태로 변형하거나 어떠한 방법으로 학습할 것인가에 초점을 맞추고 연구되었다. N-gram, 최대 엔트로피 모델(Maximum Entropy Model), 은닉 마코프 모델(Hidden Markov Model), 결정 트리 모델(Decision Tree Model), 신경망 모델(Neural Network Model) 등이 기존 연구에서 활용된 기계학습 방법이라고 할 수 있겠다[5, 7, 11, 12].

또한 학습 방법 이외에 학습에 사용되는 자질을 선택하는 방법에 대한 연구가 진행되었는데 대표적인 방법으로는 식(2)와 같은 카이제곱 통계량(CHI Statistics)을 사용하여 자질과 화행간의 정보량을 측정한 후, 화행 결정에 큰 영향을 미치는 효과적인 자질을 선택하는 방법이 있다[7].

$$\chi^2(F,S) = \frac{(a+b+c+d)\times(ad-cb)^2}{(a+c)\times(b+d)\times(a+b)\times(c+d)} \quad (2)$$

a : 자질 F를 가지고 화행 S로 부여된 발화 수
b : 자질 F를 가지고 화행 S로 부여되지 않은 발화 수
c : 자질 F를 가지지 않고 화행 S로 부여된 발화 수
d : 자질 F를 가지지 않고 화행 S로 부여되지 않은 발화 수

2.3 자질의 종류와 한국어 화행

기존 연구에서 사용된 문장 자질 또는 구문 정보는 문장 형태, 본용언, 보조용언, 시제, 부정형 여부, 단서 단어 등이 있으며 문맥 자질로서는 이전 화행, 화자 정보 등이 있다[11, 12].

표 1. 문장 자질

종류	예
문장 형태	평서문, 명령문, Wh의문문, YN의문문
본용언	'알'(동사), '돕'(동사), '어렵'(형용사) 등
보조용언	'드리', '주', '되', '하', '싶', '않' 등
시제	과거, 현재, 미래
부정형 여부	유, 무
단서 단어	'네', '아니요', '안녕히', '감사', '매진' 등

한국어 화행의 종류는 설계자의 판단에 따라 다양하게 정의될 수 있지만 기존의 연구에서 대표적으로 사용된 화행은 표 2와 같다[7, 11, 12].

표 2. 한국어 화행

화행	설명 (예)
Introducing-oneself	자신을 소개 (홍길동이라고 하는데요.)
Opening	대화를 시작 (무엇을 도와드릴까요?)
Closing	대화를 마침 (감사합니다. / 수고하세요.)
Accept	승인 (네. / 알겠습니다.)
Acknowledge	상대방이 계속 말하도록 호응 (네. / 788에. / 그리고요?)
Ask-confirm	앞에서 발화된 내용을 확인 (홍길동이라고요?)
Ask-if	Ask-confirm을 제외한 YN의문문 (유나이티드 항공입니까?)
Ask-ref	Wh의문문 (언제 출발하시길 원하십니까?)

Request	특정 행동을 요구 (예약 좀 해주세요.)
Expressive	감정을 표현 (좋네요. / 기쁩니다.)
Correct	상대방의 발화에 대한 수정 (아니요, 그 항공편은 무착륙 항공편입니다.)
Reject	거절 (어렵겠습니다.)
Promise	특정 행동에 대한 약속 (본 내용으로 예약하겠습니다.)
Inform	독립적으로 정보 제공 (샌프란시스코를 경유합니다.)
Offer	부가 (확인해보겠습니다.)
Suggest	제안 (왕복편은 어떠십니까?)
Response	응답 (5월 2일이요. / 예.)

본 논문에서는 위와 같은 문장 자질과 문맥 자질, 그리고 17개의 화행을 사용하며, 특히 단서 단어에 대해서는 실험에 사용된 대화 코퍼스의 형태소 분석 결과로부터 추출된 단어/품사 정보를 식(2)의 카이제곱 통계량을 사용하여 순위화한 후, 해당 발화에 존재하는 최상위 단어/품사 정보를 선택하도록 하였다.

3. 제안하는 방법

3.1 다음 발화의 후보 화행 추정

현재 발화의 화행을 결정하기 위해서는 식(1)을 통해 알 수 있듯이 이전 화행과 현재 발화의 문장 자질을 알아야 한다. 하지만 대화가 화행의 흐름을 벗어나거나 약간의 변화를 일으키기만 해도 이전 화행으로부터 추정할 수 있는 현재 발화의 화행 정보는 불명확하게 된다. 그럼 다음과 같은 상황을 생각해보자.

U : 사용자 / A : 에이전트

■ 예 1.
U : "5월 2일에 출발하고 싶습니다." **(Request)**
A : "(평서문 / 단서 단어 : '알'(동사))"

■ 예 2.
U : "5월 2일에 출발하고 싶습니다." **(Request)**
A : "(평서문 / 단서 단어 : '알'(동사))"
U : "네." **(Accept)**

그림 1. 상황 예시

예 1을 살펴보면 사용자가 5월 2일에 여행을 떠나고 싶으니 해당 정보를 찾아달라는 '요구'를 하고 있음을 알 수 있다. 그럼 에이전트는 사용자의 요구에 따라 해당 정보를 찾아보겠다는 '승인' 및 '부가'적인 발화를 할 것이다. 즉, 에이전트의 대답이 평서형이고 단서 단어가 동사인 '알'이라는 것을 알고 있으므로 해당 발화가 '승인'일 경우에는 '알겠습니다.'와 같은 발화를, '부가'적인 표현일 경우에는 '해당 항공편을 알아보겠습니다.'와 같은 발화를 했을 것으로 예상할 수 있다. 예 2는 예 1과 동일한 발화가 진행된 후에 사용자가 '승인'의 의미를 가진 발화를 하는 상황이다. 에이전트의 화행이 'Accept'일 때 이후 사용자의 화행이 'Accept'일 수는 없으므로 이러한 사실을 통해 에이전트의 화행이 'Accept'가 아닌 'Offer'라는 것을 알 수 있다.

이렇게 현재 발화 이후의 발화에 대한 화행 정보를 알고 있다면 화행을 결정하는데 있어서 본 정보가 유용하게 사용될 수 있다. 하지만 실제 대화에서 현재 발화 이후의 화행을 정확하게 알 수 있는 방법은 없다. 다만 다음 화행으로서 추정되는 상위 후보 화행들을 활용하여 현재 화행을 결정하는데 도움을 줄 수 있을 것이다. 즉, 후보 화행들과 연결이 가장 자연스러운 화행이 현재 화행으로서 가능성이 높다는 것이다. 그림 2는 후보 화행을 추정하는 방법과 추정된 후보 화행이 현재 화행에 영향을 주는 모습을 시각적으로 표현한 것이다.

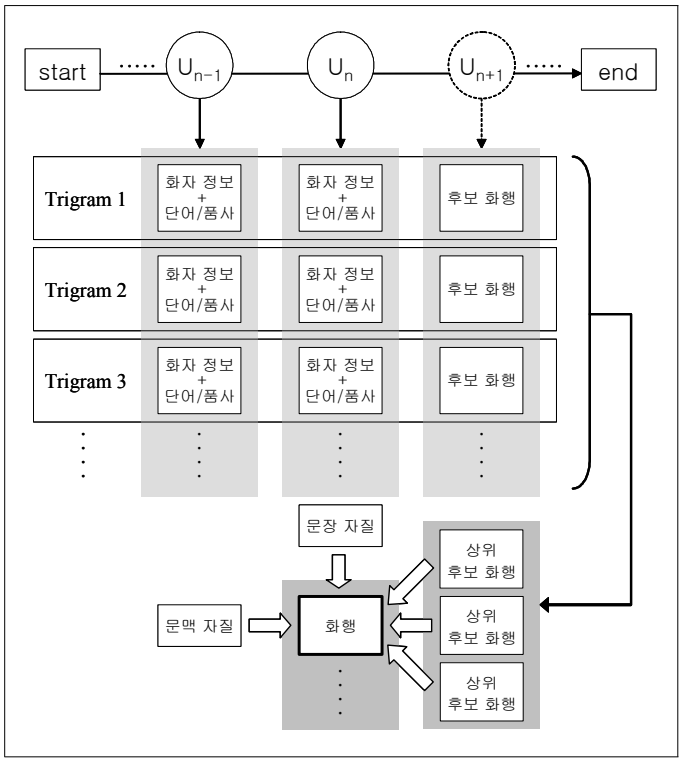

그림 2. 후보 화행 추정

후보 화행을 구하기 위해서는 현재 발화 이전의 화행 정보를 사용하는 대신 이전 발화의 화자 정보와 단어/품사 정보를 사용한다. 이는 이전 화행을 시스템이 잘못 결정했을 경우 후보 화행을 추정하는데 있어서 악영향을 미치기 때문이다. 후보 화행을 추정하는 과정을

자세히 설명하면, 먼저 대화 코퍼스의 형태소 분석 결과와 화자 정보를 추출하여 '화자 변경 정보'를 포함한 '이전 발화의 단어/품사' + '현재 발화의 단어/품사' + '다음 화행'으로 이루어진 trigram 데이터를 만든다. 그리고 각각의 trigram 데이터에 대한 카이제곱 통계량을 구하여 순위화한 후, 구하고자하는 화행의 이전 발화와 현재 발화의 단어/품사 정보와 일치하는 trigram 데이터를 통계량이 큰 것부터 선택하여 해당 데이터의 마지막 항목인 다음 화행 정보를 후보 화행으로서 결정한다. 표 3은 대화 코퍼스로부터 추출된 trigram 데이터와 카이제곱 통계량의 일부를 기록한 것이다.

표 3. Trigram 데이터와 카이제곱 통계량

순위	Trigram 데이터	화자 변경 유무	카이제곱 통계량
1	blank(blank)/ 익스프레스(CMCN)/**Opening**	유	4119.68
9	수고(CMCPA)/.(g)/**end**	유	3292.06
11	감사(CMCPA)/.(g)/**Closing**	유	3222.80
12	감사(CMCPA)/하(fpd)/**Closing**	유	2864.63
77	익스프레스(CMCN)/ 무엇(CTP3)/**Introducing-oneself**	무	1348.56
164	으로(fjcao)/알(YBDO)/**Promise**	유	389.63
262	이(fpd)/비싸(YBHO)/**Suggest**	유	322.77
708	.(g)/ㅂ니까(fmofq)/**Response**	유	212.63
962	예약(CMCPA)/알(YBDO)/**Promise**	유	159.50
1390	안녕(CMCPS)/려고(fmocs)/**Ask-ref**	무	122.49

위 표에서 등장하는 'blank(blank)'는 이전 발화가 존재하지 않을 경우 임시로 사용하는 단어/품사 정보이며 'end'는 다음 발화가 없을 경우에 사용하는 임시 화행 정보이다. 후보 화행은 이전 발화와 현재 발화의 단어/품사 정보에 따라 중복되어 선택될 수 있으며 현재 문장의 문장 자질이 충분한 정보를 제공해주지 못할 경우 높은 효과를 발휘한다. 물론 현재 발화의 문장 자질이 화행을 결정하기 위한 충분한 정보를 제공할 경우에는 오히려 화행 결정에 방해가 될 수 있다.

3.2 이전 발화의 문장 자질 활용

선행된 발화의 화행 정보는 현재 발화의 화행을 결정하는데 큰 영향을 미친다. 기존 연구에서는 식(1)의 첫 번째 확률값을 결정하기 위해 선행된 모든 발화의 화행을 참고하는 대신 현재 발화와 가장 근접하고 직접적으로 영향을 미치는 이전 화행만을 선택하여 참고한다. 식(3)은 현재 발화를 결정하기 위해 기존 연구에서 사용하는 확률 모델을 표현한 것이다[11].

$$S(U_n) \approx \underset{S_n}{\operatorname{argmax}} P(S_n|S_j, S_k)P(F_n|S_n) \quad (3)$$

여기서 S_n은 현재 화행을, S_j와 S_k는 담화 구조(Discourse Structure)를 통해 추정된 현재 화행에 직접적으로 영향을 미치는 가장 근접한 2개의 화행을, F_n은 현재 발화의 문장 자질을 의미한다. 또한 담화 구조를 화행 분석 없이 미리 파악한다는 것은 어려운 문제이므로 j와 k를 n-2와 n-1로 치환하여 사용할 수 있다. 하지만 기존 연구에서는 이러한 확률 모델을 적용할 경우 이전 발화의 화행을 정확하게 알고 있다는 가정 하에서 성능을 측정하거나, 이러한 가정 없이 시스템이 분석한 이전 화행 정보를 적용하되 오히려 선행된 발화의 잘못된 화행 결정이 이후 화행에 영향을 주어 오류가 발생한 경우가 있었다.

이전 화행의 추정으로부터 나타날 수 있는 이러한 문제점을 배제하기 위해 본 논문에서는 이전 발화의 문장 자질을 현재 발화의 화행 결정에 직접 사용함으로써 이전 발화에서 추정된 정보의 사용을 최소화하였다. 이전 화행을 결정하는데 사용했던 문장 자질은 잠재적으로 해당 발화의 화행 정보를 포함하고 있으며 발화 자체에 대한 특성을 가지고 있기 때문에 추정된 화행보다 유용하게 사용할 수 있다. 식(4)는 이와 같은 사실을 바탕으로 식(3)을 변경한 것이다.

$$S(U_n) \approx \underset{S_n}{\operatorname{argmax}} P(S_n|F_{n-2}, F_{n-1})P(F_n|S_n) \quad (4)$$

앞에서도 언급했듯이 문장 자질은 기존 연구에서 사용된 문장 형태, 본용언, 보조용언, 시제, 부정형 여부, 단서 단어를 사용하였고, 단서 단어에 대해서는 표 4와 같이 카이제곱 통계량을 통해 순위화하여 해당 발화에 존재하는 최상위 단어/품사 정보를 선택하고 이에 해당하는 화행 정보를 화행 결정에 참고하도록 하였다.

표 4. 단서 단어와 카이제곱 통계량

순위	단서 단어와 화행	카이제곱 통계량
1	무엇(CTP3)/**Opening**	1162.06
2	겠(fmbtf)/**Accept**	1118.73
3	안녕히(SBO)/**Closing**	1094.88
5	수고(CMCPA)/**Closing**	875.67
15	잠시(CMCN)/**Offer**	388.60
17	매진(CMCPA)/**Reject**	382.53
43	어디(CTP3)/**Ask-ref**	187.43
76	부탁(CMCPA)/**Request**	106.40
147	습니까(fmofq)/**Ask-if**	50.76
370	그리고(SBJ)/**Inform**	13.41

이밖에도 발화간의 화자 변경 정보를 문장 자질과 함

께 사용하여 현재 화행을 결정하는데 도움을 주었다.

4. 실험결과 및 분석

본 논문에서는 실제 사용자와 항공예약전문가 간의 대화내용을 기록한 SRI ATIS Collection의 일부를 번역 및 정제한 코퍼스를 사용한다. 이 코퍼스는 총 1479개의 발화와 18개의 대화로 구성되어 있으며 각 발화에 대한 문장 자질과 형태소 분석 결과, 그리고 정답 화행이 부여되어있다. 형태소 분석 결과는 포항공과대학교 지식 및 언어공학 연구실에서 개발한 형태소 분석기(KoMA)를 사용하였다. 실험 데이터의 세부 정보는 표 5와 같다.

표 5. 실험 데이터

화행	개수	화행	개수
Introducing-oneself	33	Expressive	55
Opening	27	Correct	3
Closing	36	Reject	15
Accept	181	Promise	18
Acknowledge	102	Inform	232
Ask-confirm	49	Offer	31
Ask-if	119	Suggest	21
Ask-ref	132	Response	338
Request	87		

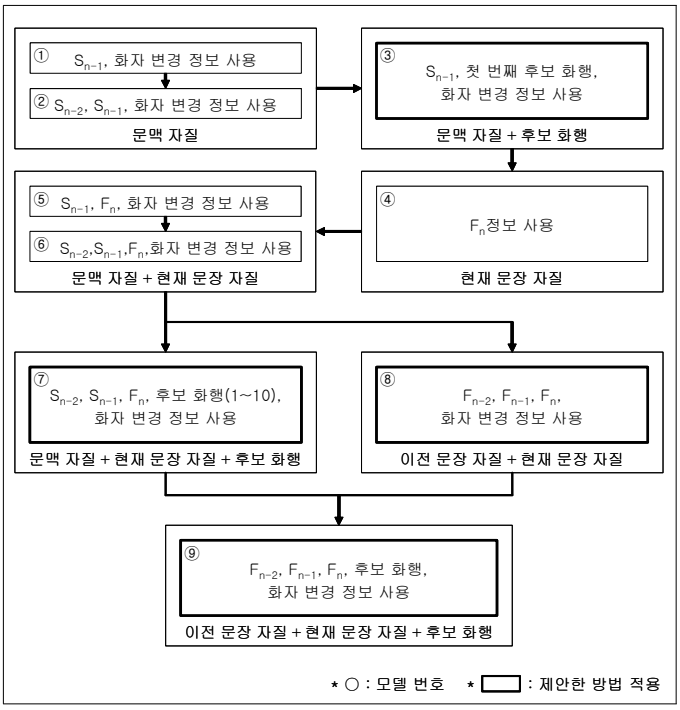

그림 3. 실험 순서

실험은 앞에서 소개한 문장 자질, 문맥 자질, 후보 화행, 정답 화행 정보를 대화 코퍼스로부터 추출하고 이를 데이터화하여 입력 자질이 많은 경우에도 안정적인 성능을 보이는 SVM(Support Vector Machine)기법을 적용하였다. 실제 실험은 자바(Java)기반 기계학습 알고리즘 툴인 Weka를 활용하여 학습 및 성능평가를 수행하였다[13]. 그리고 실험 데이터들의 수가 비교적 적은 편이므로 학습 데이터와 평가 데이터를 정적으로 분리하지 않고 총 실험 데이터의 10%를 평가 데이터로 하고 남은 90%를 학습 데이터로 하여 성능평가를 수행한 후, 같은 방식으로 평가 데이터와 학습 데이터를 순차적으로 갱신하여 전체 성능평가를 수행하는 10-fold cross validation방법을 사용하였다.

본 실험은 각각의 자질 정보가 화행 결정에 어떠한 영향을 미치는지 확인하고 기존의 방법과 제안한 방법에 대한 비교성능평가를 수행하기 위해 그림 3과 같은 순서로 진행하였다.

4.1 이전 화행 / 후보 화행 성능평가

S_{n-1}을 사용한 경우(모델1), S_{n-2}와 S_{n-1}을 사용한 경우(모델2), S_{n-1}과 다음 발화의 첫 번째 후보 화행을 사용한 경우(모델3)에 대한 성능평가 결과는 표 6과 같다.

표 6. 모델별 성능평가 결과 1 (단위 : %)

화행	F-Measure		
	모델1	모델2	모델3
Introducing-oneself	83.1	82.7	88.0
Opening	68.9	87.5	76.2
Closing	55.6	98.6	57.1
Accept	45.0	47.3	60.9
Acknowledge	38.8	37.7	44.7
Ask-confirm	11.5	21.1	11.5
Ask-if	0.0	17.3	46.6
Ask-ref	1.4	29.9	56.8
Request	0.0	50.3	11.2
Expressive	19.0	50.5	22.0
Correct	0.0	0.0	100.0
Reject	0.0	0.0	0.0
Promise	0.0	0.0	42.6
Inform	56.2	58.8	72.3
Offer	0.0	0.0	16.7
Suggest	0.0	25.0	0.0
Response	91.2	91.6	91.0
정확도(accuracy)	**50.57**	**55.78**	**61.87**

모델1은 1개의 이전 화행만 사용하여 현재 화행을 결정하였기 때문에 Introducing-oneself와 Response와 같은 대화의 흐름에 의존적인 화행에 대해서만 좋은 성능을 보였다. 모델2는 2개의 이전 화행을 사용하여 현재 발화의 화행을 결정하였기 때문에 모델1보다 전반적으로 높은 성능을 보였다. 하지만 대화의 흐름에 의존적인 Opening과 Closing만이 끄게 올랐을 뿐 Correct, Reject, Promise, Offer와 같은 문장 자질에도 크게 영

향을 받는 화행에 대해서는 효과를 보지 못했다.

모델3은 본 논문의 3장에서 제안했던 다음 발화에 대한 후보 화행을 추정하여 적용한 것으로 카이제곱 통계량을 통해 순위화된 후보 화행 정보로부터 최상위에 위치한 후보 화행을 선택하여 실험에 반영한 것이다. 본 모델 역시 모델2보다 전반적인 성능향상을 보였고 특히 기존 모델에서 효과를 보지 못했던 화행인 Correct, Promise, Offer의 성능이 향상되었다. 하지만 Opening과 Closing같은 대화의 흐름에 보다 의존적인 화행에 대해서는 오히려 성능이 떨어지는 결과를 보였다.

4.2 이전 화행 / 현재 문장 자질 / 후보 화행 성능평가

F_n을 사용한 경우(모델4), S_{n-1}과 F_n을 사용한 경우(모델5), S_{n-2}와 함께 S_{n-1}과 F_n을 사용한 경우(모델6)에 대한 성능평가 결과는 표 7과 같다.

표 7. 모델별 성능평가 결과 2 (단위 : %)

화행	F-Measure		
	모델4	모델5	모델6
Introducing-oneself	91.7	95.7	95.7
Opening	87.5	94.1	94.1
Closing	95.7	100.0	100.0
Accept	83.0	93.4	94.3
Acknowledge	14.3	87.0	90.7
Ask-confirm	28.1	45.2	61.9
Ask-if	82.8	78.7	86.7
Ask-ref	95.0	94.2	96.4
Request	79.3	92.2	98.9
Expressive	74.8	81.9	84.5
Correct	75.0	85.7	100.0
Reject	74.1	88.9	92.9
Promise	10.0	27.3	25.0
Inform	53.1	85.8	89.6
Offer	90.3	91.5	95.1
Suggest	83.3	87.8	92.3
Response	65.9	94.2	95.7
정확도	**71.06**	**88.24**	**91.62**

모델4는 현재 문장 자질만을 가지고 성능을 평가한 것으로 이전 화행에 의존적인 Acknowledge, Response와 같은 화행을 제외한 나머지 화행에 대해서 좋은 성능을 보였다. 모델5는 1개의 이전 화행과 현재 문장 자질을 가지고 성능을 평가하였는데 모델4보다 매우 높은 성능을 보였으며, 2개의 이전 화행을 현재 문장 자질과 함께 사용한 모델6 또한 보다 높은 성능향상을 보였다. 이러한 성능향상은 통계 기반 모델을 구현하기 위해 갖추어야할 기본적인 정보들을 모두 사용하였기 때문임과 동시에, 실험 데이터 자체의 특성 때문이라고 할 수 있겠다. 즉, 본 실험에서 사용된 코퍼스 자체가 항공예약이라는 특정 도메인에 한정되어있고 비교적 적은 양의 발화로 인해 대화의 흐름 및 각 발화에서 사용된 문장 자질의 양이 한정되어있어서 이와 같은 높은 성능이 나타난 것으로 판단된다.

그럼 제안한 방법을 본 코퍼스에 적용한 실험에 대해서 살펴보자. 표 8은 모델6에 추정된 후보 화행을 상위 순위부터 하나씩 추가해가면서 성능을 평가한 것이다.

표 8. 후보 화행 수별 성능평가 결과 (단위 : %)

추가한 상위 후보 화행 수	정확도
1	92.49
2	93.17
3	95.33
4	95.27
5	95.61
6(모델7)	**96.08**
7	95.94
8	**96.08**
9	95.74
10	95.88

실험결과를 통해 알 수 있듯이 현재 발화의 화행을 결정하기 위해 후보 화행을 추가할 경우 서로 중복될 수 있는 6개 또는 8개의 후보 화행을 선택하였을 때 가장 높은 성능을 보였다. 즉, 1개나 2개의 후보 화행을 선택하기보다는 좀 더 많은 후보 화행을 선택하는 것이 성능향상에 도움이 되며, 특정 양을 초과하여 후보 화행을 선택할 경우 성능에 부정적인 영향을 미친다는 것을 알 수 있었다.

표 9. 모델6·7간의 성능평가 결과 비교 (단위 : %)

화행	F-Measure		성능향상
	모델6	모델7	
Introducing-oneself	95.7	95.7	0.0
Opening	94.1	94.1	0.0
Closing	100.0	100.0	0.0
Accept	94.3	97.2	2.9
Acknowledge	90.7	95.3	4.6
Ask-confirm	61.9	84.8	22.9
Ask-if	86.7	92.7	6.0
Ask-ref	96.4	100.0	3.6
Request	98.9	96.6	-2.3
Expressive	84.5	99.1	14.6
Correct	100.0	100.0	0.0
Reject	92.9	100.0	7.1
Promise	25.0	100.0	75.0
Inform	89.6	95.7	6.1
Offer	95.1	93.5	-1.6
Suggest	92.3	89.5	-2.8
Response	95.7	96.6	0.9
정확도	**91.62**	**96.08**	**4.46**

표 9는 기존의 화행 결정 방법(모델6)[11]과 제안한

방법(모델7)의 성능을 비교한 것으로 전체 정확도는 4.46% 향상되었고 몇 가지 화행을 제외한 나머지 화행에 대해서도 전반적인 성능향상을 보였다. 특히 기존의 방법은 Promise를 Inform으로, Ask-confirm을 Ask-if로 잘못 판단하는 경우가 대부분이었으나 제안한 방법을 통해서 이를 구분하기위한 단서를 보다 많이 제공함으로써 각 화행의 성능이 서로 향상되는 결과를 가져올 수 있었다. 한편, Request, Offer, Suggest는 오히려 성능 감소를 보였는데 Request를 Offer와 Response로, Suggest를 Inform로 잘못 판단하여 발생한 결과로서, 추정된 이후 화행보다 현재 문장 자질이 해당 화행들을 결정하는데 보다 효과적임을 보여주는 예라고 할 수 있겠다. 이러한 점을 보완하기 위해서는 특정 화행에 큰 영향을 미치는 자질에 대해서 화행 결정을 위한 가중치를 부여할 필요성이 있다.

4.3 이전 문장 자질 / 현재 문장 자질 성능평가

F_{n-2}와 F_{n-1}, 그리고 F_n을 사용한 경우(모델8)와 기존의 모델6과의 성능평가 결과를 비교하면 표 10과 같다.

표 10. 모델6·8간의 성능평가 결과 비교 (단위 : %)

화행	F-Measure		성능향상
	모델6	모델8	
Introducing-oneself	95.7	92.5	-3.2
Opening	94.1	90.6	-3.5
Closing	100.0	100.0	0.0
Accept	94.3	93.5	-0.8
Acknowledge	90.7	79.6	-11.1
Ask-confirm	61.9	85.1	23.2
Ask-if	86.7	93.5	6.8
Ask-ref	96.4	96.6	0.2
Request	98.9	95.4	-3.5
Expressive	84.5	90.2	5.7
Correct	100.0	100.0	0.0
Reject	92.9	100.0	7.1
Promise	25.0	94.1	69.1
Inform	89.6	88.9	-0.7
Offer	95.1	91.5	-3.6
Suggest	92.3	92.3	0.0
Response	95.7	95.8	0.1
정확도	**91.62**	**92.49**	**0.87**

모델8은 실제 이전 화행을 적용했을 때(모델6)보다 향상된 성능을 가져왔으며 이는 이전 발화의 문장 자질이 이전 화행보다 유용한 정보로 활용될 수 있는 가능성을 보여준다. 물론 이전 화행에 의존적인 Introducing-oneself, Opening, Acknowledge와 같은 화행에 대해서는 성능이 감소되었으나 이러한 문제는 이전 문장 자질의 종류를 보완해감으로써 해결할 수 있을 것이라고 생각한다.

4.4 이전·현재 문장 자질 / 후보 화행 성능 평가

기존의 모델8에 모델7의 후보 화행 정보들을 함께 고려한 것(모델9)과 모델7과의 성능평가 결과를 비교하면 표 11과 같다.

표 11. 모델7·9간의 성능평가 결과 비교 (단위 : %)

화행	F-Measure		성능향상
	모델7	모델9	
Introducing-oneself	95.7	92.5	-3.2
Opening	94.1	90.6	-3.5
Closing	100.0	100.0	0.0
Accept	97.2	98.1	0.9
Acknowledge	95.3	95.7	0.4
Ask-confirm	84.8	94.8	10.0
Ask-if	92.7	96.3	3.6
Ask-ref	100.0	99.6	-0.4
Request	96.6	98.3	1.7
Expressive	99.1	98.2	-0.9
Correct	100.0	100.0	0.0
Reject	100.0	100.0	0.0
Promise	100.0	100.0	0.0
Inform	95.7	95.4	-0.3
Offer	93.5	91.5	-2.0
Suggest	89.5	92.3	2.8
Response	96.6	97.8	1.2
정확도	**96.08**	**96.96**	**0.88**

제안한 두 가지 방법을 함께 적용한 모델9는 모델6과 모델8간의 성능향상치와 거의 동일한 성능을 추가적으로 낼 수 있었다. 그림 4는 지금까지 수행한 실험결과들을 종합하여 그래프로 나타낸 것이다.

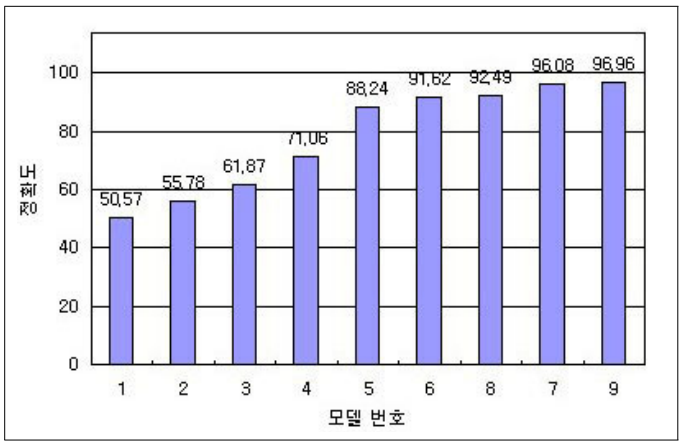

그림 4. 성능평가 결과 그래프

5. 결론

대화 에이전트를 구현하기 위해서는 자연언어처리기술에 바탕을 둔 다양한 연구 분야들의 적용이 필요하

며, 각각의 분야들은 자신의 성능을 최대한 발휘함으로써 성공적인 대화 에이전트를 구현할 수 있다. 본 논문은 대화 처리 분야에서 중요한 위치를 차지하고 있는 화행 분석에 대하여 다루었으며, 현재 발화 이후에 나타날 수 있는 후보 화행들을 추정하고 이전 발화의 문장 자질을 이전 화행 대신 사용함으로써 기존의 연구 방법과 차별화를 둠과 동시에 성능향상을 꾀했다. 실제로, 제안한 방법을 통해 얻어진 결과는 기존의 결과보다 향상된 성능을 보였으며 이는 현재 화행 결정에 대한 이후 화행의 유용성과 추정되지 않은 실제 문장 자질의 명확성에 의한 것으로 판단된다.

하지만 앞의 실험결과에 대한 분석에서 언급했듯이 실험에 사용된 코퍼스가 특정 도메인, 즉 항공예약 도메인에 한정되어있고 비교적 적은 양의 발화로 이루어져있기 때문에 대화의 흐름이 강한 방향성을 가지게 되었다. 발화를 이루고 있는 형태소 및 화행 결정에 큰 영향을 미치는 단서 단어도 화자의 다양한 발화들을 포괄할 수 있을 만큼 다양하지 못해서 해당 실험에 대한 기본 성능이 높은 것으로 여겨진다. 다시 말하면, 한정된 자원으로 인한 대용량 대화 코퍼스에서의 성능평가를 수행하지 못해 제시한 방법론에 대한 보편적인 효용성을 보일 수 없었다. 향후에 대용량 대화 코퍼스를 사용하여 현재의 방법론의 유용성과 개선점, 그리고 보다 선구적인 방법론에 대한 연구가 모색되어야 할 것이다.

감사의 글

본 연구는 첨단정보기술 연구센터를 통한 과학재단 및 2007년도 두뇌한국21사업의 지원을 받았고 정통부 및 정보통신연구진흥원의 정보통신선도기반기술개발사업의 연구결과로 수행되었습니다.

참고 문헌

[1] J. Allen, et al., "The TRAINS Project: A case study in building a conversational planning agent," Journal of Experimental & Theoretical Artificial Intelligence, vol. 7, no. 1, pp. 7-48, 1995.

[2] L. Lambert and S. Carberry, "A tripartite plan-based model of dialogue," Proceedings of the 29th annual meeting on Association for Computational Linguistics, pp. 47-54, 1991.

[3] J. Chu-Carroll and S. Carberry, "Response generation in collaborative negotiation," Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pp. 136-143, 1995.

[4] C. Doran, et al., "Comparing Several Aspects of Human-Computer and Human-Human Dialogues," Current and New Directions in Discourse and Dialogue, pp. 133-159, 2003.

[5] A. Stolcke, et al., "Dialogue act modeling for automatic tagging and recognition of conversational speech," Computational Linguistics, vol. 26, no. 3, pp. 339-373, 2000.

[6] E. Ivanovic, "Dialogue act tagging for instant messaging chat sessions," Proceedings of the ACL Student Research Workshop, pp. 79-84, 2005.

[7] 김경선, 서정연, “자질 선택 기법을 이용한 한국어 화행 결정,” 정보과학회논문지 : 소프트웨어 및 응용, vol. 30, no. 3, pp. 278-284, 2003.

[8] M. Core and J. Allen, "Coding Dialogs with the DAMSL Annotation Scheme," Proceedings of the AAAI Fall Symposium on Communicative Action in Humans and Machines, pp. 28-35, 1997.

[9] SRI ATIS Collection, http://www.ai.sri.com/~communic/amex/amex.html

[10] 이현정, 이재원, 서정연, “자동통역을 위한 한국어 대화 문장의 화행 분석 모델,” 정보과학회논문지(B), vol. 25, no. 10, pp. 1443-1452, 1998.

[11] W. Choi, et al., "An Integrated Dialogue Analysis Model for Determining Speech Acts and Discourse Structures," IEICE Transactions on Information and Systems, vol. E88-D, no. 1, pp. 150-157, 2005.

[12] S. Lee and J. Seo, "Korean Speech Act Analysis System Using Hidden Markov Model with Decision Trees," International Journal of Computer Processing of Oriental Languages, vol. 15, no. 3, pp. 231-243, 2002.

[13] I. Witten and E. Frank, "Weka - Machine Learning Algorithms in Java," Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations, pp. 264-321, 2000.