한국지능시스템학회 논문지 2010, Vol. 20, No. 6, pp. 741-747
DOI : 10.5391/JKIIS.2010.20.6.741

2단계 문장 추출 방법을 이용한 회의록 요약

Meeting Minutes Summarization using Two-step Sentence Extraction

이재걸 · 박성배 · 이상조
Jae-Kul Lee, Seong-Bae Park and Sang-Jo Lee

경북대학교 컴퓨터공학과

요 약

본 논문은 회의록의 특징을 반영한 회의록을 요약 방법을 제안한다. 회의록은 일반 문서와 달리 회의의 진행자가 전체 흐름을 주도하고, 회의 진행에 사용하는 단어들이 존재하며, 발언자들 간의 대화에 종속관계가 있다는 특징이 있다. 제안한 방법은 먼저 회의의 흐름을 찾기 위해 사전에 구축된 회의 진행에 특화된 단어사전과 TextRank 알고리즘을 사용하여 진행자의 주제 문장들을 추출한다. 다음으로 추출된 문장들을 회의록에 있는 참석자들의 문장과 유사도를 계산하여 회의의 주제 문장과 관련있는 중요 문장을 추출한다. 마지막으로 사용자가 흐름을 편히 알 수 있도록 추출된 문장들 사이에 종속관계를 분석하여 최종적으로 회의록을 요약한다. 국회 전자회의록을 대상으로 실험한 결과, 제안한 방법이 회의록을 요약하는 비율 전 구간에서 기존의 요약 방법들보다 더 나은 성능을 보인다.

키워드 : 문서요약, 회의록 요약, 문장 추출

Abstract

These days many meeting minutes of various organizations are publicly available and the interest in these documents by people is increasing. However, it is time-consuming and tedious to read and understand whole documents even if the documents can be accessed easily. In addition, what most people want from meeting minutes is to catch the main issues of the meeting and understand its contexts rather than to know whole discussions of the meetings. This paper proposes a novel method for summarizing documents considering the characteristics of the meeting minutes. It first extracts the sentences which are addressing the main issues. For each issues expressed in the extracted sentences, the sentences related with the issue are then extracted in the next step. Then, by transforming the extracted sentences into a tree-structure form, the results of the proposed method can be understood better than existing methods. In the experiments, the proposed method shows remarkable improvement in performance and this result implies that the proposed method is plausible for summarizing meeting minutes.

Key Words : Text Summarization, Meeting Minutes Summarization, Sentence Extraction

1. 서 론

국회 및 지방의회, 공공기관 및 일반기업에서는 많은 회의를 하고 진행된 회의의 내용을 회의록 형태로 기록하여 보관한다. 최근에는 기록된 회의록을 전자문서 형태로 변환하여 인터넷에 공개함으로써 일반인들이 쉽게 접근하여 그 내용을 볼 수 있다[1]. 하지만 회의록은 보관의 용도로 작성되어 회의의 시작에서 끝까지 모든 의사에 관한 발언을 총망라하여 게재한 것으로 일반인들이 회의의 전체적인 흐름이나 대략적인 내용을 파악하기에는 적합하지 않다. 따라서 회의록의 주요 내용을 유지하면서 전체의 흐름을 쉽게 파악할 수 있도록 하는 문서요약 방법이 필요하다.

문서요약이란 문서가 담고 있는 핵심 의미를 유지하면서 문서의 크기를 효과적으로 줄여 축약버전을 생성하는 작업으로 기존에 많은 관련 연구들이 있었다[2, 3, 4]. 하지만 이러한 연구들은 문서 전체를 하나의 주제로 보고 요약을 하기 때문에 하나의 문서에서 여러 세부 주제들이 나타나는 회의록의 요약에는 적합하지 않다. 회의록을 요약하기 위해서는 회의록의 특징을 반영한 요약 방법이 필요하다.

회의록은 회의가 진행되면서 나타나는 참석자들의 모든 발언을 시간 순서대로 기록한 문서로 일반적인 문서와는 달리 다음의 세 가지 특징이 있다. 첫째, 회의의 진행에 따라 여러 세부 주제들이 나타나고, 그 흐름을 진행자가 주도한다. 회의록은 회의가 진행됨에 따라서 하나의 안건에 대해서 내용설명, 의견제시, 토론 등과 같은 여러 세부 주제들이 나타난다. 그리고 이런 세부 주제들은 진행자의 발언에 따라 변한다. 둘째, 회의의 흐름을 판단하는데 중요한 역할을 하는 단어들이 존재한다. 세부 주제를 변화시키는 진행자의 발언에

접수일자 : 2010년 10월 21일
완료일자 : 2010년 11월 29일
"본 논문은 본 학회 2010년도 추계학술대회에서 선정된 우수논문입니다."
감사의 글 : 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 휴먼인지환경사업본부-신기술융합형 성장동력사업의 지원을 받아 수행된 연구임(No. 2010K001130)

는 “상정”, “의사진행발언”, “반대토론” 등과 같은 단어들이 존재한다. 위 단어들은 회의록의 세부 주제를 변화시키는 단어로 회의의 흐름을 파악하는데 중요한 역할을 한다. 셋째, 회의록은 참석자들 사이의 대화를 기록한 문서로 대화들 사이에 종속관계가 나타난다. 진행자의 발언 이후에 나오는 참석자들의 발언은 진행자의 발언에 종속적이고, 참석자들의 발언 사이에서도 종속적인 관계가 나타날 수 있다.

본 논문에서는 앞서 살펴본 특징들을 반영한 회의록 요약 방법을 제안한다. 제안하는 방법은 먼저 회의록의 세부 주제별로 중요 문장을 추출하여 문서 크기를 줄인다. 이를 위해 회의록에서 진행자의 발언 문장들만을 대상으로 문장의 중요도 값을 계산하고, 회의의 흐름을 판단하는데 중요한 역할을 하는 단어들이 포함되어 있을 경우 중요도 값에 가중치를 적용하고 세부 주제문장을 선별한다. 다음으로 각 주제별로 해당 주제와 관련이 높은 문장들을 추출한다. 최종적으로 전체적인 흐름을 쉽게 파악할 수 있도록 추출된 중요 문장들 사이에 종속관계를 분석하여 트리 형태로 회의록을 요약한다.

본 논문의 2장에서는 문서요약에 관한 기존의 관련 연구들을 알아보고, 3장에서는 본 논문에서 제안하는 회의록의 특징을 반영한 회의록 요약에 대하여 설명한다. 4장에서는 회의록 요약 방법의 평가를 위한 실험 결과를 기술하고, 마지막으로 5장에서는 결론을 맺고 향후 연구 과제를 검토한다.

2. 관련연구

문서요약이란 주어진 문서의 기본적인 내용을 유지하면서 문서의 복잡도, 즉 문서의 크기를 줄이는 작업이다[14]. 문서요약은 크게 추출요약과 생성요약으로 나눌 수 있다[3]. 추출요약은 문서에서 중요하다고 생각하는 일부분 (중요 구/문장/절)을 선택하여 제공하는 것이다. 이에 비해 생성요약은 추출요약 방법에 원본 문서의 내용을 더 간결하게 바꿔 쓰는 (Paraphrasing) 작업이 추가된다. 즉, 생성 요약은 추출 요약보다 원문을 더 강력하게 축약한다고 볼 수 있다. 그러나 이러한 작업은 자연어 생성 기술을 필요로 하고, 이는 아직은 성숙되지 않은 기술 분야라는 문제가 있어 상대적으로 접근과 구현이 쉬운 추출요약에 관한 연구가 주로 이루어지고 있다.

기존의 문서요약 방법으로 단어의 빈도, 문장의 위치와 같은 문서의 형태에서 나타나는 통계적인 정보를 이용하여 문서를 요약한 연구들이 있었다. Luhn[7]은 문서의 주제를 표현하는 단어는 자주 사용된다는 직관에 의거하여 가장 많이 사용된 단어를 문서의 주제어라고 보았다. Edmundson[8]은 문장의 위치에 따라 문장의 중요도가 다르다는 점을 연구하였으며, 아울러 ‘significant’, ‘hardly’, ‘impossible’ 과 같은 단서어도 중요 문장을 파악하는데 중요한 역할을 할 수 있음을 보였다. 이 외에 통계적 특징을 학습을 통해 습득함으로써 성능 향상을 꾀한 연구[9]도 있었으나, 이런 방법들은 지나치게 단순한 통계에 의존함으로써 문서의 주제를 파악하는데 한계를 보였다.

정보검색의 질의확장 기법을 이용한 요약방법이 있다. 질의 기반의 문서요약이란 사용자 의도에 맞는 요약 결과를 얻기 위하여 문서의 내용을 포괄적으로 요약하기 보다는 사용자에게 중심이 되는 단어, 즉 사용자가 관심을 갖는 정보에 근거하여 요약을 하는 방법이다[2,3]. Sanderson[3]은 INQUERY 검색 시스템의 지역적 문맥분석 (local context analysis)을 이용하여 주어진 사용자 질의에 대하여 사용자 주도 요약을 생성하였다. 하지만 질의확장을 사용하지 않은 요약 방법에 비해 성능 향상을 보이지 못했다. 이에 대해 Goldstein et al.[2]은 의사 적합성 피드백이나 문서의 제목, 첫 문장 등의 자질을 추가하여 다양하게 질의를 확장하는 방법을 제안하였다.

최근에는 문서에 포함된 문장들로 그래프를 만들어 문서를 그래프로 보고 중요한 문장을 찾는 요약 방법에 관한 연구가 있었다. Mihalcea와 Rarau[4]는 Brin과 Page[5]의 PageRank 알고리즘의 개념을 텍스트 문서에 적용한 TextRank 알고리즘을 제안하였다. 이 알고리즘은 문장 간의 연결 그래프를 만들어 요약하는 방법으로 문장 간에 공통 단어로 문장들 사이의 연결 그래프를 만든 다음 PageRank 알고리즘의 개념을 도입하여 중요 문장을 추출한다. TextRank 알고리즘은 비지도 (Unsupervised) 학습을 통해 문서요약이 가능하고 언어의 제약 없이 다양한 언어의 문서요약에 사용될 수 있다는 장점이 있다. 또한, 최근에 연구된 논문에 따르면 TextRank 알고리즘은 한국어 문서요약에서 높은 성능을 보이는 것으로 나타났다[6]. 본 논문에서 회의록의 세부 주제를 찾기 위해 진행자의 발언 문장들의 중요도 값을 계산할 때 이 TextRank 알고리즘을 사용하였다. TextRank 알고리즘에 관한 자세한 내용은 본 문에서 다시 다루도록 한다.

3. 회의록 요약방법

본 논문에서 제안하는 회의록 요약 방법은 그림 1과 같이 두 가지 과정으로 수행된다. 중요문장 추출과정은 회의록의 크기를 축약하는 과정으로 회의록의 주요 내용을 포함하고 있는 중요한 문장들을 추출하는 과정이고, 트리 생성과정은 회의록 전체의 흐름을 쉽게 파악할 수 있도록 추출된 중요문장들을 트리형태로 구축하는 과정이다.

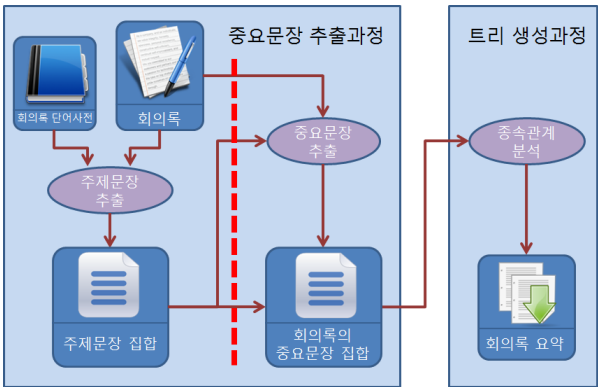

그림 1. 회의록 요약의 흐름도
Fig. 1. Procedure of Proposed Summarization Model

중요문장 추출과정은 2단계로 나누어 수행한다. 먼저, 주제문장 추출단계에서는 회의록으로부터 진행자의 발언 문장들만을 대상으로 회의록 단어사전을 고려하여, 주제로 적합한 문장들을 추출한다. 다음으로 중요문장 추출단계에서는 요약할 회의록과 주제문장 추출단계에서 생성된 주제문장들을 입력으로, 먼저 회의록 전체를 주제별로 분리하고, 각 주제별로 주제와 관련이 있는 문장을 추출한다.

트리 생성과정은 중요문장 추출과정에서 생성된 회의록의 중요문장 집합을 입력으로 받는다. 회의록의 중요문장 집합의 문장들을 주제별로 분리하여 노드를 생성하고 각 주제 안에서 문장들의 종속관계를 분석하여 트리에서 노드 문장들의 위치를 결정한다. 회의록의 중요문장 집합의 모든 문장들에 대한 위치가 결정되면 이를 트리형태로 표현하여 회의록 요약문을 생성한다.

3.1 중요문장 추출과정

3.1.1 주제문장 추출단계

이 단계에서는 회의록과 회의록 단어사전을 입력으로 진행자의 발언 문장으로부터 회의록의 세부 주제를 찾고, 찾아진 결과를 이용하여 주제문장 집합을 생성한다. 입력으로 받는 회의록은 시간 순서대로 기록된 발언 내용들로 이루어져 있고, 각 발언 내용은 발언자와 발언문장으로 구성된다. 그리고 회의록 단어사전은 회의의 흐름을 파악하는데 중요한 역할을 하는 단어들의 집합으로 이루어져 있다.

회의록의 세부 주제를 찾기 위해 입력받은 회의록의 발언 내용들 중 발언자가 진행자인 문장들을 선택하여 진행자의 발언문장 집합을 생성하고, 생성된 진행자의 발언문장 집합에 TextRank 알고리즘을 이용하여 문장들의 중요도 값을 계산한다. 회의록은 기존에 구축되어 있는 학습데이터가 적고 학습데이터를 직접 구축하기 위해서는 비용이 많이 들기 때문에 문장의 중요도 값 계산에 지도 (Supervised) 학습방법을 사용하기 어렵다. 따라서 본 논문에서는 비지도 (Unsupervised) 학습방법 중 높은 성능을 보이는 TextRank 알고리즘[4]을 사용한다.

TextRank 알고리즘을 사용하기 위해서는 문서를 그래프 형태로 변환하여야 한다. 본 논문에서는 문서에 포함된 문장들을 노드로 보고 서로 다른 문장에서 같은 단어가 나오는 문장들을 연결하여 그래프를 생성한다. 이렇게 생성된 그래프에 TextRank의 문장 중요도를 계산하는 식 (1)을 이용하여 각 문장들의 중요도 값을 계산한다.

$$TR(S_i) = (1-d) + d \cdot \sum_{S_j \in In(S_i)} \frac{w_{ji}}{\sum_{S_k \in Out(S_j)} w_{jk}} TR(S_j) \quad (1)$$

식 (1)에서 $TR(S_i)$는 문장 S_i의 TextRank 값을 의미하고, d는 현재 문장에서 다른 문장으로 이동할 확률 (damping factor) 값으로 Brin과 Page[5]가 제안한 값 0.85를 사용한다. $In(S_i)$는 문장 S_i로 들어오는 연결을 가지는 문장들의 집합이고, $Out(S_j)$는 문장 S_j에서 나가는 연결을 가지는 문장들의 집합이다. 마지막으로 w_{ij}는 문장 S_i와 문장 S_j 사이의 연결 가중치를 의미한다. 본 논문에서는 문장 S_i와 문장 S_j 사이에 공통적으로 존재하는 단어들의 빈도수를 이용하여 유사도 식 (2)를 정의하였으며, 이 식으로 계산된 유사도 값을 가중치 w_{ij}로 사용한다.

$$Similarity(S_i, S_j) = \frac{|\{n_k | n_k \in S_i \wedge n_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \quad (2)$$

단어 n_k는 문장 S_i와 문장 S_j 사이에 공통으로 포함된 단어들을 의미한다. 본 논문에서는 문장에 포함된 단어들 중에서 명사만을 사용한다. 그 이유는 기존의 연구에 따르면 그래프 기반의 한국어 문서요약에서는 문장의 유사도 계산에 명사만을 사용하는 것이 가장 높은 성능을 보였기 때문이다[6, 12]. 이 식에서 $|S_i|$ 는 문장 S_i의 단어 개수를 의미하고 문장에 포함된 단어가 많을수록 유사도 값이 커지는 것을 막기 위해 각 문장의 단어의 개수에 대해 log를 취해 나누어 주었다.

최종적으로 본 논문에 사용된 문장의 중요도를 계산하는 식은 다음과 같이 정리된다.

$$TR(S_i) = (1-d) + d \cdot \sum_{S_j \in Edge(S_i)} \frac{Similarity(S_i, S_j)}{\sum_{S_k \in Edge(S_j)} Similarity(S_j, S_k)} TR(S_j) \quad (3)$$

본 논문에서 생성한 그래프는 무향 그래프 (Undirected Graph)이므로 원래 TextRank 식에서 사용된 문장 S_i로 들어오는 문장들의 집합 $In(S_i)$와 문장 S_j에서 나가는 문장들의 집합 $Out(S_j)$는 문장 S_i와 연결된 문장들의 집합인 $Edge(S_i)$로 대체되었다. 또한, 문장 S_i와 문장 S_j 사이의 연결 가중치 w_{ij} 역시 마찬가지로 식 (2)로 대체하였다.

회의록에는 진행자의 발언들 중 세부 주제를 변화시키는 발언에 자주 포함되는 중요한 의미를 가지는 단어들이 존재한다. 반대로 중요한 의미 없이 상투적으로 쓰이는 단어들이나 불용어와 같은 단어들도 존재한다. 회의록 단어사전은 이러한 단어들을 모아서 긍정 단어와 부정 단어로 분리하여 구축하였다. 회의록 사전에 포함된 단어가 해당 문장에서 나타날 경우에 계산되어진 문장의 중요도 값에 가중치 값을 문장의 중요도 계산에 반영한다. 이 가중치 값은 긍정 단어의 경우에는 양수 값을 가지고, 부정 단어의 경우에는 음수 값을 가진다. 이렇게 가중치까지 모두 적용한 진행자의 발언 문장의 중요도 값은 식 (4)와 같이 계산된다.

$$Score(S_i) = \begin{cases} (1+p) \cdot TR(S_i) & n_k \in D \wedge n_k \in S_i \\ TR(S_i) & otherwise. \end{cases} \quad (4)$$

식 (4)에서 D는 회의록 단어사전이고 p는 단어의 가중치를 나타낸다. 단어 n_k가 회의록 단어사전 D와 문장 S_i에 공통으로 포함되어 있을 경우 문장의 중요도 값 $TR(S_i)$에 해당 가중치 값 $1+p$를 곱한다. 문장 S_i에 회의록 단어사전 D의 단어가 포함되어 있지 않은 경우는 기존에 계산된 문장의 중요도 값 $TR(S_i)$를 그대로 사용한다.

회의록 단어사전의 가중치까지 모두 반영한 문장들의 중요도 값 $Score(S_i)$를 내림차순으로 정렬하고, 상위 x퍼센트의 문장을 추출함으로써 주제문장 집합을 생성한다. 이 때, 매개변수 x는 최종적으로 생성할 요약문의 요약 비율이다.

본 단계에서는 중요문장 추출단계에서 세부 주제별 중요문장들을 찾기 위해 주어진 회의록을 주제별로 분리한다. 그림 2는 추출한 주제 문장들을 주어질 때, 회의록을 주제별로 분리하는 과정을 나타낸다.

왼쪽 (실선)과 같은 회의록이 존재할 때 원본 회의록에서 추출되지 않은 진행자의 발언 문장들을 제거하면 오른쪽 (점선)과 같은 형태의 문서가 생성된다. 이렇게 생성된 문서에서 추출된 진행자의 중요문장부터 다음에 추출된 진행자의 중요문장 이전까지가 하나의 주제로 분리된다.

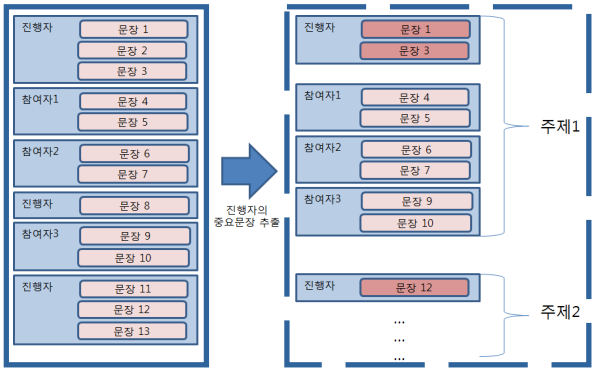

그림 2. 세부 주제의 분리
Fig. 2. Separate topics

회의에서 발언을 할 때는 일반적으로 한 번에 여러 문장을 발언한다. 예를 들어, 회의에서 상대편의 의견에 대한 반대토론을 할 때 그 내용을 한 문장으로 발언하는 것이 아니라 여러 문장에 걸쳐서 발언하게 된다. 이렇게 같은 발언자가 연속으로 발언하는 문장들을 하나의 집합으로 묶어 발언 기회로 정의한다. 진행자의 중요 문장을 주제별로 분리하면 하나의 주제에 진행자의 중요 문장이 둘 이상 포함이 될 수 있다. 하지만 이렇게 같은 발언 기회에 나온 문장들은 서로 관련된 주제에 관해 발언한 문장들이므로 이들을 묶어 하나의 주제 문장으로 집합을 생성한다.

3.1.2 중요문장 추출단계

이 단계에서는 주제문장 추출단계에서 생성된 주제문장 집합과 회의록을 입력으로 각 주제별로 해당 주제와 관련이 있는 문장만을 추출한다. 이 단계를 통해 진행자의 흐름 즉 회의 세부 주제와 관련이 있는 참석자들의 발언들을 찾고, 주제에 따르도록 요약한다. 입력으로 받는 회의록은 주제문장 추출단계에서의 회의록과 같고, 주제문장 집합은 이전 단계에서 생성된 주제별로 묶인 진행자의 중요문장들의 집합이다.

본 논문에서는 세부 주제문장과 회의록의 참석자 문장들과의 유사도를 측정하기 위해, 문서 표현에 널리 사용되는 bag of word 모델과 동일하게 문장을 단어의 집합으로 간주하고 이를 벡터로 표현한다. 이 벡터의 유사도를 계산함으로써 문장의 유사도를 계산한다. 두 벡터의 유사도는 식(5)의 코사인 유사도 (Cosine Similarity)를 사용하여 계산한다.

$$Sim(S_i, V_j) = \frac{S_i \bullet V_j}{\| S_i \| \| V_j \|} \quad (5)$$

식 (5)에서 S_i는 벡터로 표현된 진행자의 발언 문장을 제외한 참석자의 문장이고, V_j는 벡터로 표현된 주제문장이다. 또한, $\| S_i \|$는 벡터 S_i의 크기 (Norm)이다. 주제문장 추출단계와 동일하게 계산된 유사도 값 $Sim(S_i, V_j)$을 내림차순으로 정렬하여 상위 x퍼센트의 문장을 추출한다. 그리고 앞서 추출한 진행자의 중요문장들과 본 단계에서 추출한 중요문장들을 발언 순서대로 정렬하여 최종적으로 회의록의 중요문장 집합을 생성한다.

3.2 트리 생성과정

회의록 문서의 크기를 줄이는 과정인 중요문장 추출과정을 모두 마치면 문서의 표현을 다르게 하여 전체적인 내용과 흐름을 쉽게 파악할 수 있도록 하는 트리 생성과정을 진행한다. 이 과정은 이전 과정에서 생성한 회의록의 중요문장 집합을 입력으로 각 문장들 사이의 종속 관계를 분석하여 트리 형태로 표현한다. 입력으로 받는 회의록의 중요문장 집합은 이전 과정에서 회의록에서 중요하다고 판단된 문장들이 발언 순서대로 정렬된 형태의 문서이다.

일반적으로 회의록에서 발언 문장은 이전에 발언한 문장들에 대한 응답이므로 본 논문에서는 종속관계 분석은 이전에 나온 문장들과 관계만 분석한다. 또한, 종속관계 분석의 단위를 발언한 각 문장이 아닌 각 발언 기회별로 분리된 문장 집합으로 한다. 앞서 정의했듯이 회의록의 발언 문장들은 한 번의 발언 기회에 한 문장만을 발언하는 것이 아니라 여러 문장을 발언한다. 그러므로 입력받은 회의록의 중요문장 집합도 같은 발언 기회에 두 개 이상의 문장이 중요문장으로 선택될 수 있다. 이런 문장들을 각 발언 기회별로 분리하여 발언 집합 $M = \{M_k | k = 1,2,...,n\}$을 생성한다. 이렇게 발언 기회별로 분리된 문장들은 같은 발언 기회 M_k에 포함된 문장 $S_i \in M_k$라면 이 문장들은 그 이전의 발언 기회들 $M_k (n < k)$중 같은 하나의 발언 기회와 종속관계가 있다고 가정한다. 각 발언 기회별로 그 이전에 나오는 발언 기회들과 유사도를 측정하여 종속관계의 분석한다. 그림 3은 이러한 유사도 측정을 기반으로 트리를 생성하는 과정을 기술한 의사코드이다.

```
1: Input - 회의록의 중요문장 집합 I
2: M = combine(I)              // M = {M_k | k = 1,2,...,n}
3: T = tree
4: root = root node of T
5: for each M_k ∈ M do
6:     if isModerator(M_k) then
       // M_k가 진행자의 발언 기회이면 참 아니면 거짓
7:         root.addChild(M_k)
8:     else
9:         A = ancestor(M_{k-1})
           // root 노드로부터 목표 노드까지 경로에 있는 노
           드들의 집합을 리턴
10:        node = argmax_{A_j} Sim(A_j, M_k)
11:        node.addChild(M_k)
12:    end if
13: end for
14: return T
```

그림 3. 트리 생성과정 의사코드
Fig. 3. Pseudo code of tree construction procedure

트리에서 노드의 위치를 결정할 때 진행자의 발언 기회이면 $root$의 바로 하위 노드로 추가한다. 진행자의 발언 기회가 아니면 이전 발언 기회의 노드에서 $root$까지 경로에 존재하는 모든 노드를 가져와 유사도를 비교한 후 가장 유사도가 높은 노드의 하위 노드로 추가한다. 위 과정을 모두 수행하면 트리형태의 회의록 요약본이 생성되며, 그 예로 그림 4와 같다.

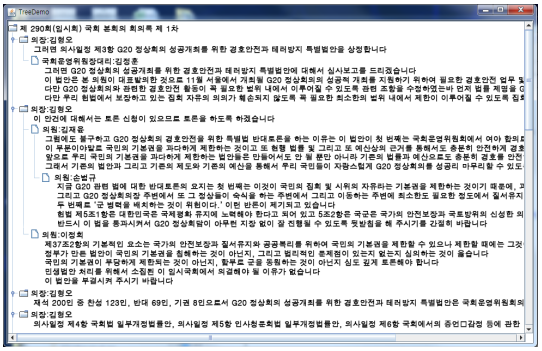

그림 4. 제안한 방법을 이용하여 생성한 회의록 요약문의 일부분

Fig. 4. Part of result meeting minutes summary of using proposed method

위 그림은 본 논문에서 제안한 방법을 사용하여 생성한 회의록의 요약문 중 일부분이다. 생성한 요약문은 추출된 중요 문장들이 각 발언 기회별로 트리 형태로 표현되어 있어 전체 흐름을 쉽게 파악할 수 있다. 즉, 전체의 큰 흐름을 진행자(의장)의 중요 문장들로 파악할 수 있고, 각 세부 주제별로 참가자들의 중요 발언을 볼 수 있다. 또한, 추출된 중요문장들을 살펴보면 회의의 중요 흐름들이 진행자의 문장에서 추출되었고, 다른 참가자들의 문장은 해당 주제별로 진행자의 문장과 관련된 문장들이 많이 추출된 것을 볼 수 있다.

4. 실험 및 평가

4.1 실험 데이터

제안한 요약 시스템의 성능 평가를 위해 대한민국 국회 회의록 시스템[1]의 전자 회의록 문서를 사용하였다. 본 논문에서는 제 18대 국회 본회의 제 276회 회의록부터 제 291회 회의록까지 총 56개의 회의록 중에서 36개의 회의록에 대하여 평가를 수행하였다. 실험에 사용한 회의록의 평균 문장 수는 약 1,524 문장이고 평균 단어 수는 약 10,932 단어이다. 국회 회의록 시스템에서는 회의록에 대한 요약본을 제공하지 않으므로 해당 36개의 회의록에 대해서 수작업으로 요약문을 생성하여 정답 요약문으로 사용하였고, 요약 비율은 20%-25%로 하였다. 정답으로 생성된 요약문은 참석자들의 발언 기회별로 요약된 문장과 해당 발언자 정보, 그리고 전체 정답 트리 구조가 포함되어 있다.

4.2 성능 평가 함수

본 논문에서 제안한 방법은 문서의 크기를 줄이는 과정과 문서의 표현을 변화시키는 과정으로 나누어지므로 각 과정에 맞게 따로 평가를 해야 한다. 우선 문서의 크기를 줄이는 중요문장 추출과정에 관한 성능은 DUC[10]에서 평가 방법으로 사용되고 있는 ROUGE-N[11]과 정보 검색에서 평가 방법으로 사용되는 F-measure의 두 가지 방법으로 측정한다. 그리고 문서의 표현을 변화시키는 트리 생성과정에 관한 성능은 트리의 각 노드에 대하여 *root* 노드에서 목표 노드에 이르기까지의 경로를 비교하여 그 정확도를 측정하는 *accuracy* 식을 정의하여 성능을 평가하였다.

먼저 단어를 얼마나 정확하게 추출 하였는지를 평가하기 위해 ROUGE-N 평가 방법을 사용하였다. ROUGE-N 평가 방법의 경우 식 (6)을 이용하여 제안한 방법이 생성한 요약문에 포함된 단어들과 정답 요약문에 포함된 단어들의 정확도를 측정한다.

$$ROUGE-N=\frac{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S}Count(gram_n)} \quad (6)$$

$gram_n$은 $n-gram$의 단어이고, $Count(gram_n)$은 정답 요약문의 $n-gram$의 개수이며, $Count_{match}(gram_n)$은 제안한 방법이 생성한 요약문과 정답 요약문이 동시에 발생한 최대 $n-gram$의 수 이다. 정리하면, ROUGE-N 평가 방법은 생성 요약문과 정답 요약문에서 발생한 $n-gram$에 대한 재현율 (recall) 값이다. ROUGE-N 평가 방법은 N의 값에 따라 $n-gram$이 결정 되는데, 본 논문에서는 $uni-gram$을 측정하는 ROUGE-1 평가 방법을 이용하여 성능을 측정하였다.

다음으로 문장을 얼마나 정확하게 추출하였는지를 평가하기 위해 F-measure 방법을 사용하였다. F-measure는 정확률 (precision)과 재현율 (recall)을 하나의 값으로 표현한다. F-measure는 식 (7)과 같이 정의된다.

$$F_\beta=(1+\beta^2)\cdot\frac{precision\cdot recall}{\beta^2\cdot precision+recall} \quad (7)$$

β는 정확률과 재현율 사이의 중요도를 설정하는 인자로, 본 논문에서는 정확률과 재현율을 같은 중요도로 보는 F_1-measure를 사용하였다.

마지막으로 트리 생성의 성능을 평가하기 위한 방법으로 *accuracy* 식을 정의하여 사용하였다. 이 식은 정답 요약문의 트리와 제안한 방법으로 생성된 트리에 공통으로 나타나는 노드에 대해서 *root* 노드로부터 해당 노드까지 경로가 모두 같은 노드의 비율을 측정하는 방법으로 식 (8)과 같이 정의된다.

$$accuracy=\frac{\sum_{n\in\{N_R\cap N_A\}}I(P(n,T_R),P(n,T_A))}{|N_R\cap N_A|} \quad (8)$$

식 (8)에서 N_R은 제안한 방법으로 생성된 트리의 노드 집합이고, N_A는 정답 요약문 트리의 노드 집합이다. I는 표시함수 (indicator function)로 두 인자의 값이 같으면 1, 다르면 0을 반환한다. 함수 $P(n,T)$는 트리 T에서 노드 n의 경로를 반환하는 함수로 *root* 노드로부터 노드 n까지 가는 경로에 포함된 모든 노드들의 집합을 반환한다.

4.3 실험 및 결과

중요문장 추출과정에 관한 제안한 방법의 성능 평가를 위해 두 가지 베이스라인과 비교하였다. 첫 번째 베이스라인은 TextRank 알고리즘[4]만을 이용한 요약 방법으로, 모든 실험에서 'TextRank'로 표현한다. 두 번째 베이스라인은 'TextRank+Dic'으로, TextRank 알고리즘과 회의록의 단어 사전 가중치를 결합하여 문장의 중요도를 계산하여 회의록을 요약한다. 베이스라인 및 제안한 방법에서의 TextRank 알고리즘 파라미터는 모두 동일하게 설정하였다. 문장에서 명사 상당어구들을 추출하기 위해 형태소 분석기로는 HAM (Hangul Analysis Module)[13]을 사용하였다. 트리

생성과정에 관한 평가는 비교 대상 없이 제안한 방법의 성능만을 측정하였다.

모든 실험에서 요약 비율을 10%에서 50%까지 10% 구간별로 5개의 요약문을 생성하고, 그 결과를 분석하였다. 50% 이상 요약을 할 경우, 요약되지 않는 문장들 간의 공기한 단어가 존재하지 않아 TextRank 알고리즘이 적절히 수행되지 문제가 존재하여 최대 50%로 요약 비율을 설정하였다. 그림 5는 정답 요약문과 같은 단어를 포함하고 있는지를 ROUGE-1으로 평가한 결과이다.

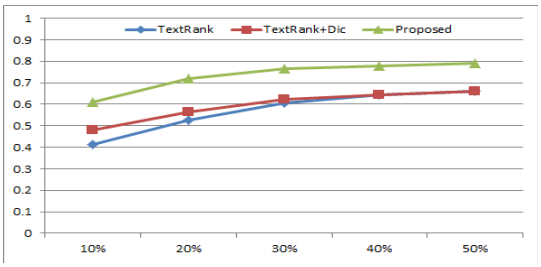

그림 5. ROUGE-1으로 평가한 결과
Fig. 5. ROUGE-1 values of three methods

실험 결과, 먼저 'TextRank+Dic'가 'TextRank'에 비해 성능이 10%-30% 구간에서 높음을 볼 수 있다. 이 결과를 통해 회의록의 단어사전이 요약에 의미가 있음을 알 수 있다. 하지만, 40%-50% 구간에서는 비슷한 결과를 내는데, 이는 ROUGE-1이 단어 추출에 대한 평가로, 이 두 방법이 비슷한 주제의 문장으로 최종 요약되어 같은 단어의 파악된다. 다음으로 제안한 요약 방법이 'TextRank+Dic'에 비해 전 구간에 걸쳐 높음을 볼 수 있다. 이는 중요문장 추출과정 중 중요문장 추출단계에서 주제문장과 관련이 있는 문장들이 적절히 추출되었음을 의미한다. 특히 'TextRank'에 비해 0.13에서 최대 0.19정도 높은 성능을 보인다. 이를 통해 제안한 방법이 회의록의 특징을 잘 반영하여 요약한다는 것을 의미한다.

다음으로 정답 요약문과 얼마나 같은 문장을 포함하고 있는지를 평가한다. 그림 6은 F_1-measure으로 문장 추출에 대한 성능을 평가한 결과이다.

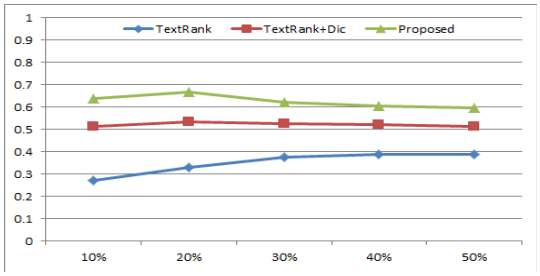

그림 6. F_1-measure으로 평가한 결과
Fig. 6. F_1-measure values of three methods

그림 6에서 보듯이, ROUGE-1으로 평가한 결과와 유사하게 제안한 방법이 베이스라인 방법에 비하여 모든 요약 비율에서 높은 성능을 보인다. 특히, 제안한 방법은 요약 비율이 20%인 구간에서 약 0.66으로 가장 높은 성능을 보인다. 이는 정답 요약문의 요약 비율이 20%-25%이므로 해당 구간에서의 성능이 가장 높게 나온 것으로 분석된다. 이 실험에서는 ROUGE-1 평가와는 달리 'TextRank+Dic'이 'TextRank'보다 전 구간에서 높은 성능을 보인다. 이는 회의록의 단어사전이 주제문장 추출에 도움을 주는 것을 의미한다.

중요문장 추출과정에 관한 위의 두 실험 결과, 제안한 요약 방법이 기존의 요약 방법에 비해 회의록 요약에서 중요문장들을 추출함을 알 수 있다. 또한, 제안한 방법에서 회의록의 단어사전을 이용하여 가중치를 부여하는 것이 성능향상에 도움을 줌을 볼 수 있다.

마지막으로 제안한 방법이 얼마나 전체 흐름을 쉽게 파악할 수 있게 요약하는 지를 정답의 트리 구조와 생성된 요약문의 트리 구조 비교를 통해 평가한다. 그림 7은 *accuracy* 식을 이용하여 트리 생성의 정확도를 측정한 결과이다.

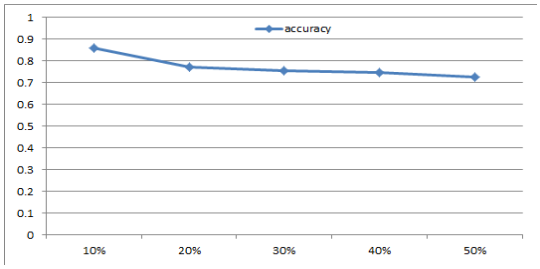

그림 7. 트리 생성의 정확도 측정 결과
Fig. 7. Accuracy of tree-structure

실험 결과, 10% 구간에서 가장 높은 정확도를 보이고 요약 비율이 증가할수록 정확도는 감소하는 것을 볼 수 있다. 이는 트리 생성을 측정하는 평가함수 *accuracy* 가 두 노드의 *root*로부터 모든 경로가 정확히 일치해야만 해당 노드가 제대로 생성된 것으로 평가하기 때문에, 위의 결과가 나타나는 것으로 분석된다. 즉, 요약 비율이 커질수록 정답 요약문에는 포함되지 않는 문장들이 많이 추출되고, 추출된 문장들이 트리 구조에 포함되며, 이는 성능 향상에 노이즈로 작용한다. 비록 요약문의 생성 비율이 높아질수록 성능이 감소하지만 전 구간에서 0.7 이상의 성능을 보이므로 이는 회의록 요약에서 유의미한 값으로 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 회의록의 특징을 반영하여 회의록을 요약하는 방법을 제안하였다. 제안한 방법은 크게 2가지 관점으로, 중요문장을 추출하여 문서의 크기를 줄이는 방법과 문서의 표현을 변형하여 전체 흐름을 알아보기 쉽게 하는 방법으로, 회의록의 요약을 수행하였다.

제안한 회의록 요약 방법은 먼저 문서의 크기를 줄이기 위해 회의록이 회의가 진행됨에 따라서 여러 세부 주제가 나타나는 특징을 반영하여 진행자의 문장에서 세부 주제를 찾는다. 다음으로 찾아진 세부 주제별로 각 주제와 문장들의 유사도를 계산하여 회의록의 중요한 문장들을 추출한다. 최종적으로 회의록의 전체 흐름을 쉽게 알아볼 수 있게 회의록의 문장들이 대화문인 특징을 반영하여 추출된 중요한 문장들 사이의 종속관계를 분석하여 트리형태로 표현하였다.

제안한 방법 성능은 중요문장 추출과정과 트리 생성과정으로 나누어 살펴보았다. 중요문장 추출과정에서 제안한 모

텔과의 성능 평가를 위해 TextRank 알고리즘만을 사용한 방법과 TextRank 알고리즘에 회의록의 단어사전을 사용한 방법과 비교하였다. 실험 결과, 모든 요약비율에서 높은 성능을 보였다. 이를 통해, 제안한 방법이 회의록의 특징을 잘 반영하여 요약함을 볼 수 있다. 트리 생성과정에서는 요약비율 전 구간에서 제안한 방법의 성능이 유의미함을 보였다.

실험을 통해 미리 구축된 회의록 단어사전이 중요한 문장을 선택하는데 영향을 준다는 것을 보였다. 하지만 모든 도메인에 대해 이러한 단어사전을 직접 구축하는 것은 시간적으로나 경제적으로 힘든 일이다. 향후 연구로 이러한 단어사전을 자동으로 구축하는 기법에 관한 연구를 하겠다. 또한, 현재 제안한 방법의 트리 생성기법이 생성하는 요약문의 비율이 높아짐에 따라서 성능이 저하되는데, 이러한 문제점을 보완할 수 있는 새로운 트리 생성기법에 관해서도 연구를 할 예정이다.

참 고 문 헌

[1] 대한민국 국회 회의록 시스템, Available: http://likms.assembly.go.kr/record/index.html, 2002, [Accessed: October 21, 2010]

[2] J. Goldstein, M. Kantrowitz, V.Mittal, and J. Carbonell, “Summarizing Text Documents: Sentence Selection and Evaluation Metrics”, In *Proceedings of ACM-SIGIR'99*, pp.121-128, 1999

[3] M.Sanderson, “Accurate User Directed Summarization from Existing Tools”, In *Proceedings of 7th International Conference on Information and Knowledge Management*, pp. 45-51, 1998

[4] Rada Mihalcea, Paul Tarau, “TextRank: Bringing Order into Texts”, In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 404-411, 2004.

[5] S.Brin and L.Page, “The Anatomy of a Large-scale Hypertextual Web Search Engine”, *Journal of Computer Networks and ISDN Systems*, Vol. 30, No. 1-7, pp. 107-117, 1998.

[6] 홍진표, 차정원, “TextRank 알고리즘을 이용한 한국어 중요 문장 추출”, *2009 한국컴퓨터종합학술대회 논문집*, Vol. 36, No. 1(C), pp. 311-314, 2009

[7] H. P. Luhn, “The Automatic Creation of Literature Abstracts”, *IBM Journal of Research and Development*, Vol. 2, No.2, pp. 159-165, 1958.

[8] H. P. Edmundson, “New Methods in Automatic Extracting”, *Journal of the Association for Computing Machinery*, Vol. 16, No. 2, pp. 264-285, 1969.

[9] J. Kupiec, J. Pedersen, and F. Chen, “A Trainable Document Summarizer”, In *Proceedings of 18th ACM-SIGIR Conference*, pp. 68-73, 1995.

[10] Document Understanding Conferences, Available: http://duc.nist.gov/index.html, 2000, [Accessed: September 15, 2010]

[11] Chin-Yew Lin, “ROUGE: A Package for Automatic Evaluation of Summaries”, In *Proceedings of the ACL-04 Workshop: Text Summarization Branches Out*, pp. 74-81, 2004.

[12] 송원문, 김영진, 김은주, 김명원, “동적 연결 그래프를 이용한 자동 문서 요약 시스템”, *정보과학회 논문지: 소프트웨어 및 응용*, Vol. 36, No. 1(C), pp. 311-314, 2009

[13] 강승식, *한국어 형태소 분석과 정보 검색*, 홍릉과학출판사, 2002

[14] Inderjeet Mani, *Automatic Summarization*, John Benjamins Publishing Company, 2001.

저 자 소 개

이재걸 (Jae-Kul Lee)

2006년 : 안동대학교 컴퓨터공학과 졸업
2006년～현재 : 경북대학교 컴퓨터공학과 석사과정

관심분야 : 자연언어처리, 기계학습
Phone : 053-940-8692
E-mail : jklee@sejong.knu.ac.kr

박성배 (Seong-Bae Park)

1994년 : 한국과학기술원 컴퓨터과학과 졸업 (학사)
1996년 : 서울대학교 대학원 컴퓨터공학과 졸업(석사)
2002년 : 서울대학교 대학원 컴퓨터공학과 졸업(박사)
2004년～현재 : 경북대학교 IT대학 컴퓨터학부 교수

관심분야 : 기계학습, 자연어처리, 텍스트마이닝, 정보추출, 생명정보학
E-mail : sbpark@sejong.knu.ac.kr

이상조 (Sang-Jo Lee)

1974년 : 경북대학교 수학교육과 줄업
1976년 : 한국과학기술원 전산학과 졸업 (석사)
1994년 : 서울대학교 컴퓨터공학과 졸업 (박사)
1976년～현재 : 경북대학교 IT대학 컴퓨터학부 교수

관심분야 : 자연언어처리, 기계번역, 정보검색, 정보추출
E-mail : sjlee@knu.ac.kr