유사 적합성 피드백 기반의 문서 요약 기법을 이용한 효과적인 스니펫 생성

An Effective Snippet Generation Method using Text Summarization Techniques based on Pseudo Relevance Feedback

안홍국, Hongguk An*, 고영중, Youngjoong Ko**, 서정연, Jungyun Seo*
*서강대학교 컴퓨터학과, **동아대학교 컴퓨터학과

요약 정보 검색의 결과로 나타나는 요약문을 스니펫(snippet)이라 한다. 사용자는 자신이 원하는 정보를 얻기 위해 문서를 검색하는데, 이 때 스니펫은 사용자가 원하는 문서를 찾는데 중요한 역할을 한다. 본 논문에서는 정보검색 분야에서 높은 성능을 보이는 유사 적합성 피드백을 자동 문서 요약에 맞게 적용하여 높은 성능의 스니펫 생성 시스템을 구현한다. 우선, 사용자의 질의가 포함된 문장들을 일차적으로 요약 문장 후보로 추출한다. 그리고 추출된 문장 후보로부터 명사들을 질의 후보로 고려한다. 각 문장이 질의의 포함 여부에 따라 문장의 적합성을 판단하게 되고, 유사 적합성 피드백 확률 모델에 적용한 후 질의 후보들의 가중치를 추정하여 가중치 순위를 통해 확장할 질의들을 결정한다. 확장된 질의들과 기존의 질의들의 가중치를 합산하여 각 문장의 순위를 매기게 되고 가장 높은 순위의 문장들이 스니펫으로 제시된다. 논문에서 제안한 기법은 추가적인 핵심 질의들을 자동으로 확장하여 중요한 문장을 추출할 수 있다. 이 연구를 위해서 일반 상용 정보 검색 서비스에서 제공하는 스니펫을 수집하였고 이들의 정확도와 시스템의 정확도를 비교하였다. 실험 결과를 통해 살펴본 제안된 시스템의 성능은 상용 정보 검색기에서 제공되고 잇는 스니펫의 정확도보다 우수한 성능을 보였다.

핵심어: 스니펫(Snippet), 문서 요약(Text Summarization), 유사 적합성 피드백(Pseudo Relevance Feedback)

1. 서론

인터넷의 발달로 인하여 많은 사람들이 인터넷 문서를 통해서 정보를 얻고 있다. 사용자는 원하는 정보를 얻기 위해 그 정보와 관련 있는 질의를 정보 검색기에 입력하여 문서를 검색한다. 정보 검색기는 사용자가 입력한 질의에 따른 검색 결과를 제목과 문서의 요약문을 보여준다. 이렇게 제시되는 요약문을 스니펫(snippet)이라 한다[1]. 검색 결과 문서들의 수가 무수히 많은데다가 사용자는 문서의 제목과 스니펫(snippet)만을 통해서 문서를 선택하기 때문에 스니펫의 성능은 정보 검색에 있어 중요한 부분을 차지한다. 만약 스니펫(snippet)이 문서의 내용을 충분히 표현할 수 있는 문장으로 이루어진다면 사용자는 보다 빠르고 정확하게 판단하여 본인의 요구에 부합하는 문서를 찾을 수 있을 것이다.

스니펫 생성은 문서집합 중 적합문서를 선택하는 정보검색처럼, 한 문서 안의 문장집합 중 적합문장을 선택하는 문제로 볼 수 있다. 이렇게 문제의 정의를 바꾸어 생각하면 정보검색 기법을 스니펫 생성에 활용할 수 있다. 우리는 정보검색 분야에서 적합성 피드백 방법이 높은 성능을 나타내는데 주목하였다. 적합성 피드백 방법은 초기 검색 결과의 적합성 여부에 대해 피드백을 받아 질의를 수정하고 이를 재검색하는 과정을 통하여 검색 성능을 향상시키는 방법이다. 만약 사용자의 피드백이 없다면 초기 질의를 확장할 수 없기 때문에 제시된 방법이 유사 적합성 피드백 기법이다. 유사 적합성 피드백은 사용자의 피드백 없이 사용자 질의에 대한 초기 검색 결과로부터 적합성 정보를 얻어 초기 질의를 변경한 후 재 검색하여 검색 성능을 향상시키는 방법이다[2]. 본 논문은 초기 질의어로 문장의 적합성을 판단하고 이 적합성 정보를 토대로 질의를 확장한 후 이 확장 질의를 반영하여 중요 문장을 추출하는 스니펫 생성 시스템을 제안한다.

1장은 논문에 대해 간략한 소개를 하고 2장은 관련된 연구에 대해 알아본다. 3장은 논문에서 제안하는 시스템의 구성과 적용한 방법론에 대해 설명하였다. 4장에서는 논문에서 행한 실험과 그 결과를 소개하고 5장에서는 결론과 향후 과제에 대해 논한다.

2. 관련연구

자동 문서 요약에 대한 기존 연구들을 생산되어 나오는

요약문의 초점과 영역에 관하여 나누는 것이 일반적인 방법이다[3]. 이 장에서는 일반적인 자동 요약 방법론과 적합성 피드백을 이용한 질의 분해에 관한 연구에 대해서 알아본다.

2.1 일반적인 문서 요약

일반적인 문서 요약은 문서의 분석 수준에 따라 표층 수준(surface-level), 단위 수준(entity-level), 담화 수준(discourse-level)의 방법론으로 나눌 수 있다[4].

2.1.1 표층 수준 접근 방법론

전통적인 방법론으로써, 단어의 빈도, 문장의 위치, 단서어와 같이 대상 문서에서 형태적으로 드러나는 통계 정보를 이용하여 문서 요약을 시도한다. 요약 속도가 빠르고 시스템 구현이 쉬우나, 다의어, 유사어와 같은 의미 구분을 하지 못함으로써, 문서를 지나치게 단순한 통계 테이블로 간주한다. Luhn은 문서의 주제를 표현하는 단어는 자주 사용된다는 직관에 의거하여, 가장 많이 사용된 단어를 문서의 주제어라고 보았다[5]. 요약은 이러한 주요 단어가 포함된 문장을 추출함으로써 생성된다. Edmunson은 첫 번째 사용된 문장(또는 마지막에 사용된 문장)과 같이 위치에 따라 문장의 중요도가 다르다는 점을 연구하였으며, 아울러 'significant', 'hardly'와 같은 단서에도 중요 문장을 파악하는데 중요한 역할을 할 수 있음을 보였다[6]. 이 외에 이런 통계적 특징을 학습을 통해 습득함으로써 성능 향상을 꾀한 연구도 있었다[7]. 이 방법들은 모두 자동 문서 요약 연구의 선구적인 역할을 하였으나, 앞서 지적한 바와 같이 지나치게 단순한 통계에 의존함으로써, 문서의 주제를 파악하는데 한계를 보였다.

2.1.2 단위 수준 접근 방법론

단위 수준 접근 방법론은 문서의 내부 단위(entity) 간의 관계를 이용한 방법론이다. 주로 단어의 중첩, 동시 발생(co-occurrence), 참조와 같은 특징을 이용한다. Aone 등은 표층 수준의 접근이 가지고 있는 한계를 극복하고자 워드넷(WordNet)을 이용하여 유사어 또는 상하위어를 하나의 개념으로 묶어서 빈도를 계산하였다[8]. 즉 자주 사용된 단어가 주제어라기 보다는 자주 사용된 개념이 주제어라는 보다 효과적인 접근 방법이다. Bazilay와 Elhadad는 단어 간의 의미적 거리를 워드넷을 이용하여 계산하고, 이를 이용하여 문서 내 사용된 단어들의 어휘 체인(lexical chain)을 자동 생성한 다음, 강한 어휘 체인이 있는 문장을 추출함으로써 요약을 생성하였다[9]. 이 방법 역시 문서를 형태적으로만 파악하지 않고, 단어간의 관계를 중시한 방법론이다. 이러한 방법론은 표층 수준의 분석에 비해 보다 의미론적인 접근을 시도한 것으로써 질 높은 요약을 생성한다. 대부분의 단위 수준 접근 방법론은 단어간의 의미 거리(semantic distance)를 파악하기 위해서 워드넷이 필요한데, 한국어의 경우 단위(entity)들 간의 거리를 파악하기 위해 워드넷과 같은 언어 자원을 사용하기가 어려운 문제가 있으며, 영어의 경우에도 워드넷을 유지 보수하기 위해서는 많은 시간과 비용이 소요된다. 이러한 문제를 해결하기 위해서 김건오는 워드넷(WordNet)을 대신하여 유사 어휘들을 클러스터링 하는 어휘 클러스터링(lexical clustering) 방법을 제안함으로써 고비용의 의미 체계가 없이도 단어의 의미 관계를 사용할 수 있게 하였다[10]. 이러한 언어자원(knowledge-base)을 사용하는 방법은 고품질의 요약문을 생성할 수 있지만, 속도나 확장 가능성 면에서 아직 많은 개선이 필요하다.

2.1.3 담화 수준 접근 방법론

담화 수준 접근 방법론은 각 문장의 의미와 문장간의 관계 분석 등을 통한 문맥 구조의 파악을 바탕으로 이루어진다. 담화 이론에 따르면 문서는 크게 중심부분(nucleus)과 주변부분(satellite)으로 구성된다는 가정 하에 두 부분 사이의 수사관계(rhetorical relation)를 이용하여 요약을 생성하는 방법이다[11]. 이 방법에서는 문서가 주어지면 일단 수사 구조 분석 알고리즘에 의해 해당 문서의 담화 트리(discourse tree)를 생성한다. 담화 트리의 단말 노드는 구, 절, 문장 등이 되며, 내부 노드는 해당 자식 노드 사이의 수사 관계를 표현하게 된다. 요약을 생성하기 위해서는 담화 트리의 각 노드에 대해 상위 노드가 하위 노드보다 높은 값을 갖도록 중요도 값을 부여하여 그 값이 높은 순으로 정렬한다. 정렬 결과 순위가 높은 구, 절, 문장들을 요약으로 제시한다. 그러므로, 이러한 방법들은 성능 높은 담화 분석기(discourse parser)가 필요하다. 이 방법론이 전체 문서에 대한 수사관계를 필요로 하지는 않는다고 알려져 있다[12]. 주어진 문서를 요약하기 위해서는 모든 문장의 수사관계를 트리로 구성하는데 반해 실제 요약은 전체 문서의 모든 수사관계를 분석해야 하므로 시간이 많이 걸리고, 수사 관계가 별로 없는 문서의 경우에는 성능에 한계를 보인다. 아울러 수사관계의 모호성(ambiguity) 등 해결해야 할 과제들이 많이 남아 있다.

2.2 질의확장을 이용한 요약

통계기반의 접근방법이면서도 모델이 난잡해지는 문제를 해소할 수 있는 방법으로 질의확장 기법을 이용한 연구가 있다[13][14]. 이 방법은 문서요약을 적합한 문장의 선택 작업으로 간주하여, 정보검색에서 사용하는 질의확장 기법을 문서요약에 적용한 것이다. Sanderson은 INQUERY 검색 시스템의 지역적 문맥분석(local context analysis)을 이용하여 주어진 사용자 질의에 대하여 사용자 주도 요약을 생성하였으나, 질의확장을 사용하지 않은 요약 방법에 비해 성능 향상을 보이지 못했다[13]. 이에 대해 Goldstein은 새로운 질의를 만들 때 유사도가 높은 최상위 문장 하나(의사 적합성 피드백), 제목, 문서의 첫 문장 등을 첨가하여 좀 더 다양하게 질의확장을 적용함으로써, 질의확장이 성능 향상에 기여함을 보였다[14]. 한경수는 한국어 신문기사를 말뭉치 대상으로 하여 질의 확장 과정에서 질의가 편향되어 요약이 잘못 생성되는 문제를 완화시키는 방법을 제안하여 실험하였다[15].

2.3 적합성 피드백

이 장에서는 일반적인 정보검색 분야에서 사용되는 적합성 피드백에 대하여 간단히 설명한다.

2.3.1 적합성 피드백

정보검색에서 가장 중요하면서도 어려운 문제 중의 하나는 사용자가 원하는 정보를 찾기 위한 효율적인 질의를 작성하는 일이다. 하지만, 전체 문서집합의 구성에 대해 미리 알고 있지 않는 한 이상적인 최적의 질의는 작성할 수 없다. 대신 최초에는 시험적 질의(tentative query)로 검색을 수행한 후, 이전의 검색 결과에 대한 평가에 기반하여 다음 번 검색의 질의를 개선시키는 방법이 적합성 피드백(relevance feedback)이다[16].

적합성 피드백은 특정 질의와 적합한 문서들은 유사한 벡터로 표현된다고 가정한다. 따라서, 어떤 문서가 주어진 질의에 적합하다고 판단되면 질의를 적합한 문서와의 유사도가 증가하도록 변환하여 질의를 개선시킨다. 이렇게 개선된 질의는 최초 적합하다고 판단된 문서와 유사한 문서들을 추가적으로 검색하여 더 많은 량의 적합 문서를 검색해 낼 수 있다.

실제 이 적합성 피드백을 이용할 때에는 전체 문서집합에 대해 적합문서와 부적합문서를 미리 알 수 없으므로, 이미 적합성이 알려져 있는 문서들의 정보에 기반하여 질의확장을 수행한다. 이때의 적합성을 사용자가 알려주는 방법을 사용자 적합성 피드백(user relevance feedback)이라 하고, 사용자의 개입 없이 초기질의로 검색된 결과 문서 중 상위 문서를 적합한 문서로 간주하여 적합성 피드백을 적용하는 방법을 유사 적합성 피드백(pseudo relevance feedback)이라 한다.

적합성 피드백을 통해 질의를 확장해 가는 과정은 다음과 같이 식으로 표현할 수 있다[17].

$$Q^{new} = \alpha Q^{old} + \frac{\beta}{|R|}\sum_{Di \in R} D_i - \frac{\gamma}{|N|}\sum_{Di \in N} D_i \qquad (1)$$

여기서 Q^{new} 는 새로 확장된 피드백 질의 벡터를 Q^{old} 는 확장되기 전 단계의 질의 벡터를 의미한다. R과 N은 각각 초기 검색된 문서집합 중에서 적합하다고 판단된 문서집합과 부적합하다고 판단된 문서집합을, |R|과 |N|은 각각 해당 문서집합의 문서 개수를 뜻한다. α, β, γ는 이전 단계의 질의, 적합문서집합, 부적합문서집합 간의 중요도를 조율하는 상수이다. 식 (1)은 적합문서나 부적합문서의 정보를 각 문서집합의 크기로 정규화하여 질의확장에 적용하는 방법이다.

2.3.2 확률 모델(The Probabilistic Model)

확률 모델은 적합 문서와 부적합 문서에 나타나는 질의어의 분포에 기초해 Robertson & Sparck Jones에 의해 제안된 모델이다[18].

$$w = \log \frac{p(1-q)}{q(1-p)} \qquad (2)$$

Sparck Jones는 이러한 상대적 가중치 식을 이용하여 적합성 피드백을 수행에 적용하였다[19]. 이 실험에는 사용자들로부터 적합 문서로 피드백 받은 문서를 이용하여 초기 질의어에 대한 가중치를 다시 부여를 했는데, 일반적 IDF값을 이용할 경우보다 높은 성능을 보여, 적합성 피드백을 위해서는 확률적 가중치 방법이 유용함을 보여 주었다.

Croft & Harper는 사용자 피드백을 받는 대신 초기 검색에 의해 검색된 문서를 이용하여 초기 질의어의 가중치를 다시 부여하는 방법을 제시했다[20].

실제로 확률적 가중치 방법 자체는 질의를 확장할 수 있는 스킴을 제공하지 않는 것으로 알려져 있다. 그러나 많은 연구자들은 질의 확장에 확률적 가중치 방법을 사용하려고 여러 가지 시도를 했다.

Harper & van Rijsbergend는 초기 검색의 적합 문서를 이용해 질의어 가중치를 재조정하고, MST(Maximum Spanning Tree)를 이용하여 질의어에 직접 연결된 모든 키워드를 초기 질의에 확장해 주는 방법을 제시하였다[21].

Wu & Salton은 Cranfield collection을 사용한 실험에서, 초기 검색에 의한 적합 문서를 이용한 질의어 재가중치 부여와 함께 적합 문서의 모든 키워드를 확장어로 사용하여 질의를 확장했을 때 32.7%의 정확도 향상이 있음을 보였다[22].

3. 유사적합성 피드백 기반의 문서요약

정보검색이 문서집합에서 사용자가 원하는 몇 개의 적합한 문서를 찾아내는 것이라면, 문서요약은 한 문서, 즉 문장집합에서 그 문서의 내용을 대표하는 몇 개의 문장을 찾아내는 작업으로 생각할 수 있다[14].

정보검색과 연관지어 문서요약의 문제 정의를 바꾸면 정보검색에 이용되는 여러 가지 기법들을 문서요약에 적용할 수 있다. 본 논문에서는 정보검색에 이용되는 유사 적합성 피드백에 기반한 질의확장 기법을 문서요약에 적용한다. 제안하는 시스템은 검색 엔진의 결과로 나타나는 문서들을 중요 문장들로 추출하여 요약문을 생성한다. 여기서 고려할 사항은 검색 엔진의 결과로 요약문이 나타나기 때문에 사용자는 실시간으로 검색 결과를 볼 수 있어야 한다. 만약 많은 시간이 소요되거나 많은 자원을 사용하게 될 경우 사용자들이 불편을 느낄 수 있고, 시스템에 부하가 생겨서 심각한 문제를 발생시킬 수 있다.

이러한 문제를 방지하기 위하여, 요약문을 생성하는데 소요되는데 최소한의 자원으로 최고의 효율을 얻기 위한 방법을 제시한다. WordNet과 같은 언어 자원을 활용하거나 기계학습 방법을 적용하는 것은 제한한다. 따라서 이 시스템은 웹 문서의 요약문이라는 특수한 상황에 맞게 요약문서를 생성하는데 발생하는 자원을 최소한으로 사용하는 통계식 방법을 채택하였다.

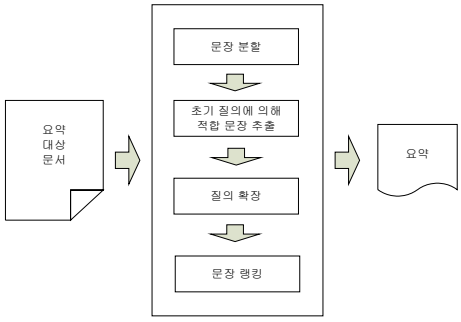

그림 1 제안 시스템의 전체 구성도

시스템에 요약 대상 문서가 입력이 되면 이 문서를 문장 단위로 분할하게 된다. 분할 된 문장은 품사 태거를 사용하여 명사를 추출한 후 명사들의 열로 구성한다. 그리고 나서 초기 질의어 즉 사용자가 처음 입력한 질의어를 포함하는 적합 문장을 추출한다. 모든 적합 문장들의 명사들은 질의 확장을 위한 후보가 되고 유사 적합성 피드백 확률 모델을 통해 가중치가 높은 후보들을 질의어로 확장하게 된다. 각 문장은 포함하고 있는 질의어의 가중치 합과 그 문장의 위치 정보를 합산하여 점수를 얻고 이 점수가 가장 높은 문장들을 요약문으로 제시한다. [그림 1]은 제안 시스템의 전체 구성도를 보여 주고 있다.

3.1 적합성 피드백 기법을 적용하기 위한 전처리

유사 적합성 피드백 기법을 이용한 문서요약은 통계적인 방법을 사용한다. 이러한 통계적인 방법을 사용하기 위하여 각 문장에 포함된 단어들을 추출하여야 하는데 이 때 품사 태거(POS tagger)를 사용하여 추출한다.

품사 태거는 문장 단위만을 분석할 수 있으므로 문서를 문장들로 나누는 과정이 선행되어야 한다. 인터넷 문서는 일반적인 문서와는 달리 다양한 형태로 구성이 되어 있어, 문장을 분할하는 작업은 쉽지 않다[23]. 현재 다양한 기술들이 제시되어 있지만, 실시간으로 결과를 보여줘야 하는 스니펫 생성의 제약 조건을 만족시키기 위하여 본 시스템에서는 간단한 휴리스틱(heuristic)을 통해서 문장을 분할한다. 시스템에 입력된 문서들은 제목과 본문으로 이루어져 있는데 본문을 온점(.), 물음표(?), 느낌표(!)와 같은 마침표에 의해 구분한다. 이 때 온점은 소수점과 웹사이트 주소를 나타내는 점과 구분하기 위하여 문장의 온점 뒤에 여백이 있는지 없는지를 확인하여 구분한다.

이렇게 분할된 문장들은 품사 태거를 거치면서 단어와 품사의 쌍으로 구성이 된다. 이 때 분류한 품사들을 모두 사용하게 되면 시스템의 속도가 저하될 뿐 아니라 문장의 점수 계산에 있어서 성능에 악영향을 미친다. 그러므로 필요한 종류의 품사를 사용하는 것이 시스템의 속도 저하를 막을 뿐만 아니라 시스템의 정확도를 높이는데 크게 기여할 수 있다는 점을 고려하여 각 품사들을 달리하며 실험을 수행하였다.

우선 일차적으로 명사들만을 대상으로 실험을 하였고, 그 외의 품사들을 추가로 확장하여 실험을 한 결과 명사들만을 대상으로 한 것이 제일 좋은 성능을 보였다.

따라서 이 시스템에서 사용하는 품사들은 다음의 표 1과 같다.

표 1 시스템에서 사용하는 품사와 그 예

품사	예
일반명사	홈런
고유명사	이승엽
명사추정범주	요미우리

3.2 유사적합성 피드백 기법 적용

이제 추출한 단어를 이용하여 유사 적합성 피드백 기법에 적용한다. 유사 적합성 피드백 기법은 여러 가지가 있지만 가장 기본적인 확률 모델인 Robertson/Sparck Jones에 의해 제안된 식 (3)을 사용하였다. 우리는 정보 검색 분야에서 사용되는 이 모델을 문장 추출에 적합한 방법으로 적용하였다.

$$w = \log \frac{p(1-q)}{q(1-p)} = \log \frac{(r+0.5)(S-s+0.5)}{(R-r+0.5)(s+0.5)} \quad (3)$$

p: 질의어가 적합 문장에 나타날 확률

q: 질의어가 부적합 문장에 나타날 확률

R: 초기 질의어 Q를 포함하는 문장 수

r: R 중에서 키워드 t를 포함하고 있는 문장 수

S: 초기 질의어 Q를 포함하지 않은 문장 수

s: S 중에서 키워드 t를 포함하고 있는 문장 수

질의 중심의 자동 요약에서 정보 검색 후 생성된 결과 문서들은 일차적으로 사용자가 입력한 질의어가 관심의 대상이다. 이러한 관점으로 보았을 때 질의어를 포함하는 문장들은 일차적으로 사용자에게 관심의 대상이 되는 문장이다. 또한 질의어를 포함한 문장들은 그렇지 않은 문장에 비해 적합성이 높다고 볼 수 있다.

이제 이러한 가정을 토대로 질의어 확장을 수행한다. 문장 분할 단계에서 질의어를 포함한 문장과 포함하지 않은 문장으로 나누었는데, 우선적으로 질의어를 포함한 문장이 후보 문장 대상이 된다. 질의어를 포함한 문장에서 명사들은 확장할 질의어의 후보가 된다. 이 질의어의 후보들은 그들이 적합 문장에서 발견되는 빈도와 적합하지 않은 문장에서의 발견되는 빈도를 조사하여 식(3)의 값을 구한다. 그런 후 이들을 내림차순으로 정렬하여 가장 높은 순으로 질의어들을 추출한다.

$$RFscore(S_i) = \sum_{w_i \in S_i} w_j \quad (4)$$

추출된 질의어들은 각각 고유의 값을 가지게 되고 식 (4)

에 입력이 되어 각 문장의 적합성 점수를 계산하게 된다. 식(4)에서 S_i는 i번째 문장이고, w_j는 i번째 문장이 포함하고 있는 확장된 질의어의 적합성 가중치를 의미하는데 이 가중치는 앞선 단계의 식(3)을 통해 구한 값을 의미한다. 즉 i번째 문장의 점수는 확장된 질의어 중 이 문장이 포함하고 있는 질의어의 가중치 합으로 구성된다. 이 가중치는 문서에 대한 적합성 정보를 담고 있기 때문에 문장에 포함되어 있는 질의어의 가중치 합은 곧 그 문장의 문서에 대한 적합성의 수준을 나타낸다고 할 수 있다.

이를 위해 우리는 [표 2]와 같이 질의어의 개수를 증가하면서 실험을 하였다. 사용된 데이터는 5장에서 상세히 기술되어 있다.

표 2 질의어의 개수에 따른 성능 분포

질의어의 개수	정확도
질의어 (1)	46.1%
질의어 (2)	46.1%
질의어 (3)	49.5%
질의어 (4)	52.5%
질의어 (5)	52.6%
질의어 (6)	53.9%
질의어 (7)	52.6%
질의어 (8)	52.1%

[표 2]에서 질의어(1)은 질의어 확장을 사용하지 않고 입력된 질의어만을 사용한 경우이며, 질의어(2)는 확장된 질의어를 하나 사용한 경우이다. 이 실험을 통해 확인한 바로는 가장 높은 성능을 보일 때 질의어의 개수는 6이다. 우리는 이 점을 이용하여 질의어의 개수를 6개로 고정하여서 실험하였다.

3.3 위치정보의 활용

문서의 상위 문장들이 기사 요약에서 중요하고 좋은 요약이라고 알려져 있다[14]. 이러한 가정을 확인하기 위하여 실험 문서들의 정답 문장 분포를 살펴보았다.

표 3 정답 문장의 분포

1	2	3	4	5	6	7	8 이상	총 문서
431	320	144	64	42	33	25	67	1126

[표 3]은 정답 문장 분포를 나타내고 있는데, 특히 8이상은 여덟번째 문장 그 이상에서 정답이 발견된 모든 문서 수의 합을 나타낸다. 표에서 확인할 수 있듯이 많은 문서의 정답들이 첫번째 문장과 두번째 문장에 나타나고 있음을 확인할 수 있다. 그리고 문장의 위치가 뒤쪽일 수록 정답 문장의 수가 줄어드는 것을 알 수 있었다. 이 점을 이용해 우리는 위치 정보를 제안하는 시스템에 적용하였다.

$$PosScore\ (S_i) = 1 - \frac{i-1}{N} \quad (5)$$

여기서 S_i는 i번째 문장이고 N은 문서에서 총 문장의 개수이다.

$$score(Si) = \alpha\left(\frac{RFscore(Si)}{RFscoreMAX}\right) + (1-\alpha)\left(1 - \frac{i-1}{N}\right) \quad (6)$$

그리고 식(6)은 식(4)의 점수와 식(5)의 점수를 합산한 후보 문장의 총점을 나타낸다. 두 식의 점수를 합산하기 위하여 식 (4)에 대한 점수를 정규화 한다. i번째 문장의 점수를 해당 문서의 모든 후보 문장의 점수 중 가장 높은 점수로 나누고 가중치 α 를 통해서 유사적합성 피드백 기법의 점수와 위치 정보 점수의 비중을 달리한다. 그리고 실험을 통해서 최적화 된 성능을 보이는 상수 α 를 결정하였다. 이 때의 상수 α 는 0.4이다.

4. 실험 및 평가

본 논문에서 제안하는 시스템을 평가하기 위한 실험 데이터와 평가 기준, 비교시스템을 설명하고 실험 결과를 보인다.

4.1 실험 데이터

현재 한국어 스니펫에 관한 실험 말뭉치들은 제공되어 있지 않다. 따라서 이 실험을 수행하기 위해 직접 정보 검색기를 이용하여 실험 말뭉치를 구성하였다. 다양한 분야의 실험 말뭉치를 모으기 위하여 뉴스 기사의 영역을 과학기술, 스포츠연예, 정치경제, 사회문화의 네 가지 분야로 나누었고, 이들 영역별로 최근 화제가 되고 있는 단어 20개를 질의어로 하여 총 80개를 실험에서 사용할 질의어로 구성하였다.

표 4 각 분야와 질의어의 예

분야	문서 수	질의어의 예
과학기술	258	RFID, 로봇, 무궁화 5 호, …
스포츠연예	354	설기현, 이승엽, 주몽, …
정치경제	218	북핵, 고구려, FTA, …
사회문화	296	된장녀, 레바논, 서울대, …

각 질의어는 정보검색기의 입력으로 하여 검색을 수행하며 인터넷 기사 10개씩 추출하게 된다. 이 기사들은 제목, 본문, 그리고 정보검색기에서 제공되는 스니펫으로 말뭉치를 구성하였다. 이 때 사용된 정보검색기는 상용검색기인 Naver와 Google의 뉴스검색으로 각각 800개씩 총 1600개

의 문서를 말뭉치로 수집하였다. 이렇게 수집된 문서 중 중복된 문서와 질의를 포함하지 않은 문서 그리고 질의를 포함하는 문장의 수가 두 개 이하인 경우를 제외하여 총 1126개의 문서를 실험 말뭉치로 사용한다.

현재 한국어 스니펫에 관한 실험 말뭉치가 없기 때문에 이에 관한 정확도를 측정하기 어렵다. 우리는 실험의 정확도를 측정하기 위해 각 기사의 질의어를 포함하고 있는 문장을 정답 후보로 간주하고, 각 후보에 번호를 매겨서 세 명의 요약정답 태깅 작업 참여자에게 나누어 준다. 태깅 작업 참여자는 이 정답 후보들에서 가장 문서의 내용을 잘 대표할 수 있는 문장을 하나 정한다. 그런 뒤 모든 문서의 정답을 취합한 후 3명 모두 같은 정답을 지시한 경우와 두 명이 같은 정답을 지시한 경우처럼 다수의 참여자가 지시한 답을 정답으로 간주하였다. 그리고 참여자 3명 모두 다른 답을 제시한 문서들을 따로 모아서 참여자에게 다시 나누어 주어 토의를 통해 정답을 결정하였다.

4.2 성능 평가 방법

제안하는 시스템이 추출한 요약 문장들 중 실험 정답 자료에서 제시하는 문장을 포함할 경우를 맞는 것으로 보고 이들의 정확도(accuracy)를 측정하였다. 시스템과 비교 시스템의 경우, 정확도 측정은 일반적으로 완전한 문장을 생성해낸다고 가정하기 때문에 어려운 일이 아니나 정보 검색기가 제공하는 스니펫의 정확도 측정은 쉽지 않다. 정보 검색기가 제공하는 스니펫은 화면 구성상 일정한 크기로 제한되어 있기 때문에 [그림 2]와 같이 완전한 문장일 경우 보다 불완전한 문장일 경우가 많다. 또한 불완전한 문장을 어디까지 인정할 것인가에 대한 기준이 모호하기 때문에 질의어가 포함되어 있는 불완전한 문장도 정보 검색기가 제공하는 스니펫 문장이라고 간주하고 정확도를 측정하였다. 그리고 공정한 평가를 위해서 정보 검색기의 스니펫으로 제공하는 문장의 수와 제안 시스템의 스니펫으로 제공하는 문장의 수를 같게 하였다. 그런 후 이들이 제시하는 문장들 중 정답 문장이 포함될 경우를 맞는 것으로 보고 평가하였다.

스카이, 터치 센서 지상파DMB폰 출시 네이버 디지털타임스 IT/과학 | 2006.12.14 (목) 오전 6:11
... LCD 탑재로 더욱 시원하고 선명하게 지상파 DMB 시청을 할 수 있도록 했으며, 4:3 비율의 LCD를 탑재한 일반 휴대폰과 달리, 영화관 스크린과 비슷한 15:9비율의 LCD를 채택, 한 차원 높은 스펙터클한 감동을 전달한다. DMB시청 중 Full 멀티태스킹이 가능하며 지상파...

그림 2 불완전한 문장을 지닌 스니펫의 예

4.3 비교 시스템

제안하는 방법의 성능을 비교하기 위하여 일반적으로 자동 문서 요약 시스템에서 사용하고 있는 제목과 위치 정보를 이용하여 문장을 추출하는 방법과 tf*isf를 이용하여 키워드 추출한 후 문장을 추출하는 방법을 구현하여 성능을 비교하였다. 그리고 현재 사용되고 있는 정보 검색기의 스니펫의 정확도를 측정하여 본 논문에서 제안하는 시스템과 비교하였다.

4.3.1 제목 유사도와 위치 정보를 이용한 문서 요약

이 방법은 통계적인 정보를 이용한 것으로 제목과 문장의 유사도가 높을수록 중요한 문장이며, 또한 상위의 위치에 있는 문장이 중요한 문장이란 점을 적용한 대표적인 자동요약 기법이다[6].

먼저 제목의 유사도를 측정하는 방법이다. 제목과 각 후보 문장은 가중치의 벡터로 표현한다고 가정한다. 제목의 명사들을 벡터로 구성을 하고 이와 일대일 대응하는 문장들의 벡터를 구성한다. 이들을 식 (7)과 같이 제목과 문장의 벡터의 내적으로 유사도를 측정한다.

$$Sim(T,S)=\sum_{i=1} t_i \cdot s_i \qquad (7)$$

이렇게 측정한 유사도와 위치 정보를 합한다. 위치 정보는 앞에서 사용된 식 (5)와 같은 것을 사용한다. 그래서 i번째 문장의 총 점수는 식 (8)과 같다.

$$totalscore(S_i)=b\left(\frac{Sim(Si)}{SimMAX}\right)+(1-b)\left(1-\frac{i-1}{N}\right) \quad i=1,2,...,N \qquad (8)$$

b는 제목 유사도와 위치 정보의 가중치의 비율인 상수이다. i는 문장의 순서를 나타내고 N은 총 후보 문장의 개수이다. 이 방법이 4.4절 실험결과의 비교시스템 1이다.

4.3.2 tf*isf를 이용한 문장 추출

이 방법은 tf*isf (Term Frequency - Inverted Sentence Frequency)를 사용해서 키워드를 추출한 후 이 키워드들이 각 문장에서 나타나는 빈도를 측정하여 문장의 순위를 매기는 방법이다[25].

$$w_i = tf_i \times [\log(N/n)+1] \qquad (9)$$

식 (9)에서 w_i는 i번 째 키워드의 가중치를 나타내고, tf_i는 문장에서 이 키워드의 출현 횟수를 나타내고 $log(N/n)$은 문서 내에서 이 키워드가 나타나는 문장의 역을 나타낸다. 이렇게 하여 찾아낸 키워드들로 각 문장에서 발견되는 빈도를 문장의 점수에 반영하여 문장의 순위를 매긴다. 그리하여 제일 높은 두 개의 문장을 스니펫으로 제시한다. 이 방법이 4.4절 실험결과의 비교시스템 2이다.

4.4 실험 결과

실험은 Naver를 정보 검색기로 검색하여 수집한 기사와 Google을 정보 검색기로 검색하여 수집한 기사에 대하여 실험하였다. 실험의 대상은 앞서서 설명한 비교시스템 1, 2와 각 말뭉치를 수집할 때의 정보 검색기의 스니펫이다. 성능은 참여자들이 제시한 정답을 얼마나 맞추냐에 따른 정확도(accuracy) 값으로 표현하였다. 또한 문서를 처리하는 속도를 비교하였다.

4.4.1 Naver 검색 기사

Naver를 정보 검색기로 검색하여 수집한 기사 560개를 제안한 시스템의 효과를 보여주기 위해 비교 시스템 1,2와 Naver 자체에서 제공하는 스니펫을 비교하였다.

[표 5]는 비교시스템 1, 2와 제안시스템, Naver에서 제공하는 스니펫의 성능 그리고 후보 대상이 되는 문서의 개수를 보인다. 이를 보면 알 수 있듯이 제안 시스템은 비교시스템 1보다 10.5%, 비교시스템 2보다 28% 높은 정확도를 보였다. 그리고 상용 검색기인 Naver의 경우 총 20.4%로 정확도 측면에서 낮은 성능을 보이는 것을 확인 할 수 있었다.

표 5 Naver 정보검색기로 수집한 기사들을 대상으로 한 실험결과

	비교 시스템 1	비교 시스템 2	제안 시스템	Naver	후보대상
total	317 (56.6%)	219 (39.1%)	376 (67.1%)	114 (20.4%)	560

4.4.2 Google 검색 기사

Google을 정보 검색기로 검색하여 수집한 기사 566개를 제안한 시스템의 효과를 보여주기 위해 비교시스템 1, 2와 Google 자체에서 제공하는 스니펫을 비교하였다.

표 6은 비교시스템 1, 2와 제안시스템, Google에서 제공하는 스니펫의 성능 그리고 후보 대상이 되는 문서의 개수를 보인다. 이를 보면 알 수 있듯이 제안시스템은 비교시스템 1보다 11.3%, 비교 시스템 2보다 27.7% 높은 정확도를 보였다. 그리고 상용 검색기인 Google의 경우 총 59.5%로 제안 시스템이 정확도 측면에서 9.2% 높은 성능을 보이는 것을 확인 할 수 있었다.

표 6 Google 정보검색기로 수집한 기사들을 대상으로 한 실험결과

	비교 시스템 1	비교 시스템 2	제안 시스템	Google	후보대상
total	325 (57.4%)	232 (41.0%)	389 (68.7%)	337 (59.5%)	566

4.4.3 속도 측정

스니펫을 생성하는데 소요되는 시간은 시스템의 성능을 측정하는 중요한 지표가 된다. 따라서 각 시스템이 문서를 처리하는 속도를 측정하였다. 일반적으로 상용 정보 검색기에서 제공하는 스니펫은 한 번에 10개에서 20개를 동시에 보여 주므로 10개당 처리되는 속도를 비교해 보았다. [표 7]을 살펴 보면 알 수 있듯이 제안시스템이 비교시스템 보다 속도가 느린 단점을 보인다.

표 7 문서 10개당 처리되는 속도

측정 대상	문서 10 개당 처리되는 속도
비교 시스템 1	0.02 초
비교 시스템 2	0.06 초
제안 시스템	0.08 초

5. 결론 및 향후 연구

본 논문에서는 유사 적합성 피드백을 기반으로 하는 요약 기법을 사용하여 높은 성능의 스니펫 생성 시스템을 구현하였다. 정보 검색 분야에서 사용되는 적합성 피드백을 자동 요약과 유사한 점을 보고 이들을 적용하였다. 사용자의 관심은 일차적으로 입력한 질의에 있다는 가정 하에 이 질의를 통해서 적합한 문장과 적합하지 않은 문장을 나눌 수 있다. 이렇게 나뉜 문장을 유사 적합성 피드백의 확률 모델에 적용하여 질의를 확장하고 이 확장된 질의들을 통해서 문장의 점수를 매겨 순위에 따라 문장을 추출한다. 게다가 추가적으로 위치 정보를 결합하여서 전체적으로 다른 비교 시스템들보다 높은 성능을 내는 것을 확인하였다.

그리고 지금 현재 사용되는 상용 정보 검색기의 스니펫의 성능을 확인해 보았다. 이러한 스니펫의 성능을 확인해 봄으로써 좀 더 향상될 수 있는 방안을 제시할 수 있다. 따라서 제안한 시스템이 높은 정확도를 보임으로 이러한 기술을 활용한다면 좀 더 정확한 스니펫을 생성하는 정보 검색기를 개발할 수 있을 것이다.

하지만 속도 면에서 다른 시스템에 비해 약간 떨어지는 단점이 있는데, 이를 해결하기 위해 다음과 같은 방안을 제시한다.

- 정보 검색 서비스에서 제공하는 검색어 순위를 활용하여 검색 빈도가 높은 단어에 대한 문서들은 배치작업을 수행하여 미리 스니펫을 만들어 놓을 수 있다.
- 그 외의 문서들은 기사의 특성상 앞부분의 문장들이 중요 문장임을 이용하여 첫 문장과 둘째 문장을 스니펫으로 제시한다.

향후 연구로는 다음과 같은 과제가 있다. 실험 영역을 뉴스 기사 뿐만이 아닌 블로그나 기타 웹 문서들로 확대하여 평가해 보는 것이다. 그리하여 다양한 검색 분야에서 제안하는 기법의 성능을 평가해 보고 차이점들을 분석하여 실제 검색 시스템에서 사용할 수 있을지 알아 보도록 한다.

Acknowledgement

이 연구(논문)는 산업자원부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었습니다.

참고문헌

[1] Daniel M.Mcdonald and Hsinchun Chen, "Summary in Context: Searching Versus Browsing", *ACM Transactions on Information System*, Vol. 24, No. 1, pp. 111~141, 2006.

[2] 조봉현, 이창기, 안주희, 이근배, "확률적 정보 검색 모델에서의 유사 적합성 피드백 실험", *한글 및 한국어 정보처리 학술발표 논문집*, Vol. 13, pp. 183~190 ,2001.

[3] Firmin, T. and Chrzanowski, M. J. "An evaluation of automatic text summarization systems.", *Advances in Automatic Text Summarization*, I. Mani and M. T. Maybury, Eds. MIT Press, Cambridge, MA, pp. 325~336, 1999.

[4] K. Sparck Jones, "Automatic Summarizing: Factors and Directions", *Advances in Automatic Summarization*, The MIT Press, 1999.

[5] H. P. Luhn, "The Automatic Creation of Literature Abstracts", *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 159~165, 1958.

[6] H. P. Edmundson, "New Methods in Automatic Extraction", *Journal of the Association for Computing Machinery*, Vol. 16, No. 2, pp. 264~285, 1969.

[7] J. Kupiec, J. Pedersen, and F. Chen, "A Trainable Document Summarizer.", *Proceedings of 18th ACM-SIGIR Conference*, pp. 68~73, 1995.

[8] C. Aone, M. E. Okurowski, J, Gorlinsky, and B. Larsen, "A Scalable Summarization System using Robust NLP", Proceedings of the Workshop on Intelligent Scalable Text Summarization (ACL/EACL'97), pp. 66~73, 1997.

[9] R. Barzilay and M. Elhadad, "Using Lexical Chains for Text Summarization." *Proceedings of the TIPSTER Text Pharse III Workshop*, 1998.

[10] 김건오, *어휘 클러스터링을 이용한 주제어 판별 기반의 자동 문서 요약*, 서강대학교 컴퓨터학과 석사학위논문, 2001.

[11] D. Marcu, "The Rhetorical Parsing of Natural Language Texts", *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (ACL/EACL'97), pp. 96~103, 1997.

[12] I. Mani, Automatic Summarization, John Benjamins Publishing Company, pp. 114~125, 2001.

[13] M. Sanderson, "Accurate User Directed Summarization from Existing Tools", In *Proceedings of the 7th International Conference on Information and Knowledge Management*, pp. 45~51, 1998.

[14] 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.

[15] 한경수, *질의 분해를 이용한 적합성 피드백 기반 자동 문서 요약*, 고려대학교 컴퓨터학과 석사학위논문, 2000.

[16] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley Publishing Company, 1989.

[17] R. Baeza-Yates and B Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Publishing Company, 1999.

[18] S. E. Robertson and K. Sparck Jones, "Relevance Weighting of Search Terms", *Journal of the American Society for Information Science*, Vol. 27, No. 3, pp. 129~146 , 1976.

[19] K. Sparck Jones, "Search Term Relevance Weighting Given Little Relevance", *Journal of Documentation*, Vol. 35, No. 1, pp. 30~48. 1979.

[20] W. B. Croft and D. J. Harper, "Using Probabilistic Models of Document Retrieval Without Relevance Information", *Journal of Documentation*, Vol. 35, No. 4, pp. 285~295, 1979.

[21] D. J. Harper and C. J. Van Rijsbergen, "An Evaluation of Feedback in Document Retrieval Using Co-Occurrence Data", *Journal of Documentation*, Vol. 35, No. 3, pp. 189~216, 1978.

[22] H. Wu and G. Salton, "The Estimation of Term Relevance Weights using Relevance Feedback", Journal of Documentation, Vol. 37, No. 4, pp. 194~214, 1981.

[23] J. H. Shim, D. S. Kim, J. W. Cha, G. B. Lee and J. Y. Seo, "Integrated multi-strategic Web document pre-processing for sentence and word boundary detection", Information processing & management, Vol. 38, No. 4, pp. 409~427, 2002.

[24] J. Larocca Neto, A. D. Santos, A. A. Kaestner and A. A. Freitas, "Generating Text Summaries through the Relative Importance of Topics", *Lecture Notes in Artificial Intelligence*, No. 1952, pp. 300~309, 2000.