

효과적인 브라우징을 위한 뉴스 기사 요약에 관한 연구

†이주호*, 정승도*, 조정원*, 최병욱*

*한양대학교 전자통신전파공학과

A Study on Skimming of News Article for an Efficient Browsing

†Chu-Ho Yi*, Seung-Do Jeong*, Jung-Won Cho*, Byung-Uk Choi*

*Dept. of Electrical and Computer Engineering, Hanyang Univ.

E-mail : dluck@hymail.hyu.ac.kr

요 약

수많은 종류의 비디오 데이터를 효율적으로 검색하기 위해서는 데이터를 분석하여 사용자에게 먼저 전체 비디오의 요약을 제시하는 것이 효과적이다. 본 논문에서는 기사 단위로 분할된 뉴스 기사 전체를 보여주지 않으면서도 기사의 내용을 왜곡없이 요약하여 효과적으로 사용자에게 보여주기 위한 방법을 제안한다.

본 논문에서는 사용자에게 시각적인 요약 정보를 앵커 프레임 추출 및 대표 프레임 추출을 통해 필름 스트립(film strip)의 형태로 제시하고, 기사를 소개하는 앵커의 첫 대사를 폐쇄 자막(closed-caption)을 이용하여 추출하여, 이를 기사의 내용에 대한 요약으로 필름 스트립과 같이 제시하도록 하였다.

앵커 프레임을 추출하기 위해 본 논문에서는 폐쇄 자막에서의 "앵커:" 태그가 존재하는 시간 구간과 동기된 프레임을 선정한다. 또한 대표 프레임은 공개형 자막(open-caption)이 존재하는 프레임과 빈도에 기반한 가중치가 높은 폐쇄 자막에서의 키워드와 동기된 프레임을 선정하도록 하였다.

본 논문의 뉴스 기사 요약 시스템은 시각적인 프레임 제시와 함께 기사의 내용을 바탕으로 하는 기사 요약문을 같이 사용자에게 제공함으로써 기존의 필름 스트립 형태만 제공하던 시스템에 비하여 사용자 중심의 지능

형 요약 서비스가 가능함을 실험을 통해 보인다.

I. 서 론

컴퓨터 성능의 급격한 발전과 인터넷의 대중화는 디지털 멀티미디어 데이터를 많은 분야에서 사용하게 된 요인이 되었다. 특히 동영상의 경우는 압축 기술의 발전과 더불어 데이터가 가지고 있는 정보로서의 가치가 커, 그 이용이 나날이 증가되고 있다.

이러한 멀티미디어 데이터를 효과적으로 검색하여, 사용자가 원하는 정보를 보다 빠르고 정확하게 얻어내는 것은 하나의 필수적인 요건으로 받아들여지고 있다.

그러나, 상표 검색과 같은 특별한 영역에서의 검색 환경에서나 적합한 색상(Color), 형태(Shape), 질감(texture), 움직임(motion)등의 특징 정보를 사용하는 비디오 검색 시스템으로는 사용자의 의미에 기반한 요구를 처리하는데 있어 한계를 나타내고 있는 실정이다. 이에 국외에서는 멀티미디어 데이터가 가지고 있는 영상 정보외에도 음성정보, 폐쇄 자막 등을 이용하여 비디오 내용을 분석하고, 이를 색인 및 요약 등에 이용하고자 하는 연구가 활발히 진행 중에 있다[1][2]. 국내의 경우에는 폐쇄 자막을 이용하여 뉴스 검색 시스템을 구현한 연구와, 각 뉴스 기사에 대해 폐쇄 자막내 기자의 소속 정보를 이용하여 자동으로 영역 별로 분류시켜 사

용자가 분야별 검색이 가능하도록 한 연구 결과가 발표된 바 있다[3].

본 논문에서는 사용자가 최종 검색 결과로 비디오 전체의 내용을 보기 이전 단계에서 전체 내용에 대한 요약 결과를 미리 확인할 수 있도록 하여, 보다 효과적인 브라우징이 가능한 뉴스 기사 요약 시스템을 설계 및 구현한다.

II. 대표 프레임 추출

요약 시스템의 결과로서 사용자에게 영상으로 제시할 대표 프레임은 기사가 시작되는 앵커프레임, 기사의 주요 내용을 공개 자막으로 표시하는 공개 자막 프레임 및 기사 내에서 출현 빈도가 높은 단어와 시간상으로 동기된 프레임을 추출하는 것이 기사가 나타내고 있는 내용을 가장 효과적으로 나타낼 수 있다.

1. 앵커 프레임의 추출

일반적으로 폐쇄 자막은 영상 및 음성 신호와 2~4초 정도의 지연(delay) 시간을 가지고 있다. 따라서 이러한 지연 시간을 보상하기 위해서는 폐쇄 자막을 디코더를 통해 추출하는 과정에서 시작 시점을 기준으로 상대적인 발생 시간을 체크하여 폐쇄 자막과 영상 및 음성 신호와 동기를 맞추고 있다[3]. 그림 1은 폐쇄 자막 디코더를 통하여 추출된 어절단위의 자막과 그 발생시간을 데이터베이스에 저장한 결과를 나타내고 있다.

count	word	time
734	앵커:	[379, 211]
735	다음	[379, 491]
736	소식입니다.	[380, 891]
737	남이	[381, 051]
738	물어	[381, 291]
739	있는	[381, 591]
740	꽃게가	[382, 161]
741	오늘도	[382, 491]
742	부산에서	[382, 861]
743	발견됐습니다.	[383, 931]
744	꽃게에	[384, 091]
745	남을	[384, 361]
746	남은	[384, 661]
747	꽃이	[384, 961]
748	중국의	[385, 301]
749	대리	[385, 761]
750	지역인	[386, 501]
751	것으로	[386, 831]
752	추정됩니다.	[387, 861]
753	한편	[388, 021]
754	지난	[388, 731]
755	5월부터	[389, 131]
756	남을	[389, 701]
757	냉장꽃게가	[390, 341]
758	발견됐다는	[391, 131]
759	대기가	[391, 471]
760	나오고	[391, 691]
761	있습니다.	[392, 671]
762	부산의	[392, 691]
763	배편호	[393, 131]
764	기자입니다.	[395, 901]

그림 1. 폐쇄 자막 및 발생 시간

뉴스 데이터를 기사 단위로 분할할 경우 하나의 기사는 앵커 프레임을 시작으로 기사 및 인터뷰가 이어지는 정형적인 구조를 갖는게 일반적이다.

이러한 뉴스 기사를 요약하기 위해서 가장 먼저 효과적으로 제시해 줄 수 있는 대표 프레임이 그림 2와 같은 앵커 프레임이다. 앵커 프레임은 기사의 시작을 알

림과 동시에 상위 자막 혹은 윈도우 자막(어깨걸이 자막)이 존재함으로써 어떠한 내용을 담고 있는지 파악할 수 있도록 도와주기 때문에 대표 프레임으로써의 가치가 매우 높다.

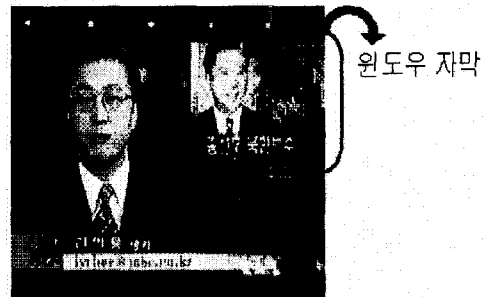


그림 2. 앵커 프레임

기존 연구에서도 앵커 프레임을 추출하기 위한 알고리즘이 소개된 바 있으나 많은 비교 연산 시간을 필요로 하는 단점을 가지고 있다[4]. 본 논문에서는 폐쇄 자막에 포함되어 있는 부가 정보 중 "앵커:" 태그를 이용한다.

앵커 프레임은 대부분 10초 이상 지속되며, 또한 폐쇄 자막이 영상 신호와 동기가 신뢰할 만큼 이루어져 있기 때문에 태그의 발생 시간과 동기화 된 영상만을 뽑아내는 것으로도 정확히 앵커 프레임을 뽑아낼 수 있다. 본 논문에서는 식(1)과 같이 앵커 태그가 발생한 시간에 3초의 딜레이 보상 시간을 더하여 앵커 프레임의 추출 위치를 구하였다.

$$\text{앵커 프레임의 위치} = (AT_{\text{time}} + 3_{\text{sec}}) \times \text{frame rate} \dots(1)$$

이와 같은 방법으로 비교 연산 등의 과정이 불필요하게 되고 단 한 프레임의 디코딩만으로도 정확히 앵커 프레임을 보여 줄 수 있다. 그림 2에서 논문에서 제시한 방법으로 추출한 앵커 프레임을 나타내고 있다.

2. 하단 공개 자막의 추출

뉴스 기사의 내용을 영상 데이터만으로 표현하기는 어렵기 때문에, 뉴스에서는 일반적으로 영상 데이터 내에 그림 3과 같이 하단에 위치하는 공개 자막의 형태로 문자 정보를 포함하고 있으며, 이러한 프레임은 대표 프레임으로 사용될 만큼의 정보량을 가지고 있어 기사의 내용을 효과적으로 전달할 수 있다.

주로 하단에 위치하는 공개 자막의 종류로는, 기사를 대표 하는 문장이나 현장·인물에 대한 내용 그리고 기자의 이름 등이 있다.

본 논문에서는 하단 공개 자막 추출의 복잡도를 줄이기 위하여 MPEG으로 압축된 뉴스 영상에서 하단 공개 자막을 추출할 대상으로 I 프레임만을 선정하였는데,



그림 3. 공개 자막이 포함된 프레임

이는 하단 자막이 약 1~2초간 화면에 나타나기 때문에 I 프레임만으로도 하단 자막이 존재하는 프레임을 뽑아 낼 수 있기 때문이다. 또한, 자막 영역이 가지고 있는 고주파 특성을 이용하고 영상의 밝기 변화에 강인하기 위해 연산에서 프레임 내에 존재하는 DC 계수를 제외시켰으며, 블록 내에서도 휘도 부분만을 고려하면 하단 자막에서 나타내는 특징을 충분히 반영할 수 있다.

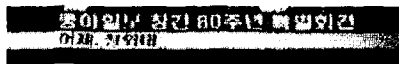


그림 4. 실제 검색 영역

하단 자막이 나타나는 프레임을 대표 프레임으로 추출하는 것이 목적이므로 I 프레임 내에서도 그림 4와 같이 프레임의 하단 부분으로 검색 영역을 제한하고, 한 블록 내에서 DCT 계수를 읽어오는 지그-재그(Zig-Zag) 순서대로 저주파 영역 다섯 부분을 그 비교 대상으로 삼았다. 여기서 저주파 영역을 다섯 부분으로 제한시킨 것은 너무 낮은 저주파수([1,0]과 [0,1])만으로는 이루어지는 검색의 경우는 하단 자막의 특징을 나타내기도 하지만 다른 자연 영상에서도 그 성분이 많이 포함되어 있어 오류일 가능성이 높기 때문이다.

8×8 블록 내에 들어가 있는 하단 자막이 들어있는 프레임은 본 논문에서 제안하고 있는 저주파수 영역([1,0], [0,1], [2,0], [1,1], [0,2]) 내에서 충분히 그 특징을 찾아 낼 수 있다.

DCT블록 내의 주파수 함수를 $f(\cdot, \cdot)$ 라고 가정하면, 다섯 개의 저주파 AC 성분 중에서 $|f[1,0]| > |f[0,1]|$ 은 가로 방향으로의 주파수 성분이 강함을 의미한다. 이와 같은 특성을 이용하여 식(2)에서와 같이 각 비교 결과를 얻는다.

$$\begin{aligned}
 A_n, m=1 & \text{ if } |fn, m[1,0]| > |fn, m[0,1]| \\
 B_n, m=1 & \text{ if } |fn, m[0,1]| > |fn, m[1,0]| \\
 C_n, m=1 & \text{ if } |fn, m[1,1]| > |fn, m[2,0]| \\
 & \text{ and } |fn, m[1,1]| > |fn, m[0,2]| \\
 D_n, m=1 & \text{ if } |fn, m[2,0]| > |fn, m[0,2]| \\
 E_n, m=1 & \text{ if } |fn, m[0,2]| > |fn, m[2,0]| \quad \dots(2)
 \end{aligned}$$

여기서, n, m 은 각각 n 번째 세로 블록과 m 번째 가로 블록을 나타낸다. 식(2)를 통해 얻어진 결과를 가지고, 식(3)을 사용하여 $H[n, m] > \delta$ 일 경우 하단에 자막이 들어가 있다고 판별하게 된다.

$$\begin{aligned}
 H[n, m] = & \sum_{n=0, m=0}^{n=N, m=M} A_n, m + B_n, m + 3C_n, m \\
 & + 5(D_n, m + E_n, m) \quad \dots(3)
 \end{aligned}$$

$H[n, m]$ 은 한 프레임 내에서 식(2)를 만족시키는 개수를 의미하며, 가중치는 하단 자막의 비율이 더 많이 들어가 있을 경우 비교 대상에서 우위를 보이도록 하기 위함으로, 실험 결과를 기초로 가중치를 산정하였다.

또한, δ 는 실험을 통해 적정하게 구해진 문턱값(Threshold value)이다. 식(2), 식(3)과 같은 연산을 사용함으로써 한 블록에서 다섯 개의 수치를 다섯 번 비교하게 되어 320×240 프레임에서 하위 1/3 부분을 검색하면 8×8 블록 단위로 40×10=400개만을 검색하는 결과가 되어 비교적 빠른 검색 속도를 보일 수 있었다.

이렇게 추출된 하단 자막이 존재하는 프레임에서 동일한 자막이 존재하는 프레임이 추출되는 것을 방지하기 위해, 검색 영역 값 H 를 기억해 두었다가 뒤에 나오는 프레임의 검색영역 값 H 와 비교하여 제한값 이상 차이가 나지 않을 경우 같은 하단 자막이 포함된 프레임으로 간주하였다.

3. 단어 빈도수에 기반한 프레임 추출

본 논문에서는 하나의 기사를 대표하는 대표 프레임으로 총 6개의 프레임을 추출하게 되는데, 앵커 프레임과 하단 공개 자막이 있는 프레임의 개수를 산정하여 5개 이하일 경우, 나머지를 단어 빈도수에 기반한 프레임을 추출하여 제시하도록 하였다.

단어 빈도수에 기반한 프레임의 추출을 위해서는 각 기사의 폐쇄 자막을 형태소 분석하여, 그 중 명사로 판명되는 단어를 뽑고, 이렇게 얻어진 단어의 빈도수를 헤아려 추출되어야 하는 프레임 수만큼의 상위 빈도수를 갖는 단어를 선정한다. 여기서 선정된 단어는 시간상 동기된 프레임을 추출하되, 앵커 태그 구간에서 나타난 단어는 제외하고, 기자 태그 구간에서 처음 등장하는 단어로 추출하도록 하였다. 이는 앵커 태그 구간에 나타난 단어와 동기된 프레임은 앵커 프레임일 경우가 대부분이기 때문이다.

III. 요약 문장 추출 및 브라우징

1. 요약 문장의 추출

기사에 대한 요약문장은 그림 5와 같이 기사를 분할하는 앵커 태그("앵커:")가 나왔을 때의 시점에서 기자

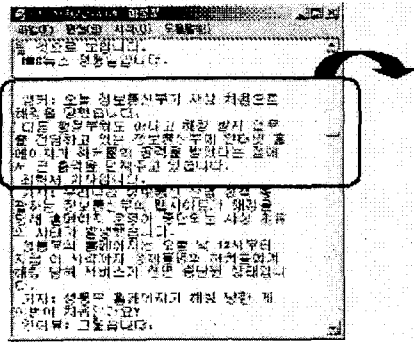


그림 5. 요약 문장의 추출

태그("기자:")가 나오기 전까지의 문장을 추출하여 나타내게 된다.

2. 브라우저

사용자는 뉴스 검색 시스템에 검색하고자 하는 키워드, 날짜, 기자명 및 기사 분류 등의 정보를 질의로 입력하여 기사 리스트를 얻게 되며, 이 리스트 가운데 하나의 기사를 선택하게 되면 기사 전체를 보기 이전에 본 논문에서 제시한 요약 시스템의 결과를 그림 6과 같이 확인하게 된다. 이 과정을 통해 사용자는 원하는 기사를 보다 빠르고 정확하게 검색할 수 있다.

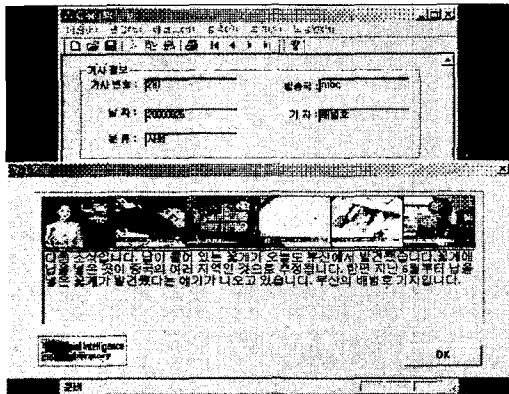


그림 6. 기사 결과 브라우저

IV. 실험 결과 및 고찰

앵커 프레임 추출은 식 (1)을 이용하여 100% 정확한 추출이 가능하다. 이 결과는 기존 연구에 비해 비교 연산 등의 추가 연산을 할 필요가 없어 추출 시간을 크게 단축할 수 있다.

하단에 공개 자막이 있는 프레임을 대표 프레임으로 추출하는 과정에서는 2000년 8월 26일 MBC 9시 뉴스에 적용한 결과, 뉴스 내에 존재하는 자막 프레임의 개수가 66개, 본 논문에서 제안한 방법으로 추출된 개수

가 62개, 그 중 정확하게 추출된 개수가 52개, 오검출된 개수가 10개이며, 오검출된 프레임중 4개는 화면 하단에 공개 자막은 아니지만 문자가 나타난 프레임으로 오검출 개수에서 제외할 수 있는 프레임이었고, 1개의 프레임은 이미 추출된 프레임이었다. 따라서 정확도가 83.9%, 재현율이 78.8%로 나타나 요약 시스템에서 사용하는 데에는 비교적 적합한 성능을 나타내었다.

본 논문의 전체 시스템 구성도는 그림 7과 같다.

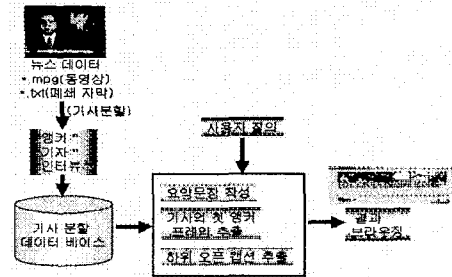


그림 7. 시스템 구성도

V. 결론 및 향후 연구과제

본 논문에서는 사용자가 미리 전체 비디오의 요약 결과를 확인하여 원하는 기사를 보다 빠르고 정확하게 검색할 수 있는 뉴스 기사 요약 시스템을 설계 및 구현하였다. 요약 과정에서는 뉴스 기사의 의미를 반영할 수 있는 대표 프레임 추출과 폐쇄 자막을 이용한 요약 문장이 제시되어, 사용자가 요구하는 의미에 기반한 검색에 적합함을 확인할 수 있었다. 향후에는 하단 공개 자막이 포함된 프레임의 추출 성능을 높이고, 폐쇄 자막을 이용하여 정리된 요약문을 제시할 수 있는 연구가 진행되어야 한다.

참고 문헌

- [1] Howard D. Wactlar, Takeo Kanade, Michael A Smith, Scott M. Stevens, "Intelligent Access to Digital Video: Informedia Project", IEEE Computer (Digital Library Initiative special), Vol. 29, No.5, pp. 46~52, May 1996.
- [2] Alexander G. Hauptmann, Michael J. Witbrock, "Informedia: News-on-Demand Multimedia Information Acquisition and Retrieval", Intelligent Multimedia Information Retrieval, AAAI Press/The MIT Press, pp. 215~240, 1997
- [3] 정승도, 조정원, 최병욱, "Closed-Caption을 이용한 뉴스 기사의 자동 분류", 멀티미디어 합동 학술대회 (JCM2000), 2000.6
- [4] 전근환 외7명, "뉴스 비디오 검색 시스템의 구현", 데이터베이스 연구회지 14권4호, pp.27-38, 1998.12