문서영상의 낱자 단위 언어 구분

권세광, 오일석
전북대학교 컴퓨터과학과
skkwon@cs.chonbuk.ac.kr, isoh@moak.chonbuk.ac.kr

Language Identification of Character-level in Document Image

Se-Kwang Kwon, Il-Seok Oh

요 약

본 논문은 문서 구조분석을 통해 얻어진 텍스트 영상에 대해 낱자 단위 분할 과정과 분할된 낱자에 대한 언어 구분 방법을 제안한다. 먼저 8방향 연결 요소를 이용한 레이블링을 수행하고 각 레이블의 거리관계와 한글 종모음의 특징을 이용하여 낱자 분리를 수행한다. 분리가 이루어진 낱자의 언어 구분은 각 낱자에 존재하는 concavity 특징을 이용하여 한글과 영어로 구분하게 된다. Concavity 특징을 찾기 위해 낱자를 이루는 흑화소 중 수직런을 이루는 흑화소 중 일부와 세리프 성분을 제거하며 그 방법을 기술한다. concavity 특징은 분리기를 통해 한글과 영어 두 가지로 분리되며, 분류기는 신경망을 이용한다. 제안된 방법은 20개의 텍스트 영상에 총 7923개의 낱자를 대상으로 실험하였으며, 낱자 분리는 97.20%의 정확도를 보였으며 분리된 낱자에 대한 언어 구분은 92.70%의 정확도를 얻을 수 있었다.

1. 서 론

본 논문에서는 인식이나 검색 시스템의 성능을 개선할 수 있는 방법의 하나로 낱자 단위 언어 구분 방법을 제안한다. 기존의 문서 영상에 대한 언어 구분은 대부분 단어 단위로 이루어졌으나 한글 문서 영상은 한문, 한글, 영어, 숫자, 특수기호 등이 혼용되어 있으며 이는 인식과 검색 시스템에 불필요한 작업 발생의 요인이 된다. [1]

낱자 단위 언어구분을 수행하기 위해 텍스트 영역에 대한 낱자 단위 분할이 이루어져야 하며, 분할된 낱자를 한글과 영어로 분리하게 된다. 낱자 단위 분할은 레이블링과 한글 종모음 특징을 이용하여 분리하게 되며, 분리된 낱자는 Concavity 특징을 이용하여 한글과 영어를 구분하게 된다. 분류 작업의 하나로 일부 수직획을 제거하는 과정을 거치며 이는 Concavity의 양을 늘리거나 줄여주는 역할을 하게 된다. 분리기는 신경망을 이용한다.

2. 낱자 분리

언어 구분 과정의 하나로 구조분석과 전처리를 통해 분리된 텍스트 역역에 낱자 단위 분리를 수행한다. 본 논문에서는 두 개의 낱자가 8방향 연결요소를 조사했을 때 하나의 연결요소로 이루어진 경우는 고려하지 않는다.

낱자 분리는 먼저 텍스트 영상을 입력으로 하여 8방향 연결요소 분석을 수행하여 얻어진 각각의 레이블링을 찾게 되며, 각각 레이블의 겹침 기울기 등을 조사하고 한글 낱자를 이루는 요소인 종모음에 대해 조사하여 낱자 단위 분할을 수행한다.

2.1 8방향 연결요소 레이블링

낱자 분리의 과정 중의 하나로 먼저 텍스트 영역에 대해 8방향 연결요소 레이블링을 수행한다 [2]. 레이블링을 통해 분리된 각각의 레이블은 한글의 경우 각 낱자에 평균 2개 이상의 레이블이 존재하고 영어는 한 개의 레이블로 이루어짐을 알 수 있다. 영어 소문자 i, j의 경우 두 개의 레이블로 나뉘어지는데 이는 2.2절에서 다루기로 한다. 이렇게 얻어진 각각의 레이블 객체는 언어구분 과정중 특징의 하나로 이용된다.

2.2 바운딩 박스

바운딩 박스의 겹침 정보를 이용하여 낱자단위 분할과 한글과 영어를 구분하는 특징으로 사용할 수 있다. 레이블의 겹침을 텍스트 영역에서 한글을 구분하는 조건 1로 하고 조건 1에 해당하는 레이블을 한글로 분류 한다. 영어의 경우 두 개 이상의 바운딩 박스가 겹치는 글자는 영어 소문자 i, j를 들 수 있다. 그림 1과 같이 i, j 는 영어에서 유일하게 2개의 레이블로 구성된다. 그리고 upper zone에 위치하는 레이블의 전체 흑화소는 레이블의 2/3 이상 차지하는 특징을 이용하여 찾는다. i, j를 만족하는 레이블은 영어로 분류 한다.

그림 1. 영문자 i, j

2.3 투영 특징 분석

각각 레이블의 바운딩 박스를 조사하여 겹침 정도를 구하여 한글 중 일부의 낱자 분리를 할 수 있다.

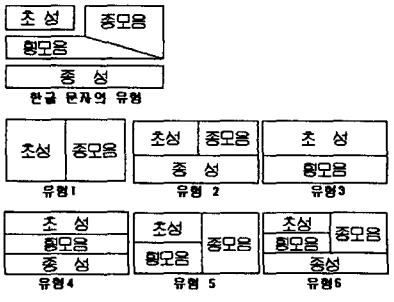

그림 2. 한글 문자의 유형과 6가지 형태

낱자 분리 과정중 한글을 그림 2과 같이 6가지 형식으로 분류할 때[김진형], 유형 1과 같은 그림 3.1의 경우 바운딩 박스의 겹침이 발생하지 않는다. 또 다른 문제는 이탤릭(italic) 스타일로 작성된 문서의 여러개의 문자에서 동시에 겹침이 발생하여 분리를 어렵게 만드는 경우가 있다.

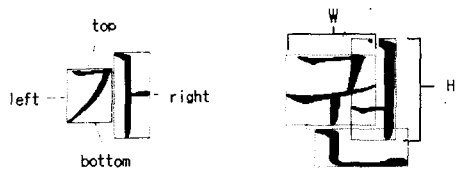

그림 3.1 유형1 그림 3.2 유형6
그림 3. 한글 분류 유형 1과 유형 6의 바운딩 박스

문서 영상의 특징 중 하나는 영어, 한자, 숫자 등이 포함 될 수 있다는 것이다. 식 1에서 추출된 특징을 통해 이러한 특징을 가진 레이블은 한글보다 영어에서 많이 발생하게 되는 것을 알 수 있다. 한글에서 발생하는 경우는 6형식 중 유형3의 앞에 위치한 글자가 모음 'ㅏ, ㅑ'를 갖게 되는 경우이다.

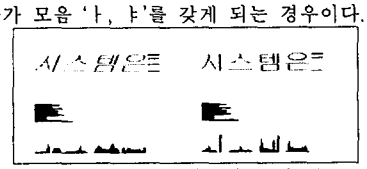

그림 4. 기울어진 문자 영상의 투영 히스토그램

그림 5는 수직방향 투영 프로파일의 히스토램을 통해 같은 글자가 이택릭 스타일을 적용함으로써 히스토그램이 변함을 보여준다 [5].

$$Pv[j] = \sum_{i=1}^{H} I(i, j)$$

$$where\ j = 1, 2, .., W \qquad 식\ 1.$$

식 1은 이진 영상의 텍스트 영상 I(W, H)에서 수직방향 투영을 통해 얻어지는 특징을 나타낸다.
이탤릭으로 처리된 글자의 경우 투영값의 수직 거리가 작게 나타나는 특징이 있다 [3].
식 2는 문자나 단어 사이의 여백 또는 문자의 폭이나 단어의 길이 변화를 고려한 특징 추출 방법으로써 본 논문에서는 각각의 레이블에 적용하여 특징을 추출한다.

$$Fp = \frac{1}{R}\sum_{j=1}^{W}(Pv[j+1]-Pv[j])^2$$

$$Pv[0] = Pv[W+1] = 0$$
$$R = Pv[j],\ R \neq 0 \qquad 식\ 2.$$

2.4 한글 종모음

한글의 낱자 분리를 어렵게 하는 특징 중의 하나가 초성과 종모음의 위치가 그림 3의 유형 1과 같은 경우이다. 한글 폰트 중 세리프가 없는 직선획을 갖는 폰트는 시스템과 돋움체를 예를 들 수 있는데에서 그림 4.1의 유형의 종모음 "ㅏ,ㅑ,ㅐ,ㅒ,ㅓ,ㅕ,ㅔ,ㅖ" 의 경우는 레이블의 겹침이 발생하지 않으므로 종모음을 찾는 루틴을 추가해야 한다. 유형 1을 만족하는 레이블을 찾기 위해 조건 1에 포함되지 않는 나머지 레이블을 조사한다. 표 1의 조건을 이용한다.

표 1. p-1과 p의 조건, p의 조건

p-1과 p의 조건

1 p-1 은 k와 p가 올 수 있다.
2 p-1의 left와 p의 right의 넓이는 p-1 혹은 p의 높이보다 작다
3 p-1과 p 각각의 top과 bottom이 동시에 같을 수 없다.

p의 조건

1 BB의 top은 p-1의 top보다 높다(p-1이 소문자일경우와 같 1)
2 넓이는 p-1의 넓이보다 작다(p-1이 소문자일 경우)
3 수평련은 두개 이하이다.
4 수평련 중심점은 수직련 중심에서 수직련 1/3길이 내에있다.

* k : 조건 1을 만족한 레이블
* p : 종모음 후보 레이블

표 1의 조건을 만족하는 레이블을 k로 한다. 낱자를 분리하면서 겹침이 발생하는 레이블과 종모음 레이블을 찾아 k가 되는 2개의 조건으로 사용하였다.

3. 언어구분

2절의 과정을 통해 k로 분류되는 낱자를 제외한 나머지 레이블 객체는 한글 중 한개의 레이블인 경우와, 영어(대,소문자), 숫자, 특수문자 등이다.
그림 7에 나타난 것처럼 소문자는 upper zone 과 middle zone의 경계 그리고, middle zone과 lower zone의 구분이 쉽다는 특징을 가지고 있다. 그러나 낱자 단위 분할 시스템에서 middle zone과 upper zone의 경계를 구별하는 기준을 정하기 어렵다. 예를 들어 한 줄에 영어 소문자가 한 개만 있다고 할 때 영어 소문자 26자 중 한개를 찾기 위한 조건을 만든 다는 것은 시스템의 성능을 저하시키는 요인이 될 것이다.
본 논문에서는 각 낱자에 존재하는 concavity 특징을 이용하여 영어와 한글을 구분하는 방법을 제안하였다 [7]. [7]에서 제안하는 방법은 concavity와 LDA(linear discriminate analysis) 등을 이용하여 단어 단위로 분류하는 방법이다. 본 논문에서는 낱자에 적용할 수 있는 방법을 제안한다.

3.1 체인코드

체인코드는 연결요소 레이블링을 수행하는 과정에서 조사할 수 있다. 체인코드는 8방향 연결성을 가진 이웃화소를 조사하여 방향을 표현한다. 그림 5의 0을 기준으로 반 시계방향으로 45° 씩 증가하고, 기준이 되는 중심 픽셀의 이웃화소에 이를 적용한다.

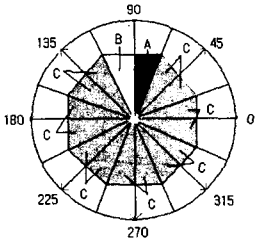

그림 5. 글자 획의 방향 성분

텍스트 영역에 존재하는 각 낱자의 일부 획을 제거하여 일부 특징을 만들거나 없앨 수가 있는데[4], 본 논문에서는 k를 제외한 나머지 레이블들은 concavity 특징을 얻기 위한 과정으로 그림 6의 조건을 만족하는 픽셀을 찾아서 제거 후보로 정하고 일부 픽셀을 삭제 하는 방법을 제안한다. 제거 후보는 그림 5의 0° 에서 90° 방향의 범위 내에 존재하게 된다. 제거 조건

조건은 제거 후보가 그림 8의 A의 범위에 포함 될 때이다. 단, 수평런(레이블 넓이의 1/3이상의 길이를 런으로 함)일 경우 후보에서 제외한다.

$$A = (\sum_{i=0}^{n-1} Ci)/2 \qquad \text{식 3}$$

* n = 제거 후보 픽셀

그림 6은 제거 조건을 만족하는 후보에 시작과 끝 점 사이에 존재하는 픽셀의 평균각을 구하여 그림 5의 A 조건에 해당하는 후보를 제거한 영상이다. 영어 대문자는 최대 두 개의 수직런을 가지고 있다. 영어의 경우 하나의 레이블로 이루어진 두 개의 수직런이 존재할 경우 그 중 하나를 제거함으로써 concavity의 발생을 줄일 수 있고 대문자 "B, D, E, F"등은 세리프가 없을 때 concavity가 존재하지 않는다. 만약 이탤릭 스타일의 폰트에 식 3을 만족하는 수직런이 존재한다면 제거 대상이 된다. 각각의 레이블 안에 존재하는 픽셀은 그림 5의 C에 해당하는 경우 각각의 중심각으로 교정한다. B의 경우 90° 로 교정한다. 그림 5의 A의 범위에 존재하는 픽셀은 제거한다. 단 제거는 최대 한 번만 이루어지게 되며 일부 레이블에서 제거조건이 안 될 경우는 제거하지 않는다.

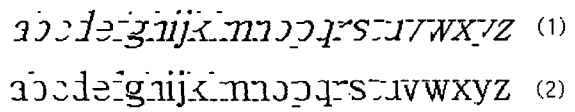

그림 6. 그림 8의 A를 제거한 결과 영상

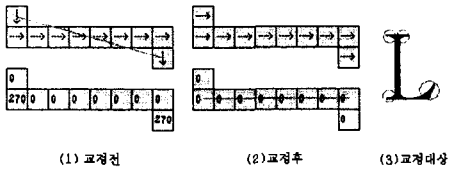

그림 7. 0° 방향 교정과 교정대상

방향각을 교정함으로써 그림 7의 (3)에서 볼 수 있는 세리프를 제거한다. 세리프는 글자를 이루는 필수 획이 아니면서 concavity를 발생시키는 요인이 되므로 제거 한다. 이와 함께 곡선은 식 3의 값에 따라 직선과 사선으로 교체된다.

3.2 concavity

concavity는 가운데가 오목하게 들어간 부분을 말한다 [7]. 그림 11은 upward concavity를 나타내고 있다.

그림 8. X와 X의 upward concavity

일반적으로 한글은 영어보다 많은 양의 concavity를 가지고 있다 [7]. upward concavity는 일부 한글에서 영어와 유사한 특징을 보이므로 본 논문에서는 upward concavity와 downward concavity를 이용한다. 각 레이블의 concavity는 신경망의 입력으로 사용된다.

3.3 정규화

추출된 concavity는 신경망의 입력으로 사용되기 위한 정규화 과정을 수행한다. 정규화를 위해 레이블의 폭과 높이가 일정하도록 공백을 삽입한다. 최대의 레이블의 폭과 높이 만큼 공백을 삽입하는 것이다 [6]. 공백을 삽입의 의미는 레이블의 폭과 높이를 늘려주는 의미이므로 그림 9에서 보는 것과 같이 화살표 방향으로 외접 사각형만 확대한다. 흑화소의 이동은 없도록 한다. 각각의 레이블 크기를 정규화하여 concavity의 수와 수직 모먼트의 평균과 수평 모먼트의 평균 값을 구하고 신경망을 통해 한글과 영어의 분류를 수행한다. 신경망의 입력은 convity의 수와 모먼트 값으로 하며 출력은 한글과 영어 두개로 구성한다. 낱자에 concavity가 존재 하지 않을 때는 낱자의 흑화소 모먼트를 입력으로 한다.

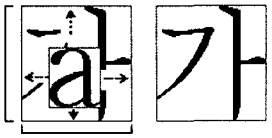

그림 9. 외접 사각형의 확대

4. 실험 결과 및 분석

제안 시스템은 Visual C 6.0을 이용해 구현하고 한글 윈도우 2000에서 펜티엄 1.8GHz에서 테스트 하였다. 테스트 DB는 논문지에서 20개의 텍스트 영역을 임의로 추출하였다. 각 텍스트 영상에는 한글, 영어, 숫자, 특수 기호가 혼용되어 있으며 DB에 존재하는 낱자의 수는 한글, 영어, 숫자, 특수기호 등을 포함하여 총 7923개이며 낱자 분리에서 97.20%의 정확률을 얻었으며 분리된 낱자에 대한 언어 구분은 92.70%의 성능을 얻을 수 있었다. 단 구분 대상에서 숫자와 특수기호는 정확도에 포함하지 않는다.

5. 결론

본 논문에서 낱자 단위 언어 구분 방법을 제안하였으며 제안 방법의 정확률을 개선하기 위해 이탤릭 스타일로 작성된 문서 영상영상에 대한 연구가 필요하며, 서로 다른 낱자의 일부화소가 연결 되어 있을 경우에 대한 연구가 차후 수행 되어야 할 것이다.

참고문헌

[1] 손홍석, '로컬 모먼트를 이용한 인쇄된 한글·영문 인식" 석사학위 논문, 전북대학교 컴퓨터 공학과 1995.
[2] 장명욱, 천대녕, 양현승, " 연결화소를 이용한 문서 영상의 분할 및 인식," 한국 정보과학회 논문지 Vol.20, No, 12, pp 1741-1751, 1993.
[3] 박문호, 손영우, 김석태, 남궁재찬, "인쇄된 한글 문서의 폰트 인식," 한국정보처리학회논문지, 제 4권, 제 8호, pp. 2017-2024, 1997.
[4] 이경철, 곽희규, 정선화, 김수형, "형식문서 상의 직선제거와 문자복원," 한국정보과학회 1999년도 춘계 학술발표논문집, pp. 555-557, 목포대학교, 1999.
[5] Peake,G.S. Tan, T.N. "Script and language identification from document images " pp.10 -17 (DIA '97) Proceedings., Workshop on , 20 June 1997.
[6]김도현, 강동구, 강민경, 차의영, "문자인식을 위한 효율적인 획 정규화", 한국정보처리학회, 추계학술발표논문집, 제8권 2호, pp.785-788, 2001.
[7]Spitz, A.L."Determination of the script and language content of document images "pp: 235 -245, Volume: 19 Issue: 3, March 1997.