碩 士 學 位 論 文

기계학습 기법을 이용한 문장경계인식

Sentence Boundary Detection Using Machine Learning Techniques

高麗大學校 컴퓨터情報通信大學院

미디어工學 專攻

朴 隋 革

2008 年 5 月

林 海 彰 敎 授 指 導

碩 士 學 位 論 文

기계학습 기법을 이용한 문장경계인식

Sentence Boundary Detection Using Machine Learning Techniques

이 論文을 工學 碩士學位 論文으로 提出함

2008 年 5 月

高麗大學校 컴퓨터情報通信大學院

미디어工學 專攻

朴 隋 革

朴隋革의 工學 碩士學位論文 審査를 完了함.

2008 年 5 月

委員長　林 海 彰 (印)

委　員　陸 東 錫 (印)

委　員　李 晟 換 (印)

요 약

2007년 2월 통계에 따르면, 현재 웹에는 297억 개의 웹 페이지가존재하고 있으며, 2008년 현재 300억 개를 넘어섰을 것으로 추정하고 있다.

이렇듯 넘쳐나는 많은 전자문서들의 내용을 컴퓨터를 통하여 효과적으로 이해 및 전달을 위해서는 형태소분석, 구문분석, 문맥 및 의미분석 등의 자연어처리 기술이 필요한데, 이러한 자연어 처리의 가장 기본 단위인 '문장'을 구분하는 작업이 요구된다. 하지만 '문장'에 대한 정의만 해도 몇 백 가지가 넘고 문법적으로 정의되어 있지 않아, 문장을 인식하는 것 또한 어려움이 있다. 일반적으로는 문장을 인식하는 데에는 문장부호가 주로 사용되나, 그것 또한 생략되거나 잘못 표기된 경우가 많아 모든 문장을 파악하기에는 어려운 점이 있다.

본 논문에서는 문장부호 또는 문법적인 규칙을 통한 규칙기반의 문장경계 인식기를 구현하여 실험 및 검증을 하고, 규칙으로는 어려운 문제들을 해결하기 위하여 언어의 통계적 특징을 활용하여 보다 범용적인 문장경계 인식기를 제안한다. 이는 대량의 코퍼스 내에서 사용되고 있는 문장 경계를 기준으로 음절 및 어절 등의 자질을 이용하여 통계적 특징을 추출하고 기계학습 기법을 활용하여 문장경계를 인식하고자 하였으며, 특정 언어나 도메인에 제한적이지 않고 범용적인 자질만을 사용하려고 노력하였다.

언어의 특성상 문장의 구분이 애매한 경우, 문장부호로 문장이 종료되지 않은 경우 또는 잘못 사용 된 문장부호 등의 경우에도 적용 가능하도록 다양한 자질을 사용하여 실험하였으며, 한국어와 영문 코퍼스에 대해서 동일한 자질을 적용하여 실험하여 본 논문에서 제시한 자질들이 한국어 이외의 다른 언어에서도 적용될 수 있는 범용적인 자질임을 확인할 수 있었다.

Abstract

Web documents have grown to 29.7billions Feb. 2007. Probably more than 30billions documents are there at now.

We need part of speech detecting, context analysis and semantic analysis for using these electronic documents and also understanding efficiently. All these techniques are needed disambiguating sentence boundary or detecting end of sentence. But lots of definitions of sentence give us confusing while detecting sentence boundaries.

Generally we use punctuation mark for finding sentence boundary, but there are many sentences which does not have punctuation mark or using punctuation marks wrongly. So we need more sophisticated solution for solving sentence boundary disambiguation.

This paper suggests general purpose sentence boundary detection system which uses language statistical information gained from corpus like syllables or lengths around sentence boundary. Besides I tried to use general purpose features which is not related with special domain or language.

I tried to learn features through using machine learning techniques empirically. also experimented for two kinds of language Korean and English to confirm the general purpose system. Finally we found out that these features are applied to both languages.

There is little modifications for experiments with before, and reasonable result came out.

목차

그림 목차

표 목차

1. 서론

1.1. 연구의 배경 및 목적

‘문장’의 사전적인 의미는 “의사를 전달하는 최소의 단위.”라고 정의되어 있고, 전통 문법에서는 “비교적 완전하고 독립된 의사전달 단위다.”라고 정의하고 있으며, 이러한 문장의 단위가 대부분의 자연어 처리 도구, 품사부착기 등의 기본단위가 되고 있다[14]. 또한 음성 인식된 문장 또는 OCR처리된 문서 등과 같이 문장경계가 모호한 경우에 대해서도 전처리 과정으로도 문장경계의 인식작업이 사용되고 있다.

하지만 ‘문장’의 정의가 문법적으로 명확하게 정의되어 있지 않으며, 문장부호 등의 사용도 불규칙적이어서, 문장경계의 구분이 명확하지 못한 경우가 많으며, 전문가에게 문의하지 않고는 제대로 구분되지 않는 경우도 있다. 대체적으로 마침표, 물음표, 느낌표 등의 문장부호를 기준으로 문장경계가 구분된다. 그러나 경우에 따라서 “문장의 시작번호, e-mail, 영어의 약어, 강조를 위한 반복, 열거 등에 사용되는 마침표 또는 잘못 사용된 구두점” 등의 경우와 같이 기계적으로 보았을 때에 문장부호 만으로는 문장경계를 판단하기에는 부족한 듯 보인다.

따라서 문장경계 인식의 가장 좋은 방법은 각 분야의 전문가가 해당 위치를 인식하는 규칙을 정하는 것이 가장 이상적일 수 있으나, 너무 많은 비용이 소모될 것으로 예상되어 추천할 만한 방법이 아니며, 본 논문에서는 규칙에 기반한 문장경계 인식기를 통하여 실험

및 검증을 실시한 후, 그에 따른 문제점들을 살펴보고, 이를 보완하기 위하여, 기계학습 기법을 사용하여 문장경계를 판단해낼 수 있는 문장경계 인식기를 제안하고자 한다.

여기서 문장경계 인식은 문서 내에서 공백으로 구분되는 모든 위치에서 문장경계인지 아닌지를 분류하는 문제로 정의해볼 수 있겠다.

다양한 언어에 대한 문장경계 인식은 최대한 많은 언어에 대한 실험이 필요하겠으나, 코퍼스 또는 해당 언어에 대한 이해의 부족으로 본 논문에서는 한국어 및 영어에 대해서만 실험하기로 한다.

기계학습에 사용된 도구로는 WEKA와 Maxent라는 학습도구를 사용하였고, 제공되는 다양한 분류 알고리즘을 통하여 실험 및 검증을 하였다. 검증방법으로는 정확률과 재현율을 사용하였으며, 학습에 사용된 말뭉치는 한글의 경우 '21세기 세종계획 말뭉치'의 구분 문석 결과를, 영문의 경우는 Wall Street Journal의 문장단위로 구분된 말뭉치를 사용하였다. 문장경계의 후보위치는 모든 어절 즉, 공백을 기준으로 실험 하였으며, 개별 어절은 다시 문장부호 또는 언어별로 별도의 토큰으로 분리해 내어 분석을 시도하였다.

1.2. 논문의 구성

본 논문은 다음과 같이 구성되어 있다. 2장에서는 국.내외 관련연구 및 본 논문에서 실험한 기법 및 실험도구 등을 살펴보고, 3장에서는 문장경계 인식을 위한 자질 및 방법에 대한 설명을 할 것이며, 4장에서는 기계학습 기법을 이용한 문장경계 인식의 실험 및 평가결과를 제시한다. 마지막으로 5장에서는 결론과 향후 과제에 대해 서술한다.

2. 관련연구

2.1 문장 및 문장경계의 정의와 범위

2.1.1 문장의 정의

문장의 정의는 현대 국어에서도 명확하게 그 경계를 정의하고 있지 못하고 있으며, 특히 한국어의 경우 그 경계가 모호하여 명확하게 구분하는 것이 어려울 수 있다.

사전적 의미에서의 문장이란 “사상이나 느낌을 단어를 연결하여 의사를 전달하는 최소의 단위”라고 말하고 있으며, ‘문’ 또는 ‘글월’이라고도 한다. 전통문법에서는 “명제(命題) 와 구별되는 하나의 완결된 의미를 지닌 단어의 집합.”으로 정의되기도 하나 동일한 문장이, 하나의 문장 “나는 책을 읽고 어머니는 바느질을 한다.” 또는 두 개의 문장 “나는 책을 읽는다. 어머니는 바느질을 한다.”로 표현될 수 있으므로 완결된 의미라는 단위 설정을 일관성 있게 할 수 없기 때문에 문장은 “음의 연쇄체이며, 앞과 뒤에는 휴지(休止)가 놓이고, 끝에는 특수한 억양이 놓이는 것.”으로 하여 의미를 배제하고 정의하는 수도 있으나, 이것도 만족할 만한 정의는 되지 못한다. L.블룸필드는 “문법적으로 더 큰 언어적 구성에 내포되지 않는 독립적인 언어형식.”이라고 정의하였다[14].

한편, 변형문법에서는 기술적(記述的)인 관점에서 기저부문(基底部門)의 공리(公理:axiom)로 정의되어 시발기호(始發記號:initial symbol)로 취급된다. 즉, S→NP+ VP(문장→명사구+ 동사구)의 S가 문장이다.

문장의 종류는 주어-서술어의 관계가 단 한 번만 성립하는 단문(單文)과 이 관계가 두 번 이상 성립하는 복문(複文)으로 구별할 수 있으며, 서법(敍法)에 따라 평서문·의문문·감탄문·명령문·청유문 등으로 나눌 수 있다[14].

2.1.2 문장부호의 정의

문장부호란? 문장 각 부분 사이에 표시하여 논리적 관계를 명시하거나 문장의 정확한 의미를 전달하기 위하여 표기법의 보조수단으로 쓰이는 부호로써, '구두점'이라고도 한다. 문장부호는 순수하게 논리적인 목적에서 사용되는 경우와, 어조상(語調上)의 쉼을 위하여 사용할 때와는 차이가 있다. 시(詩)에서는 리듬을 위해 사용하기도 한다.

문장부호의 기원적 형태는 한문 원전(原典)을 읽을 때 독해(讀解)의 필요에서 찍는 훈점(訓點)에서 찾을 수 있으며, 표현을 위해 사용된 문장부호의 발달은 로마자가 한국에 소개되면서부터 차용·발전한 것이다. 현재 가장 널리 쓰이는 명칭에는 마침표, 쉼표, 따옴표, 묶음표, 이음표 등이 있다. 보다 상세한 종류, 예제 및 설명은 다음과 같다[28].

종류	예제	설명
마침표	.(온점), 。(고리점)	서술형·명령형·청유형의 글에서 문장이 종결 어미로 끝남을 보일 때, 숫자의 정수(整數) 단위를 표시할 때, 또는 구미어의 약자 뒤나 연월일을 표시하는 숫자 뒤에 쓰인다. 가로쓰기에서는 아래 쪽 가에, 세로쓰기에서는 글줄의 가운데 찍는다(예:22.5kg, Mr., 2004.9.20. 등).

	?(물음표)	직접 의문이나 반어(反語) 및 수사의문(修辭疑問) 또는 가벼운 감탄을 나타낼 때 쓰인다.
	! (느낌표)	강한 느낌이나 부르짖음을 나타내는 감탄사나 감탄형 및 원형의 종결어미 뒤에 쓰이며, 명령·의문·권유의 글에서도 느낌을 강조할 때 또는 느낌을 가지고 사람을 부를 때도 쓰인다.
쉼표	,(반점)	의미가 중단되어서 읽을 때에 잠깐 쉬는 것이 좋을 자리에 찍는다. 또 나열항목을 구분할 때나 직접 다음에 오는 말을 수식하지 않을 때, 가벼운 감탄을 나타낼 때, 부르는 말이나 대답하는 말 뒤 또는 제시어(提示語) 아래에 쓰이며, 정수 단위의 숫자를 세 자리마다 구분할 때도 쓰인다(예:5,000,000원).
	·(가운뎃점)	몇 개의 단어를 나열할 때, 또는 두 숫자로 된 말 사이에 쓰인다(예:오이·수박·참외, 3·1절, 6·25전쟁).
	:(쌍점)	이미 서술한 말에 내포되는 사항을 다시 자세히 설명하거나 예로 들 때(예: 과일:사과·배·감 등), 한 문장이 끝나면서 다음 문장과 의미상 연결됨을 보일 때, 세부사항을 나열하고 그것을 다시 묶을 때, 긴 인용이나 진술을 이끌어 낼 때, 저자와 책이름 사이 등에 쓰이며, 비율 표시로도 쓰인다.
	/(빗금)	대응·대립되거나 대등한 것을 함께 보이는 단어·구·절 사이에 쓰고, 대등한 것으로 '또는'을 나타낼 때, 분수를 나타낼 때에 쓰기도 한다(예:선우 훈/선우훈, 이백삼십 원/230원, (○○)이/가, 1/4분기, 2/10).
따옴표	" "(큰따옴표) 『 』(겹낫표)	글 가운데 직접 대화를 보이고자 할 때, 남의 말을 직접 인용할 때 쓴다.
	' '(작은따옴표) 「 」(낫표)	특별히 쓰이는 말, 특히 강조하여 주의를 돌리려는 말과 신문이름·책이름·제목 등을 두드러지게 보일 때, 또 글월 가운데서 마음 속으로 생각하는 것 등을 보일 때

		와 따온 말 가운데서 다시 따온 말이 들어 있을 때에 쓰인다.
묶음표	( )(소괄호) { }(중괄호) [ ](대괄호)	묶음표는 다른 글과 구별하고자 하는 부분의 앞뒤에 쓰는데, 소괄호는 원어·연대·주석 등을 넣을 때, 기호 또는 기호의 구실을 하는 문자·단어·구에 쓰고, 중괄호는 여러 단위를 동등하게 묶을 때 쓰며, 대괄호는 '꺾쇠묶음'이라 하여 수학에서 자주 쓰인다.
이음표	-(줄표)	글 중간에 따옴표 모양으로 어구를 넣을 때 그 앞뒤에 쓰고, 여러 개를 나열하여 하나로 통일시키거나, 하나의 통일된 데서 여러 개로 나열시킬 때 등에 쓰인다.
	-(붙임표) ~(물결표) ……(줄임표)	한글맞춤법에는 붙임표, 물결표, 줄임표 이 밖에 모두 30종의 부호가 있다.

2.1.3 문장부호에 따른 문장경계

위에서 제시했던 문장부호에 따른 다양한 사용 예가 있으며, 이에 따라 동일한 문장부호라고 하더라도 문장경계에 사용될 수도 그렇지 않을 수도 있으며, 경우에 따라서는 경계인 경우도 있고 그렇지 않은 경우도 있었다. 이에 따른 규칙을 표로써 정리해 보았으며, 이 표는 규칙기반 문장경계 인식기에서도 활용된다[28].

문장부호	종류	설명	용례	경계
마침표	온점(.) 고리점(。)	서술, 명령, 청유	집으로 돌아가자.	○
		아라비아 숫자	1974.10.30	X
		표시문자	1.마침표	X
		준말	서.1987.3.5	X
	물음표(?)	직접질문	이름이 뭐지?	○
		반어, 수사의문	이게 은혜에 대한 보답이냐?	○

		소괄호 표현	참 훌륭한(?) 태도야.	X
	느낌표(!)	감탄사, 감탄형	앗! / 아. 달이 밝구나!	○
		강한 명령문, 청유문	지금 즉시 대답해!	○
		부르거나 대답	춘향아! / 예, 도련님!	○
		놀람, 항의	누구야! / 내가 왜 나빠!	○
쉼표	반점(,) 모점(、)	열거	근면, 검소, 협동은….	X
		짝	닭과 지네, 개와 고양이는….	X
		간접 꾸밈	성질 급한, 철수의 누이….	X
		대등, 종속	콩 심으면 콩 나고, 팥 난다.	X
		부르는/대답하는 말	얘야, 이리 오너라. / 예, ….	X
		제시어	빵, 빵이 인생의 전부이더냐?	X
		도치된 문장	이리 오세요, 어머님.	X
		가벼운 감탄	아, 깜빡 잊었구나.	X
		문장 첫 접속, 연결	첫째, 몸이 튼튼해야….	X
		끼어든 구절 앞뒤	나는, 솔직히 말하면, 그….	X
		되풀이 줄임	여름에는 바다에서, 겨울….	X
		문맥상 끊어 읽는 곳	갑돌이가 울면서, 떠나는….	X
		숫자의 나열	1, 2, 3, 4	X
		수의 폭, 개략의 수	5, 6 세기6, 7 개	X
		수의 자릿점	14,314	X
	가운뎃점(·)	쉼표열거의 분리	철수·영이, 영수·순이가 서로….	X
		특정의미의 날	3·1운동 8·15 광복….	X
		같은 계열의 단어	동사·형용사를 합하여….	X
	쌍점(:)	내포되는 종류	문방 사우: 붓, 먹, 벼루, 종이.	△
		소표제 뒤 설명	마침표: 문장이 끝남을….	△
		저자, 저서명	정약용: 목민심서, 경세유표.	△
		시분초, 장절, 대비	오전10:20/요한3:16/대비	X

			65:60	
	빗금(/)	대등, 대립	남궁만/남궁 만	○
		분수	3/4분기, 3/20	X
따옴표	큰따옴표(“ ”) 겹낫표(『 』)	직접대화	“전기가 없었을 때는 ….”	○
		직접인용문	“민심은 천심이다.”라고 ….“	△
	작은따옴표(‘’) 낫표(「 」)	인용의 재인용	“여러분! ‘하늘이 무너져도 솟아날 구멍이 있다.’고 합니다.”	△
		마음속의 말	‘만약 내가 이런 모습으로 돌아간다면, 모두들 깜짝 놀라겠지.’	○
		강조(드러냄표 대신)	지금 필요한 것은 ‘지식’이 아니라 ‘실천’입니다.	X
묶음표	소괄호(( ))	원어, 연대, 주석, 설명	커피(coffee)는 기호 식품이다. / ‘무정(無情)’은 춘원(6·25 때 납북)의 작품이다.	X
		기호, 기호적인 구실	(1) 주어 / (ㄱ) 명사	X
		빈 자리	우리 나라의 수도는 ( )….	X
	중괄호({ })	동등한 단위 표현	주격조사 { 이 가 }	X
	대괄호([ ])	묶음표 안과 상이한 발음	나이[年歲]/ 낱말[單語]/ 手足[손발]	X
		묶음표 내 묶음표	명령에 있어서의 불확실[단호(斷乎)하지 못함]은 복종에 있어서의 불확실[모호(模糊)함]을….	X
이음표	줄표(—)	부연설명	그 신동은 네 살에 — 보통 아이 같으면 천자문도 모를 나이에 — 벌써 시를 지었다.	△
		정정 또는 변명	어머님께 말했다가 — 아니,	△

			말씀 드렸다가 — 꾸중만….	
	붙임표(-)	사전 논문의 합성어, 접사나 어미	겨울-나그네/ 불-구경/ 손-발 휘-날리다/ 슬기-롭다	X
		외래어와 고유어 또는 한자어가 결합	나일론-실/ 디-장조/ 빛-에너지/ 염화-칼륨	X
	물결표(~)	'내지'라는 의미	9월 15일 ~ 9월 25일	X
		앞뒤의 말 대신	-가(家):음악~미술~	X
드러냄표	드러냄표(ˇ,°) 밑줄(___)	중요한 부분	한글의 본 이름은 훈민정음 이다.	X
		밑줄	보기에서 명사가 <u>아닌</u> 것은?	X
안드러냄표	숨김표 (××, ○○)	금기어, 비속어	배운 사람 입에서 어찌 ○○○란 말이 나올 수 있느냐? 그 말을 듣는 순간 ×××란 말이 목구멍까지 치밀었다.	X
		비밀의 유지할 경우	육군 ○○부대 ○○○ 명이 …. / 참석자는 김×× 씨,….	X
	빠짐표(□)	불분명 한 내용	大師爲法主□□賴之大□薦	X
		글자가 들어갈 자리	훈민정음의 초성 중에서 아음(牙音)은 □□□의 석 자다.	X
	줄임표(……)	할 말을 줄임	"어디 나하고 한번…….”하고 철수가 말했다.	△
		말이 없음	"빨리 말해!" "……."	○

2.1.4 문장경계의 정의

다음과 같은 문장의 경계를 구분한다면, 몇 개의 문장으로 구분할 수 있을지를 고민해 보았다.

> 『어제 김감독이 "열심히 노력하라! 그렇지 않으면 국물도 없다."라고 했다.』

앞에서 정의된 문장의 정의에 의하면, 아래와 같은 세 가지의 생각으로 구분하여 말할 수 있다.

문장1. 어제 김감독이 ~ 라고 얘기했다. (김감독이 말한 자체)

문장2. 열심히 노력하라. (인용문 안의 첫 번째 문장)

문장3. 그렇지 않으면 국물도 없다. (인용문 안의 두 번째 문장)

하지만, 경우에 따라서 위와 같이 세 개의 문장으로 구분하지 않고 하나의 문장만으로도 충분히 의미가 있을 수도 있다. 결국 문장의 구분이 경우에 따라서 어느 정도 수준의 문장을 사용할 것인지에 따라서 세 가지를 다 추출할 수도 그렇지 않을 수도 있을 것이다.

인터넷 백과사전 Wikipedia에서는 문장경계인식을 다음과 같이 정의하고 있다[25].

> Sentence Boundary Disambiguation (SBD) is the problem in natural language processing of deciding where the beginning and ends of sentences are.

"자연어처리 영역에서 문장의 시작과 끝의 위치를 결정하는 문제."라고 정의하고 있다. 여기에서 정의된 문장경계라고 하면, 문장의 시

작위치와 끝 위치를 모두 파악해 내는 것인데, 인용문에서와 같이 하나의 문장에서도 여러 문장을 추출해 낼 수 있으므로, 문장들이 겹치는 문제가 발생할 수 있다.

본 논문에서는 위에서 논의된 안긴문장 형태의 인용문의 경우, 인용문을 포함한 전체 문장을 하나의 완결된 문장으로 보고 있으며, 문장경계의 시작과 끝을 별도로 구별하지는 않고 있다. 즉, 문장경계의 위치를 판단하고 그 다음부터가 새로운 문장의 시작으로 보고 실험 및 구현을 하였다.

2.2 관련 연구

2.2.1 국내외 사례

Riley, Michael D. (1989)는 구두점이 발생하는 주변 단어의 출현 확률 및 구두점이 발견된 어절의 클래스 등의 자질을 추출하였으며, AP News 2천 5백만 단어를 통해 실험하였으며, Brown 코퍼스 에서 Decision Tree (C4.5)를 이용한 결과 99.8%의 정확률을 보였다 [13].

David D. Palmer, Marti A. Hearst, (1994)는 구두점 주변의 단어에 대한 품사의 확률정보를 이용하여 20가지 정도의 토큰으로 구분하였고, Feed-forward Neural Network를 이용하여 실험한 결과 98.5%의 정확률을 보였다[4][5].

Jeffrey C. Reynar and Adwait Ratnaparkhi, (1997)는 구두점 후보가 발생한 앞/뒤 토큰의 확률정보를 이용하였으며, Maximum Entropy 기법을 이용하여 Wall Street Journal 및 Brown 코퍼스에서 각각 98.0%, 97.5%의 정확률을 보였다[8].

임희석, 한군희 (2004)는 후보 구두점 자체의 확률, 앞/뒤 발생하는 음절 그리고 인용부호의 개수를 자질로 이용하였으며, kNN알고리즘으로 ETRI, KAIST 코퍼스에서 각각 96.73%, 98.64%의 정확률을 보였다. 또한 두 코퍼스를 모두 학습한 경우에는 98.82%의 정확률을 보였다[6].

2.2.2 기존 연구의 문제점

영어에 대한 연구는 많이 이루어졌으나 한국어에 대한 연구는 아직 많지 않은 실정이며, 또한 기존에 연구된 한국어 문장경계인식의 논문에서도 언급하였듯이 한국어 문장경계에 대한 다양한 알고리즘을 통한 실험이 필요하다. 또한 개별 언어에 대해서만 실험 및 검증이 이루어지고, 학습에 사용된 자질이 다른 도메인 또는 다른 계통 언어에 대한 실험 및 검증이 제대로 이루어지지 않았다고 볼 수 있겠다.

최근 웹 자료 및 문서의 경우 다양한 언어의 자료가 많이 발생하며, 그러한 범용적인 자질에 대한 연구가 필요하다고 생각되며, 이에 문장경계에 필요한 다양한 자질을 기계학습 기법을 통하여 학습하고 한국어 문장경계 뿐만이 아니라 영어에 대하여도 실험을 하고자 한다.

2.3 문장경계 인식을 위한 기법들

2.3.1 규칙에 의한 방법

우선 일반적인 규칙에 의한 문장경계 인식을 수행하고 그 결과에서 예외적인 사항을 다시 적용하는 방법으로 규칙에 의한 문장경계 인식을 수행하였다.

기본적인 규칙은 아래의 다섯 가지 문장부호가 발생한 경우를 후보로 정의하고 문장경계로 판단한다. 단, 아래와 같은 예외규칙을 적용하기로 한다.

문장경계 후보를 위한 문장부호

문장부호	사용예제
마침표	에고! 억울하고 분하고 원통해.
물음표	할아버지 왜 그러셔요?
느낌표	전국에 계신 독자 여러분!
큰/작은 따옴표	'뭐가 또 있습니까?', "저는 두 가지 이유가 있다고 봐요"

위의 규칙에 예외적인 규칙을 아래와 같이 적용하고 아래의 규칙에 해당하지 않는 경우를 문장경계로 판단한다.

숫자나 영문자가 마침표가 결합되어 나오는 경우는 문장경계로 인식하지 않는다.

Fr. Out

한자, 나라이름 또는 사람이름 등의 나열을 마침표를 사용하여 표현하는 경우는 문장경계로 인식하지 않는다.

한국. 미국. 영국.

강조를 위해 반복된 문장부호의 경우 마지막 문장부호만이 문장경계를 위한 부호로 인식한다.

나의 고향은 부산이다!!!

이외에도 규칙으로 표현하기 어려운 다양한 경우가 있는데 정리해보면 다음과 같다.

1. 인용부호 (작은/큰 따옴표) 규칙

1. 문장의 가운데에 "인용부호 + 공백 + 문자열" 패턴이 발견되면 인용부호를 기준으로 문장을 구분
2. 문장의 가운데에 "문자열 + 공백 + 인용부호" 패턴이 발견되면 문자열을 기준으로 문장을 구분
3. 인용부호는 시작 또는 종료 인용부호를 구분하지 않으며, 2바이트 문자열을 변환하여 적용한다.
4. 단, 문서에 따라서 인용부호가 쌍따옴표 또는 홑따옴표가 혼용되어 사용될 수 있다.
   (쌍따옴표 또는 홑따옴표의 경우 용도에 따라서 인용부호로 사용되는 경우도 있고, 강조로 사용될 수 있으므로 주의.)

2. 홑따옴표의 규칙

1. 인용으로 사용되기 보다는 주로 강조로 사용된다.
2. 문장 가운데에 "홑따옴표 + 공백 + 홑따옴표" 패턴이 발견되면 앞의 홑따옴표를 기준으로 문장으로 구분 (정규식: %s/['|‘] ['|‘]/'₩r‘/gc)

3. 감탄사 규칙

1. 문장의 가운데에 "(공백 or 인용부호) + 1음절한글 + (느낌표, 물음표, 마침표)" 패턴은 문장경계 아님
2. 문장 처음 또는 가운데에 "1~3음절한글 + (느낌표, 물음표, 마침표)" 패턴은 문장경계 아님

4. 숫자/영문자 규칙

1. "(숫자, 1~3음절영문자) + 마침표" 패턴은 문장경계 아님
2. "4음절이상영문자 + 마침표" 패턴은 문장경계
3. 한글과 영문이 섞인 경우에는 규칙적용이 어려울 수 도 있음

5. 인용문 규칙

1. "문자열 + 마침표 + (하고,라고,이라고)" 패턴이 발견되면 문장경계 아님

6. 특수기호 규칙

1. 문장 가운데에 "문자열 공백 + 특수기호(△)"패턴의 경우 문자열을 기준으로 문장경계

7. 물음표 규칙

1. "(?)"는 문장경계 아님.

8. 줄임말 규칙

1. "1~2음절한글 + (마침표,줄임표[…]) + 한글" 패턴의 경우 문장경계 아님

9. 나열식 규칙

1. 문장 가운데에 "5음절이내(한글, 한자) + 마침표 + 5음절이내한글"의 패턴이 2번 이상 반복이 이루어지면, 문장경계 아님
2. 하지만, 나열식 규칙이 1번만 반복되는 경우는 분석해내기 힘들고, 특히 한자, 나라이름, 국가명 등이 이러한 나열식이 많다.

10. 강조문자 규칙

1. "문장부호(느낌표, 물음표, 마침표) 반복" 패턴은 마지막이 문장경계

11. 슬래시 규칙

1. "문자열 공백 + 슬래시[/] + 공백" 패턴의 경우 문자열도 문장경계이고 슬래시 자체도 문장경계이고 하나의 문장은 아니지만, 공백으로 나누어지므로 구분한다

12. 괄호규칙

1. "닫는괄호[)] + 마침표" 패턴은 문장경계 아님

13. 브래킷 규칙

1. 문장 가운데에서 브래킷이 인용부호로 사용되는 경우에는 문장경계로 인식

이러한 규칙을 모든 학습집합에 잘 적용되는 것은 아니며, 현재 실험한 집합에서 추출한 규칙이다. 그리고 검증을 위하여 90%정도의 말뭉치만을 통해서 추출하였다. 실제 실험에서는 모든 규칙을 다 적용한 것은 아니며, 정확률과 재현율을 고려하여 일부 규칙만을 사용하였다.

2.3.2 의사 결정 트리

결정 트리 학습기법은 가장 널리 사용되는 귀납적인 추론 방법 중의 하나이며, 의사결정규칙을 트리구조로 표현하여 분류와 예측을 수행하는 분석 방법이다. 결정트리의 종류에는 ID3, ASSISTANT 그리고 C4.5 등이 있다. 그리고 일반적으로 아래와 같은 장점과 단점을 가진다[9].

표 1 의사 결정 트리의 장.단점

장점	단점
노이즈 데이터에 대한 강건성 분류 또는 예측의 과정의 규칙추출 도메인에 대한 지식 또는 설정 불필요 직관적이며, 구현이 용이하다 간단하고 신속한 학습과 분류가능 상대적으로 높은 정확률 점진적인 학습이 가능	연속적인 데이터에 대한 분류는 부적합 분류 선택을 위하여 어떠한 노드를 선택할 지에 대한 알고리즘 결정이 전체 정확률에 많은 영향을 미친다.

2.3.2.1 엔트로피와 정보이득

결정 트리를 구성하는 데에 있어 가장 중요한 어떤 속성을 선택할 것인지에 대한 방법 중에서 가장 일반적인 방법이 엔트로피(Entropy)값 계산을 통하여 정보이득(Information Gain)을 이용하는 것인데 다음과 같은 방법으로 계산한다.

$$Entropy(S) \equiv \sum_{i=1}^{c} - p_i \log_2 p_i$$

사건 S의 엔트로피는 S의 모든 가능한 결과값 i에 대해 i의 발생 확률 값인 p(i)과 그 확률의 역수의 로그 값의 곱의 합이 된다. 엔트

로피 값은 불확실성의 척도로써 사용되는데, 예를 들어, 10개의 문자열을 2진 문자로 표현하기 위해서는 총 몇 비트가 필요한 지를 통하여 표현될 수 있다.

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

이 수식에서는 i가 + 혹은 - 두 가지의 경우이므로 위와 같이 수식이 만들어 졌는데, 예를 들어보면, 동전을 던져, 앞면이 나오는 경우(+)와 뒷면이 나오는 경우(-)의 엔트로피 값을 구한다고 하자, 동전의 모양이 아주 일정하여 앞/뒷면이 나올 확률이 각 각 1/2인 경우에는 엔트로피의 값이 1이 나오며, 그 반대로 앞면이 나올 확률이 1이고 뒷면이 나올 확률이 0라면, 엔트로피 값은 0이 된다[21].

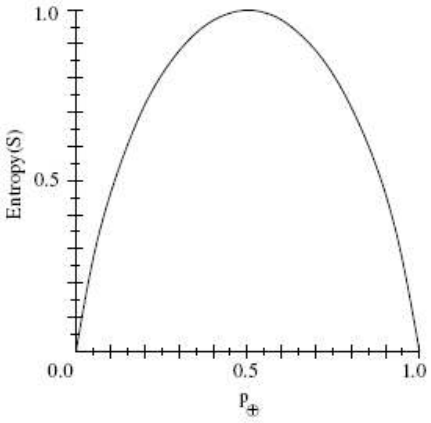

그림 1 Binary Entropy Function

표 2 엔트로피 값에 따른 변화

엔트로피	설명
Entropy == 0	모든 인스턴스가 하나의 클래스에 속하는 경우
Entropy == 1	모든 인스턴스가 각 클래스에 속할 동일한 확률
|Entropy| < 1	각 클래스에 속할 확률이 일정하지 않는 경우

즉 정보이득은 위에서 설명한 엔트로피 값이 전체에 미치는 여향을 계산한 값이며 아래와 같이 구할 수 있다.

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

즉, 엔트로피의 값이 작으면, 작을수록 정보이득은 크게 되고, 정보이득이 큰 속성을 초기에 선택하는 쪽이 좋은 결정 트리를 만들 수 있다[7].

```
ID3 (Examples, Target_attribute, Attributes)
  Examples: the training examples
  Target_attribute: the attribute whose value is to be predicated by the tree
  Attributes: a list of other attributes that may be tested by the learned
  decision tree
Create a Root node for the tree
If all Examples are positive, Return the single-node tree Root, with label = +
If all Examples are negative, Return the single-node tree Root, with label = -
If Attributes is empty, Return the single-node tree Root, with label = most
common value of Target_attribute in Examples
Otherwise Begin
  A <- the attribute from Attributes that best* classifies Examples
  The decision attribute for Root <- A
  For each possible value, vi, of A,
    Add a new tree branch below Root, corresponding to the test A = vi
    Let Examplesvi be the subset of Examples that have value vi for A
    If Examplesvi is empty
      Then below this new branch add a leaf node with label = most
      common value of Target_attribute in Examples
      Else below this new branch add the subtree
        ID3 (Examples, Target_attribute, Attributes - {A})
End
Return root
```

1. 종료조건
   ① 모든 예제가 옳은 경우
   ② 모든 예제가 그른 경우
   ③ 모든 속성이 트리에 표현된 경우
   ④ 모든 예제가 다 학습된 경우
2. 노드를 생성
3. 정보이득이 가장 큰 속성을 조건으로 선택
4. 해당 조건에 따른 자식노드들을 노드에 추가

① 각 노드에 재귀적으로 ID3함수를 호출

아래는 테니스를 치느냐 마느냐에 대한 결정 트리의 예제이다.

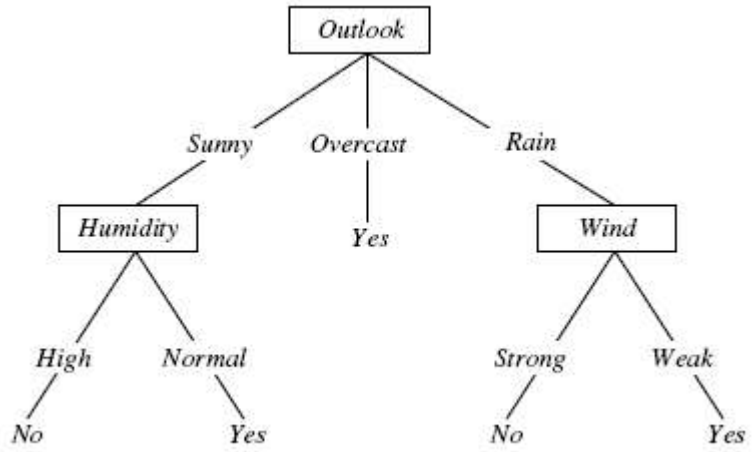

그림 2 Decision Tree, Play Tennis

```
IF (Outlook=Sunny) and (Humidity=High)
 THEN PlayTennis=No
 IF (Outlook=Sunny) and (Humidity=Normal)
 THEN PlayTennis=Yes
```

2.3.2.2 과적응과 가지치기

데이터에 잡음(Noise)가 있거나, 목적함수의 긍정적인 예제가 규칙을 표현할 수 있을 만큼 충분히 크지 않을 경우에 주어진 학습 예제들에 과적응(Overfitting)될 수 있는데, 대략 10~25%정도의 정확률이 떨어진다고 한다.

1. 가지치기 전략

① Pre-prunning: 트리 생성 도중에 가지치기를 수행.

② Post-prunning: 트리를 생성한 이후에, 가지치기를 수행.

단, 유의할 사항은 학습집합과 검증집합을 분리하여 실험하여야 한다는 점과, 통계적으로 유의미한 정보를 얻기 위해서는 검증 셋이 충분히 커야 한다는 것이다. 일반적으로 검증을 위하여 1/3정도, 학습을 위하여 2/3정도의 비율로 사용된다[7].

2.3.2.3 연속적인 값의 데이터 처리

일반적으로 결정트리의 경우 연속적인 데이터의 처리를 할 수 없는데, 이러한 연속적인 데이터의 경우는 적절한 구간을 정하거나, 별도의 분류를 하여 이산적인 값으로 변환하는 과정을 거쳐야만 한다. 이러한 과정에서 다시 정보이득 수치를 이용하여 그 임계치(Threshold)를 활용할 수도 있다.

2.3.2.4 다변량 값과 정보이득

정보이득을 통하여 트리 노드의 속성을 선택하는 경우, 다변량 값의 경우 선택되어지는 특성을 가지고 있다. 아주 극단적인 예를 들어보면, list-id 와 같은 값은 연속적인 값이면서 모든 변량의 값이 다른 경우 정보이득의 특성상 아주 이러한 값이 가장 중요한 속성으로 선택된다. 이러한 다변량의 값을 정규화 시켜주기 위해서 사용할 수 있는 접근방법이 GainRatio이다.

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)}$$

Gain값을 SplitInformation으로 정규화 한 값이 GainRatio이다.

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

다시, SplitInformation은 앞에서 언급된 엔트로피를 구하는 식과 유사한데, 엔트로피의 특성상 다양한 값을 가진 경우는 엔트로피의 값이 작아지지만, 0과 1사이의 값이 생성되므로, GainRatio를 구할 때에 다변량의 값의 경우 조정된다[21].

2.3.2.5 불완전한 값, 잡음 또는 불일치 데이터의 처리

다양한 경우가 발생할 수 있는데, 값이 빠져있는(Missing) 불완전한 값, 교정이나 제거가 필요한 잡음은 이상치(Outlier)의 값을 가지거나 오류(Error) 그리고 속성값들 사이에서 서로간에 다른 데이터를 가지는 불일치(Inconsistent)의 경우가 있다[7].

표 3 데이터 종류에 따른 변환

데이터 종류	권장방법
불완전한 값 (Missing)	Binning 가장 일반적인 값으로 대체 (평균값, 중앙값 등) Regression 다른 속성값들이 유사한 값으로 대체 Heuristic 전문가가 직접 채우기 Classification 별도의 “unknown” 값으로 분류
잡음 (Outlier, Error)	Clustering 교정 또는 제거(일정 아이템 이상의 그룹의 데이터만 선택)
불일치 (Inconsistent)	Heuristic 가장 최근 데이터를 사용

2.3.3 랜덤 포리스트 분류기

랜덤 포리스트는 많은 결정 트리들로 구성된 분류기이며, 개별 트리들의 결과 클래스들의 최빈값(Mode) 클래스를 출력으로 내보낸다. 추론 알고리즘은 Leo Breiman 과 Adele Cutler에 의해서 개발되었으며, Bell Labs의 Tin Kam Ho의 random decision forests가 시초이다[7][11].

2.3.3.1 학습 알고리즘

1. 학습할 경우의 수를 N, 분류기의 변수들의 수를 M
2. 입력변수 m은 트리의 노드에서 결정을 내릴 때에 사용되며, m은 M보다 작아야만 한다
3. 전체 N학습 데이터로부터 N번 랜덤 샘플링을 통하여 학습 데이터를 생성하는데, 한번 뽑은 데이터도 다시 뽑힐 수 있도록 복원추출을 하고, 나머지 예제들은 트리의 오류를 검증할 때에 사용된다.
4. 트리의 각 노드에 대하여, 랜덤으로 m개의 변수들을 선택하고 m개의 변수들에 기준하여 학습집합에서의 최고의 트리를 생성한다.
5. 개별 트리들은 가지치기를 하지 않고 끝까지 트리를 생성한다.

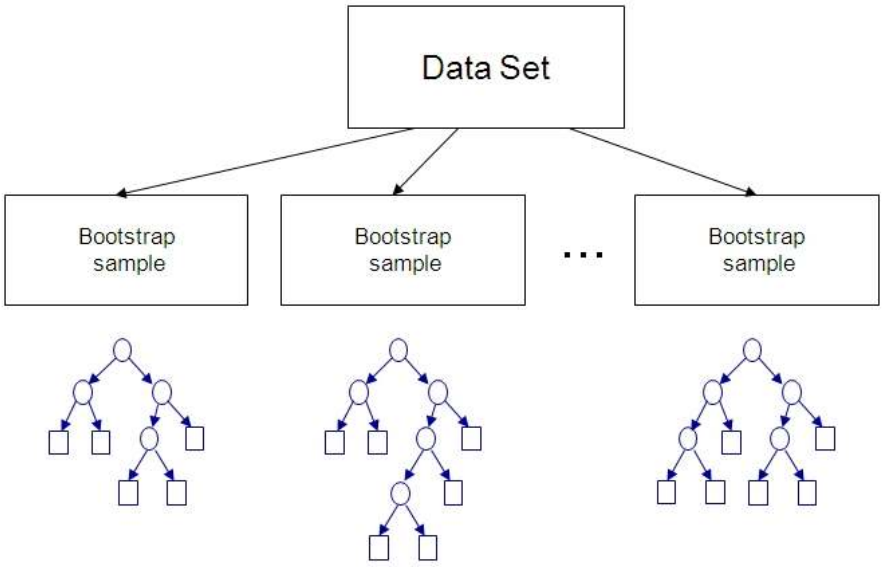

그림 3 Random Forests

2.3.3.2 알고리즘의 장점

1. 많은 데이터 집합에서 높은 정확률을 가지는 분류기이다
2. 입력변수가 아주 큰 경우에도 처리가 가능하다
3. 분류를 결정하는 변수들의 중요성도 예측할 수 있다
4. 불완전한 값들이 포함되어 있어도 좋은 성능을 보인다
5. 밸런스가 맞지 않는 데이터 집합의 경우에도 오류를 어느 정도 보정할 수 있다
6. 이러한 특징으로 인해 레이블링 되지 않은 데이터의 클러스터링, 이상치 인식 등에 사용될 수 있다
7. 학습속도도 상대적으로 빠르다.

2.4 실험도구

2.4.1 Weka

WEKA는 University of Waikato 에서 개발된 자바기반의 기계학습 도구로써GUI기반과 명령줄 기반의 인터페이스를 가지고 있다. 또한 오픈소스 데이터마이닝 도구로써 다양한 알고리즘 및 기법 등을 직접 추가 구현이 가능하며, 이미 구현된 수많은 알고리즘 등을 간단하게 사용해볼 수 있다[24].

그리고 본 논문에서 사용한 분류뿐만 아니라 군집화 및 연관규칙 등 다양한 데이터마이닝 기법을 활용할 수 있는 기능을 제공한다.

2.4.1.1 ARFF Format

Weka 는 별도의 입력데이터 포맷인 ARFF (Attribute Relation File Format)을 사용하는데 사용법은 아래와 같다[24].

표 4 ARFF Format Notation

Notation	Description	Examples
%	주석	% sentence boundary detection
@relation	데이터집합 설명	@relation sentence_boundary_detection
@attribute	속성유형	@attribute { enum, numeric, string }
@data	학습 데이터	

```
% The weather data
% This example was copied from Witten and Frank's data mining book
@relation weather

@attribute outlook { sunny, overcast, rainy }
@attribute temperature numeric
```

```
@attribute humidity numeric
@attribute windy { TRUE, FALSE }
@attribute play { yes, no }

@data
sunny, 85, 85, FALSE, no
sunny, 80, 90, TRUE, no
overcast, 83, 86, FALSE, yes
rainy, 68, 80, FALSE, yes
sunny, 69, 70, FALSE, yes
rainy, 75, 80, FALSE, yes
sunny, 75, 70, TRUE, yes
overcast, 72, 90, TRUE, yes
```

아래는 한국어 문장경계 인식을 위한 ARFF 예제이다. 사용된 속성은 총 11개이며, 결정트리의 특성상 수치형태로 표현해야 하므로, 문자열의 경우 해당 문자열이 등장할 확률로 변환하여 저장하였다.

```
@relation SentenceBoundary-weka.filters.supervised.instance.Resample-no-replacement

@attribute punct-type {마침표,느낌표,물음표,큰따옴표,작은따옴표}
@attribute prefix-pos {한글,마침표,물음표,줄임표,외국어,한자,숫자,기타}
@attribute suffix-pos { 한글,마침표,물음표,줄임표,외국어,한자,숫자,기타}
@attribute prefix-syllable-prob numeric
@attribute suffix-syllable-prob numeric
@attribute prefix-token-prob numeric
@attribute suffix-token-prob numeric
@attribute prefix-size numeric
@attribute suffix-size numeric
@attribute sentence-boundary? {yes,no}
@data
마침표,한글,한글,0.958653,0.844741,0.903459,0.777778,23,41,3,6,yes
마침표,한글,한글,0.958653,0.794043,0.888563,0.683558,10,32,6,3,yes
마침표,한글,한글,0.958653,0.826597,0.903459,0.918239,50,21,3,6,yes
마침표,한글,한글,0.958653,0.922689,0.903459,0.909185,12,23,3,6,yes
마침표,한글,한글,0.958653,0.922689,0.888563,0.815845,15,23,6,3,yes
```

3. 기계학습 기반 문장 경계 인식

3.1. 문장 경계 인식

3.1.1 코퍼스 정제 및 구조화

① 한국어의 경우 EUC-KR 에서 UTF-8 으로 변환

② 깨진 문자열 및 중복된 오분석 결과 제거

③ 문장, 어절, 토큰단위로 저장

④ 전체 학습 코퍼스를 구조화 하여 저장

3.1.2 문장경계 후보를 찾기 위한 통계정보 추출

① 코퍼스 내의 모든 문장경계로 인식된 위치의 구두점(특수문자 포함)의 빈도수

② 구두점 후보 앞/뒤의 토큰유형(본 논문에서 분류한 한국어에서 발생하는 6 가지 토큰유형)의 빈도수

3.1.3 문장경계를 판별하기 위한 통계정보 추출

① 문장경계 후보 위치에서 구두점의 유형에 따른 문장경계 유무 빈도수

② 문장경계 후보 앞/뒤 토큰에 대하여 토큰유형에 따른 문장경계 유무 빈도수

③ 문장경계 후보 앞/뒤 1 음절에 대한 문장경계 유무 빈도수

④ 문장경계 후보 앞/뒤 어절에 대한 문장경계 유무 빈도수

⑤ 문장경계 후보 앞/뒤 토큰에 대한 문장경계 유무 빈도수

3.2. 통계적 자질

3.2.1 문장경계 위치의 선택

모든 어절 즉, 공백으로 구분되는 위치를 문장경계 후보로 보았으며, 어절을 다시 토큰으로 구분하여 유형을 정의하였고, 한국어가 토큰의 유형이 다양하여, 한국어를 중심으로 표현 하였다.

3.2.1.1 문장경계 후보의 유형

① 공백, 한글, 마침표, 쉼표, 따옴표, 묶음표, 이음표, 드러냄표, 안드러냄표, 영문자, 한자, 기타기호, 숫자, 미지기호

② 모든 공백을 문장경계 후보로 인식.

나는 학교에 갑니다. 그리고 집으로 옵니다.
↑ ↑ ↑ ↑ ↑ ↑

3.2.1.2 문장경계후보 앞/뒤의 토큰명칭

① 한글, 영문자, 한자, 숫자

② 마침표 (온점, 물음표, 느낌표)

③ 쉼표 (반점, 가운뎃점, 쌍점, 빗금)

④ 따옴표 (큰 따옴표, 작은 따옴표)

⑤ 묶음표 (소괄호, 중괄호, 대괄호)

⑥ 이음표 (줄표, 붙임표, 물결표)

⑦ 드러냄표, 안드러냄표, 기타기호 등

당일 주가는 13.3% 떨어진 9.96 달러가 됐다.
한글/한글/숫자/마침표/숫자/기타기호/숫자/마침표/숫자…

3.2.1.3 문장경계 후보의 앞/뒤의 앞/뒤의 1음절

① 통계정보를 이용하여 발생할 확률을 이용

상품을 내놓는다. 내달부터는 가족간 **다 / 마침표 / 내**

3.2.1.4 문장경계 후보 앞/뒤의 토큰 문자열

① 통계정보를 이용하여 발생할 확률을 이용

잡지 않았으면 한다"며 "CNN 과 함께 대선 **한다/따옴표/며 며/따옴표/CNN**

3.2.1.5 문장경계 보 앞/뒤의 토큰의 길이

① 길이는 음절의 개수

57.7%로 지난 2003 년 12 월(104.4%) 이후 **2/./7 3/./1**

3.3. 시스템 아키텍처

3.3.1 문장경계 인식 시스템

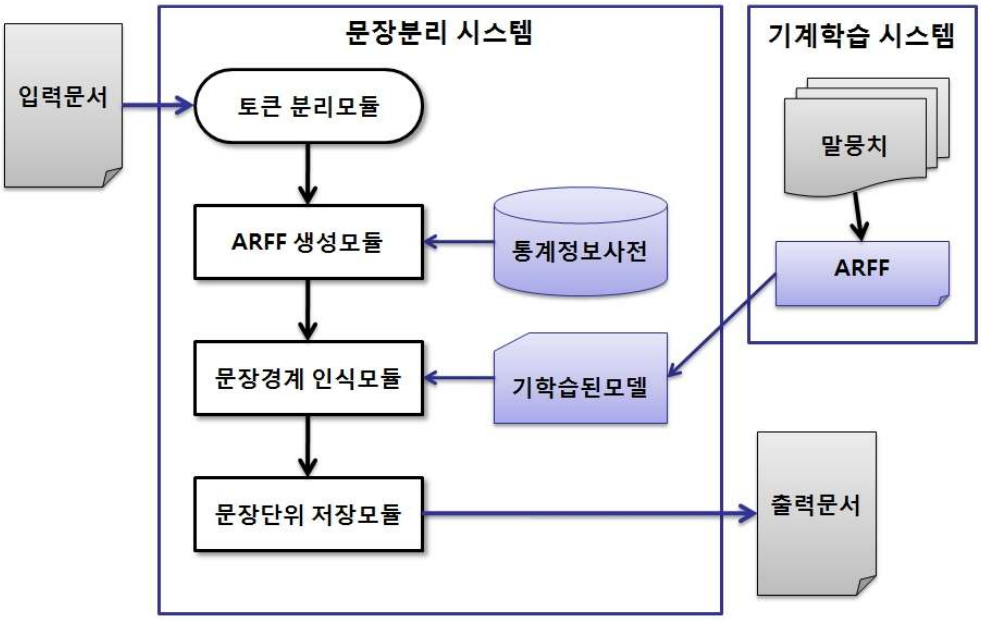

그림 4. 문장경계 인식 시스템 아키텍처

3.3.1.1 기계학습 과정

① 말뭉치를 통한 통계정보 추출 및 통계정보 사전 생성

② 추출된 통계정보를 통하여 학습을 위한 ARFF 생성.

③ 다양한 알고리즘을 통하여 실험 후, 가장 성능이 뛰어난 알고리즘을 기준으로 학습모델 생성

3.3.1.2 실험 및 검증

① 새로운 문서가 입력

② 입력문서의 모든 어절을 기준으로 나누고, 다시 어절을 토

큰단위로 구분

③ 통계정보 사전을 통하여 문장경계 후보위치의 자질정보 추출

④ 기 학습된 모델을 통하여 현재 위치의 문장경계 여부 판단

⑤ 토큰 및 어절을 병합하여 완결된 문장으로 저장 또는 출력

4. 실험 및 평가

4.1. 실험환경

본 논문에서는 한글의 경우 세종코퍼스 구문분석 결과 51,710 문장을 사용하였으며, 총 676,539 개의 어절을 통하여 학습을 수행하였다. 검증은 10-fold cross validation 방법을 사용하였다[28].

영문 코퍼스인 Wall Street Journal 의 경우는 총 110 만 어절을 사용하였고 동일한 방법으로 실험을 실시하였으며, 평가척도로는 정확률(Precision), 재현율(Recall) 및 F-measure 를 사용하였으며, 척도의 정의는 다음과 같다.

$$\mathrm{Pr}ecision = \frac{\text{시스템이추출한정답문장수}}{\text{시스템이추출한문장수}}$$

$$\mathrm{Re}call = \frac{\text{시스템이추출한정답문장수}}{\text{전체 문장수}}$$

$$F-measure = \frac{2 * \mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call}$$

4.2. 실험 및 평가

4.2.1 규칙에 의한 실험결과

앞에서 제시한 모든 규칙을 모두 적용한 것은 아니며, 실험에 의해서 정확률과 제현율의 밸런스를 고려하여 일부 규칙은 제거하거나 예외에 대한 예외규칙을 적용하기도 했다.

3.1.1.1 규칙기반의 문장경계인식 실험결과

표 5 규칙에 의한 문장경계 인식

Classifier	Precision	Recall	F-measure
Rule	97.5%	97.8%	97.7%

① 규칙기반의 성능이 꽤 좋았으나 규칙을 추가하거나 빼는 과정에서 규칙을 추가하면, 정확률이 올라가지만, 재현율은 떨어지고, 규칙을 빼면 그 반대의 현상이 나타났다.

② 규칙을 정확률과 재현율의 밸런스를 맞추면서 가장 성능이 뛰어난 실험결과를 기재하였다.

4.2.2 기계학습을 통한 실험결과

우선 1 차 실험으로써 충분히 작은 학습집합으로 다양한 기계학습 기법을 실험해 보고, 가장 성능이 뛰어난 4 개의 기계학습 기법으로 2 차 실험을 하였다.

4.2.2.1 샘플링 학습집합 학습 후 실험결과

표 6 샘플링 학습집합 실험 결과

Classifier	Precision	Recall	F-measure
BayesNet	96.2%	91.7%	93.9%
NaiveBayes	95.4%	87.8%	91.5%
Logistic	96.5%	97.7%	97.1%
Multilayer Perceptron	95.7%	97%	96.4%
kNN	95.4%	96.3%	95.8%
AdaBoost	93.4%	95.3%	94.4%
Dagging	95.2%	96.3%	95.7%
Decorate	96.7%	96.6%	96.6%
LogitBoost	94.7%	95%	94.9%
ADTree	94.2%	96.4%	95.3%
Bagging	96.6%	98.4%	97.5%
J48(C4.5)	**97% (2^{nd})**	96.9%	96.4%
Random Forest	**96.8% (3^{rd})**	98.1%	97.4%
LibSVM	**96.7% (4^{th})**	97%	96.6%
Maxent	**97.5% (1^{st})**	98%	97.8%

1. 결과분석

③ 정확률을 기준으로 하여 상위 4 개의 Classifier 선택하였으며, 문장경계 인식의 경우 재현율 보다는 정확률이 좀 더 의미 있는 지표라고 판단하였다.

4.2.2.2 전체 학습집합 학습 후 실험결과

표 7 전체 학습집합 실험 결과

Classifier	Precision	Recall	F-measure
LibSVM	98.4%	98.2%	98.3%
J48(C4.5)	98.7%	98.1%	98.4%
Random Forest	**98.7%**	**98.5%**	**98.6%**
Maxent	**98.7%**	**99.4%**	**99%**

1. 결과분석
   ① 1,000 개가 되지 않는 학습집합과 그의 100 배가 넘는 학습을 하여도 정확률 , 재현율의 향상은 적었다.
   ② 무엇보다도 베이스라인 정확률이 높았다.
   ③ 규칙기반의 문장경계 인식에서 발생하였던, 시소현상이 해소되면서, 정확률과 재현율이 동시에 오를 수 있었다.

4.2.3 기계학습 기법 선택

1. 성능의 편차는 그다지 크지 않았으나 Maximum Entropy 기법이 정확률 재현율 모두 가장 낮은 알고리즘으로 판단 되었다.

4.2.4 Wall Street Journal 코퍼스에 대한 실험

마찬가지로 검증은 10-Fold Cross Validation 으로 하였으며, 동일한 환경에서 실험 및 검증을 수행하였다

1. 실험결과

표 8 Wall Street Journal 코퍼스에 대한 실험결과

Classifier	Precision	Recall	F-measure
J48(C4.5)	97.8%	96.8%	97.3%
Random Forest	**98.2%**	**97.8%**	**98%**
Maxent	**98.6%**	**97.4%**	**98%**

2. 결과분석

① 영문코퍼스에서는 두 기법에서 거의 동일한 실험결과를 확인할 수 있었다.

② 동일한 자질에 대한 정보를 추출하였으며, 별도의 Heuristic 을 추가하지 않고도 높은 성능을 나타냄으로써 해당 자질이 범용적임을 확인할 수 있었다.

4.2.5 자질에 대한 성능실험

각 자질을 제거하고 실험하였을 경우에 해당 자질이 얼마나 전체 정확률을 높이는 데에 기여하는 지를 테스트 해보았다. 전체 실험집합에서 테스트는 하지 못하였으며, 샘플링 한 결과에서 해당 자질에 대한 상대적인 비교를 수행하였다.

표 9 개별속성의 자질에 대한 성능실험 결과

Removed Feature	Precision	Recall	F-measure
Complete Set (All Features)	**97.5%**	**98.2%**	**97.8%**
Base Set (Punctuation)	92% **(-5.5%)**	93.6% **(4.6%)**	92.8% **(-5%)**
Prefix-token-type	97.4% (-0.1%)	98% (-0.2%)	97.7% (-0.1%)
Suffix-token-type	97.4% (-0.1%)	98.2%	97.8%
Prefix-syllable	97.5%	98.2%	97.8%
Suffix-syllable	97.6% **(+0.1%)**	97.9% (-0.3%)	97.8%
Prefix-token	**97.8**% **(+0.3%)**	97.9% (-0.3%)	97.8%
Suffix-token	97.3% (-0.2%)	97.6% (-0.6%)	97.4% (-0.4%)
Prefix-token-length	96.4% (-1.1%)	98.4% **(+0.2%)**	97.4% (-0.4%)
Suffix-token-length	97.3% (-0.2%)	98.3% **(+0.1%)**	97.8%

1. 선택된 자질을 제거하고 실험한 결과, 앞/뒤 토큰의 길이 토큰의 유형, 구두점 순으로 긍정적인 영향을 보였다.
2. 반면에 뒤 첫 음절 앞의 토큰 등의 자질은 오히려 정확률을 떨어뜨리는 현상을 보였다

4.2.6.1 기여도가 떨어지는 자질을 제거하고 실험

표 10 기여도가 떨어지는 자질 제거 후 실험결과

Features	Precision	Recall	F-measure
12 features	**98%**	98.9%	**98.4%**
10 features	98.5%	**98.1%**	**98.4%**

1. 위에서 제시된 정확률을 떨어뜨리는 자질을 제거하고 실험한 결과 확실히 정확률은 올라갔으나, F-measure 는 동일하게 나타났으며, 전체적으로 해당 자질을 제거하여 정확률은 향상시킬 수 있었으나, 재현율은 떨어졌다.

4.2.6 학습데이터 크기에 따른 성능 변환

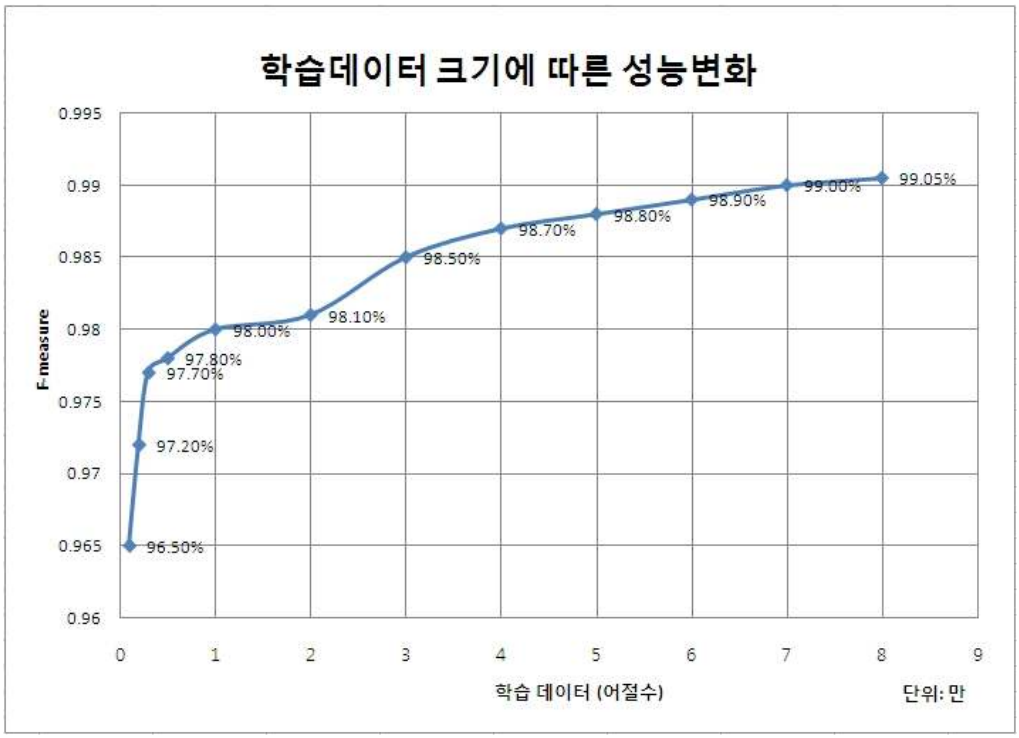

그림 5 학습데이터 크기에 따른 성능변화

1. 학습 데이터가 10,000 개 어절 정도에 거의 98% 이상의 성능을 보이며, 그 이상의 데이터에서 선형적으로 증가하였다
2. 그래프 상으로는 잘 보이지 않지만, 일부 성능이 튀는 현상도 보였다

5. 향후 연구 계획 및 결론

5.1. 향후 연구 계획

본 논문에서는 모든 어절을 문장경계 후보를 결정하고 문장경계의 정의를 안은문장 등을 하나의 문장으로 보아 문장경계 인식을 시도하였다. 하지만, 하나의 인용문이 길어질 수도 있고, 그에 따른 문제점도 발생할 수 있겠다. 또한 웹 문서와 같이 구두점 자체가 없거나 정형화 되지 않은 문서의 경우에는 본 논문에서와 같이 말뭉치를 통한 학습 모델로는 분석하기 어려울 것으로 보인다.

웹 문서 또는 일반 문서 등의 경우에도 현재 자질들이 얼마나 잘 동작하는 지에 대해서도 실험해보고, 그러한 문제들도 해결할 수 있는 새로운 자질에 대한 것들도 연구해 보고자 한다.

이 이외에도 자연어 처리 및 형태소 분석기 등의 모듈에서 문장경계 인식을 통하게 되었을 때에 얼마나 성능이 개선될 수 있는지에 대한 실험도 의미 있는 연구분야라 보여진다.

5.2. 결론

본 논문에서는 다양한 기계학습 기법을 사용하여 문장경계 인식방법을 실험 하였으며, 그 중에서 Decision Tree, SVM, Random Forest 및 Maximum Entropy 기계학습 기법이 가장 나은 성능을 보였다. 그리고 기존의 한국어 문장경계 인식의 논문에서 사용했던 특정 구두점 정보에 의존하지 않고서도 보다 나은 성능을 보이며, 구두점의 발생횟수에 따른 편차 등의 현상이 없어졌다. 또한 기존의 논문들에서 문

장경계의 후보위치로 삼았던 특정 문장부호가 있는 위치 뿐만 아니라 모든 어절의 위치를 후보로 삼았다는 점도 의미가 있다고 보여진다.

이러한 한글에서 사용되었던 자질들을 영문 코퍼스에도 동일하게 실험한 결과 높은 정확률을 보임으로써 선택한 자질이 특정 언어에 종속적이지 않으며, 보다 범용적인 것임을 확인할 수 있었다.

끝으로 다양한 기계학습 기법을 적용한 실험을 통하여 성능을 향상시킬 수 있었다는 점도 의미를 둘 수 있겠다.

5.3. 참고문헌

[1] Andrei Mikheev, Tagging sentence boundaries, 2000

[2] D. Hillard, M. Ostendorf, A.Stolcke, Y. Liu, E. Shriberg, Improving Automatic Sentence Boundary Detection with Confusion Networks, 2004

[3] Daniel J. Walker, David E. Clements, Maki Darwin and Jan W. Amtrup, Sentence Boundary Detection A Comparison of Paradigms for Improving MT Quality, 2002

[4] David D. Palmer, Marti A. Hearst, Adaptive Sentence Boundary Disambiguation, 1994

[5] David D. Palmer, Marti A. Hearst, Adaptive Multilingual Sentence Boundary Disambiguation, 1997

[6] Heui-Seok Lim, Kun-Heui Han, Korean Sentence Boundary Detection Using Memory-based Machine Learning, 2004

[7] Ian H. Witten, Eibe Frank, Data Mining Practical Machine Learning Tool and Techniques, 2005

[8] Jeffrey C. Reynar and Adwait Ratnaparkhi, A Maximum Entropy Approach to Identifying Sentence Boundaries, 1997

[9] Jiawei Han, Data Mining: Concepts and Techniques Second Edition

[10] Kazuya Shitaoka, Kiyotaka Uchimoto, Tatsuya Kawahara, Hitoshi Isahara, Dependency Structure Analysis and Sentence Boundary Detection in Spontaneous Japanese, 2004

[11] Leo Breiman, Adele Cutler, http://www.stat.berkeley.edu/~breiman

[12] Mark Stevenson and Robert Gaizauskas, Experiments on Sentence Boundary Detection, 2002

[13] Michael D. Riley, Some applications of tree-based modeling to speech and language, 1989

[14] Naver Encyclopedia – http://100.naver.com

[15] Neha Agarwal, Kelley Herndon Ford, Max Shneider, Sentence Boundary Detection Using a MaxEnt Classifier, 2005

[16] Nilani Aluthgedara, Recognizing Sentence Boundaries and Boilerplate, 2003

[17] Shimei Pan and James C. Shaw, Instance-based sentence boundary determination by optimization for natural language generation, 2005

[18] Sung Dong Kim, ByoungoTak Zhang, Yung Taek Kim, Reducing parsing complexity by intra-sentence segmentation based on maximum entropy model, 2000

[19] Tibor Kiss, Jan Strunk, Viewing sentence boundary detection as collocation identification, 2002

[20] Tibor Kiss, Jan Strunk, Unsupervised Multilingual Sentence Boundary Detection 2006

[21] Tom M. Mitchell, Machine Learning Textbook, McGraw-Hill, 1997

[22] Yang Liu, Andreas Stolcke, Elizabeth Shriberg, Mary Harper, Using Conditional Random Fields For Sentence Boundary Detection In Speech, 2005

[23] Young-Ae Seo, Yoon-Hyung Roh, Ki-Young Lee, Sang-Kyu Park, CaptionEye EK English-to-Korean Caption Translation System Using the Sentence Pattern, 2000

[24] WEKA-Machine Learning Software - http://www.cs.waikato.ac.nz/ml

[25] Wikipedia – The Free Encyclopedia - http://wikipedia.org/

[26] Yahoo! Search Blog - http://www.ysearchblog.com/archives/000172.html

[27] Pandia Search Engine News – http://www.pandia.com/seq/383-web-size.html

[28] 21 세기 세종계획 - http://www.sejong.or.kr/

[29] 문교부 고시 한글맞춤법 – 부록 문장부호

감사의 글

본 논문이 있기 까지는 항상 따뜻하고 인자하신 지도로써 부족한 저를 올바른 길로 이끌어 주신 임해창 교수님께 진심으로 감사 드립니다. 부족한 논문을 검토해 주시고 많은 조언을 해주신 육동석 교수님, 이성환 교수님께도 감사를 드리며, 마지막 까지 실험과 논문의 내용에 대해 도움 주셨던, 고려대학교 자연어처리 연구실의 이주영 선배님께 감사를 드립니다.

처음 대학원에 진학을 결정하고 임해창 교수님을 찾아 뵈었을 때에 정보검색에 대해서는 아무것도 모른 채 가르침을 주십사 무턱대고 교수님을 찾아온 저에게 격려를 해주시던 모습이 아직도 눈에 선한 것 같습니다. 그 때의 교수님의 충고가 없었더라면 지금의 저도 없었을 것이란 생각에 다시 한번 교수님께 감사를 드립니다.

그리고 기존 논문에 관한 조언을 해주신 임희석 박사님과 데이터 마이닝 수업을 통해서 많은 가르침을 주셨던 강재우 교수님께도 감사드립니다.

대학원의 길로 들어설 수 있도록 조언과 도움을 준 한 학기 선배이기도 한 희원이에게 고맙다는 말을 전하고 싶고, 검색엔진 뿐만 아니라 인생의 선배로써 조언을 많이 해주셨던 영길이형, 상호형 그리고 순간 순간 힘들때 마다 저에게 많은 힘이 되어주셨던 많은 검개그 식구들(장형석, 김정은, 김형준, 이왕재)께 고마움을 전합니다.

처음 대학원의 진학을 결정하고 많은 고민을 하고 있을 때에 “스스로의 발전을 위해서라면 열심히 노력해야지”하시면서 배려를 해주셨던, 지금은 예전회사가 된 퓨쳐윈 신동범 사장님 그리고 물심양면으로 저를 도와주셨던 변회균 이사님, KERIS 프로젝트 중에 학교수업 때문에 일찍 퇴근했지만, 늘 열심히 하라고 격려해주셨던 박기환 차장님, 이종선 부장님 그리고 학업을 계속 할 수 있도록 기회와 배려를 해주신 오픈마루 김범준 실장님과 늘 옆에서 조언과 격려를 아끼지 않으셨던 윤종완 부장님 그리고 오픈마루 검색팀 모든 분들께 감사를 드립니다.

무엇보다도 2 년 6 개월이라는 긴 기간을 한마디 불평도 없이 지켜봐 주고 격려해 준 사랑스런 아내 영미와 내 인생의 두 가지 보물 첫째 소원이, 둘째 시훈이와 이 기쁨을 같이 나누고 싶습니다.