기계학습 기법을 이용한 문장경계인식

박수혁*, 임해창**
*고려대학교 컴퓨터 정보통신대학원
**고려대학교 컴퓨터학과
e-mail : *park.suhyuk@gmail.com, **rim@nlp.korea.ac.kr

Sentence Boundary Detection Using Machine Learning Techniques

Su-Hyuk Park*, Hae-Chang Rim**
* Graduate School of Computer and Information Technology, Korea University
** Dept. of Computer Science and Engineering, Korea University

요 약

본 논문은 언어의 통계적 특징을 이용하여 범용의 문장경계 인식기를 제안한다. 제안하는 방법은 대량의 코퍼스 내에서 사용되고 있는 문장 경계를 기준으로 음절 및 어절 등의 자질을 이용하여 통계적 특징을 추출하고 다양한 기계학습 기법을 사용하여 문장경계를 인식하고자 하였다. 또한 특정 언어나 도메인에 제한적이지 않고 범용적인 자질만을 사용하려고 노력하였다.

언어의 특성상 문장의 구분이 애매한 경우 또는 잘못 사용 된 구두점 등의 경우에도 적용 가능하도록 다양한 자질을 사용하여 실험하였으며, 한국어와 영문 코퍼스에 대해서 동일한 자질을 적용하여 실험하여 본 논문에서 제시한 자질들이 한국어 및 다른 언어권의 언어에도 적용될 수 있는 범용적인 자질임을 확인할 수 있었다.

한국어 문장경계 인식을 위한 기계학습 및 실험을 위해서 세종계획 코퍼스를 사용하였으며, 성능척도로는 정확률과 재현율을 사용하였으며, 실험결과 제안한 방법으로 99%의 정확률과 99.2%의 재현율을 보였다. 영문의 경우는 Wall Street Journal 코퍼스를 사용하였으며, 동일한 자질을 적용하여 실험한 결과 98.9%의 정확률과 94.6%의 재현율을 보였다.

1. 서론

‘문장’의 사전적인 의미는 '의사를 전달하는 최소의 단위'로 정의되어 있으며, 전통 문법에서는 '비교적 완전하고 독립된 의사전달 단위다'라고 정의하고 있으며, 이러한 문장의 단위가 대부분의 자연어 처리 도구, 품사부착기 등의 기본단위가 되고 있으며, 음성인식된 문장 또는 OCR 처리된 문서 등의 문장경계가 모호한 경우에 대해서도 전처리 과정으로도 문장경계의 인식작업이 반드시 요구된다.

하지만 실례에서는 문장경계의 구분이 명확하지 못한 경우가 많이 보이며, 사람이 보아서도 제대로 구분되지 않는 경우도 많다. 일반적으로 마침표, 물음표, 느낌표 등의 기호를 기준으로 문장경계가 구분된다고 볼 수 있으나, "문장의 시작번호, e-mail, 영어의 약어, 강조를 위한 반복, 열거 등에 사용되는 마침표 또는 잘못 사용된 구두점의 경우와 같이 그렇지 않은 경우도 많다.

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 대량의 코퍼스에서 추출한 문장경계의 후보로 삼을 수 있는 구두점을 추출하고 기계학습 기법을 사용하여 문장경계로 인식되는 위치를 표현할 수 있는 다양한 자질을 학습하고, 향후 새로운 문서에 대하여 문장경계를 판단해낼 수 있는 문장경계 인식기를 제안한다.

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 하였다.

기계학습에 사용된 도구로는 WEKA Workbench 라는 데이터마이닝 툴킷을 사용하였으며, 제공되는 다양한 분류 알고리즘으로 실험한 결과 알고리즘에 의해서 정확률과 재현율이 크게 차이 나는 경우는 없었으나, 성능 면에서 Random Forest 알고리즘이 가장 좋은 성능을 보였고, 학습에 사용된 코퍼스는 원시 코퍼스만을 사용하였으며, 어절기준으로 구분하고 다시 어절에서 구두점 후보가 발생하면, 구두점을 포함하여 별도의 토큰으로 분리해 내어 분석을 시도하였다.

2. 관련연구

Riley, Michael D. (1989)는 구두점이 발생하는 주변 단어의 출현확률 및 구두점이 발견된 어절의 클래스 등의 자질을 추출하였고, 확률정보는 AP News 2 천 5 백만 단어를 통해 구하였으며, Brown 코퍼스 에서 Decision Tree (C4.5)를 이용한 결과 99.8%의 정확률을 보였다.

David D. Palmer, Marti A. Hearst, (1994)는 구두점 주변의 단어에 대한 품사의 확률정보를 이용하여 20 가지 정도의 토큰으로 구분하였고, Feed-forward Neural Network 를 이용한 결과 98.5%의 정확률을 보였다.

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.

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

이와 같이 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.

하지만 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.

3. 문장 경계인식

가. 코퍼스 정제 및 구조화
  1) 한국어의 경우 문자열 인코딩을 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

나. 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

다. 문장경계를 판별하기 위한 통계정보 추출
  1) 모든 문장경계 후보 위치에서 구두점의 유형에 따른 문장경계 유무 빈도수
  2) 모든 문장경계 후보 앞/뒤 토큰에 대하여 토큰유형에 따른 문장경계 유무 빈도수
  3) 모든 문장경계 후보 앞/뒤 1 음절에 대한 문장경계 유무 빈도수
  4) 모든 문장경계 후보 앞/뒤 어절에 대한 문장경계 유무 빈도수
  5) 모든 문장경계 후보 앞/뒤 토큰에 대한 문장경계 유무 빈도수

4. 통계적 자질

가. 문장경계 위치의 선택

모든 문장경계에서 발생한 음절 또는 문자에 대한 통계정보를 통하여, 가장 많이 발생한 상위 99.9%의 음절 또는 문자가 출현하는 위치를 문장경계 후보로 보았으며, 한국어의 경우 총 5 가지의 구두점(마침표, 물음표, 느낌표, 큰/작은따옴표)이 문장경계 후보로 볼 수 있었다. 영어의 경우 총 3 가지의 구두점(마침표, 물음표, 큰 따옴표)이 후보로 나타났다.

한국어의 경우가 문장경계 후보종류가 더 많으므로 한국어를 중심으로 표현 하였다.

나. 문장경계 후보를 위한 구두점
  1) 마침표, 물음표, 느낌표, 큰/작은따옴표
  2) 후보가 발생한 경우 문장경계 후보로 인식.

나는 학교에 갑니다. 그리고 집으로 옵니다.
↑ ↑

다. 구두점후보 앞/뒤의 토큰유형
  1) 한글/ 외국어/ 숫자/ 특수문자
  2) 유형 1 (마침표, 물음표, 느낌표)
  3) 유형 2 (쉼표, 콜론, 빗금)
  4) 유형 3 (따옴표, 괄호, 줄임표, 붙임표)
  5) 유형 4 (기타 특수문자)
  6) 유형 5 (유형에 포함되지 않는 문자열)

당일 주가는 13.3% 떨어진 9.96 달러가 됐다.
숫자/./기호 4 숫자/./숫자 한글/.

라. 구두점후보의 앞/뒤의 앞/뒤의 1 음절
  1) 통계정보를 이용하여 발생할 확률을 이용

상품을 내놓는다. 내달부터는 가족간
다/./내

마. 구두점후보 앞/뒤의 토큰 문자열
  1) 통계정보를 이용하여 발생할 확률을 이용

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

바. 구두점후보 앞/뒤의 토큰의 길이
  1) 길이는 음절의 개수

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

사. 구두점후보 이전/다음 구두점 후보까지의 거리
  1) 거리는 사이에 존재하는 토큰의 수

0.6 -2.5 와트의 TDP 규격을 따르며 1.8GHz 까지
1/./3 3/./6 6/./3

5. 기계학습 기법 선택

가. 다양한 기계학습 기법으로 실험
  1) 충분히 작은 학습집합 샘플링

해당 코퍼스에 가장 성능이 뛰어난 기계학습 기법 선택을 위함이므로, 결과에 왜곡이 발생하지 않을 정도의 작은 집합을 샘플링 하였다.

  2) 추출된 학습집합을 통하여 다양한 알고리

즘에 대하여 실험을 수행

3) 실험 대상 알고리즘 중에서 상위 2 개의 알고리즘을 선택

4) 충분히 큰 학습집합에 대하여 다시 실험

선택한 알고리즘이 전체 학습집합에서도 잘 동작하는 지를 위한 실험을 다시 수행하였다.

나. 1 차 실험결과

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

1) 샘플링 학습집합 학습 후 실험결과

Classifier	Precision	Recall	f-measure
BayesNet	0.962	0.917	0.939
NaiveBayes	0.954	0.878	0.915
Logistic	0.965	0.977	0.971
Multilayer Perceptron	0.957	0.97	0.964
kNN	0.954	0.963	0.958
AdaBoost	0.934	0.953	0.944
Bagging	0.966	0.984	0.975
Dagging	0.952	0.963	0.957
Decorate	0.967	0.966	0.966
LogitBoost	0.947	0.95	0.949
ADTree	0.942	0.964	0.953
J48(C4.5)	**0.97 (1st)**	0.969	0.964
Random Forest	**0.968 (2nd)**	0.981	0.974
REPTree	0.96	0.983	0.971

2) 결과분석

① 총 885 개의 어절 (0.1%)을 학습

② 정확률기준 상위 2 개의 Classifier 선택하였으며, 문장경계 인식의 경우 재현율보다는 정확률이 좀 더 의미 있는 지표라고 판단하였다.

다. 2 차 실험결과

1) 30% 샘플링 학습집합 학습 후 실험결과

Classifier	Precision	Recall	f-measure
Decision Tree C4.5	0.979	0.98	0.98
Random Forest	**0.983**	**0.986**	**0.985**

2) 70% 샘플링 학습집합 학습 후 실험결과

Classifier	Precision	Recall	f-measure
Decision Tree C4.5	0.98	0.989	0.984
Random Forest	**0.991**	**0.992**	**0.991**

3) 결과분석

① 265,529 어절(30%), 619,567 어절(70%)

② 1,000 개가 되지 않는 학습집합과 그의 1,000 배가 넘는 학습집합과 정확률 , 재현율 모두 3%정도 밖에 향상이 없었다.

라. 기계학습 기법 선택

1) 시스템 성능상 Random Forest 알고리즘에서 모든 학습집합을 학습 시킬 수는 없었으며, 70%정도 (약 60 만 어절)의 학습을 통하여 가장 낮은 알고리즘으로 판단됨

6. 실험결과

본 논문에서는 한글의 경우 세종코퍼스 100 만 어절 중 70 만 어절을 사용하여 99.1%의 정확률을 얻어낼 수 있었으며, Spoken 과 Written 을 비율을 동일하게 하여 실험을 실시하였으며, 학습 및 실험은 10-fold cross validation 방법을 사용하였다.

영문 코퍼스인 Wall Street Journal 의 경우는 총 110 만 어절 모두를 사용하였고 동일한 방법으로 실험을 실시하였다.

평가척도로는 문장 정확률(Precision), 문장 재현율(Recall) 및 F-measure 를 사용하였으며, 척도의 정의는 다음과 같다.

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

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

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

가. 세종계획 코퍼스에 대한 실험

Spoken 과 Written 모두 학습에 포함시켰으며, 알고리즘은 가장 성능이 뛰어난 Decision Tree 와 Random Forest 알고리즘을 선택하여 실험하였다

1) 실험결과

Classifier	Precision	Recall	f-measure
Decision Tree	0.98	0.989	0.984
Random Forest	**0.991**	**0.992**	**0.991**

① Spoken 코퍼스와 Written 코퍼스에 대하여 90%정도를 Random Sampling 하여 학습하고 나머지 10%를 이용하여 실험한 결과 가장 성능이 좋았다.

② Classifier 의 경우 Random Forest 가 더 나은 성능을 보여주었다.

나. Wall Street Journal 코퍼스에 대한 실험

마찬가지로 검증은 10-Fold Cross Validation 으로 하였으며, 한국어와는 달리 자질의 수 및 정보의 크기가 달라서 모든 학습집합을 다 사용할 수 있었다

1) 실험결과

Classifier	Precision	Recall	f-measure
Decision Tree	**0.989**	**0.946**	**0.967**
Random Forest	0.981	0.942	0.961

2) 결과분석

① 아주 큰 차이는 아니지만, 한국어와는 달리 Decision Tree 가 조금 더 좋은 성능을 보여주었다.

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

다. 자질에 대한 성능실험

각 자질을 제거하고 실험하였을 경우에 해당 자질이 얼마나 전체 정확률을 높이는 데에 기

여하는 지를 테스트 해보았다. 전체 실험집합에서 테스트는 하지 못하였으며, 샘플링 한 결과에서 해당 자질에 대한 상대적인 비교를 수행하였다.

Removed Feature	Precision	Recall	f-measure
Complete Set (All Features)	**0.975**	**0.982**	**0.978**
Base Set (Punctuation)	0.92 **(-5.5%)**	0.936 **(4.6%)**	0.928 **(-5%)**
Prefix-token-type	0.974 (-0.1%)	0.98 (-0.2%)	0.977 (-0.1%)
Suffix-token-type	0.974 (-0.1%)	0.982	0.978
Prefix-syllable	0.975	0.982	0.978
Suffix-syllable	0.976 **(+0.1%)**	0.979 (-0.3%)	0.978
Prefix-token	**0.978 (+0.3%)**	0.979 (-0.3%)	0.978
Suffix-token	0.973 (-0.2%)	0.976 (-0.6%)	0.974 (-0.4%)
Prefix-token-length	0.964 (-1.1%)	0.984 **(+0.2%)**	0.974 (-0.4%)
Suffix-token-length	0.973 (-0.2%)	0.983 **(+0.1%)**	0.978
Prefix-token-distance	0.975	0.983 **(+0.1%)**	0.979 **(+0.1%)**
Suffix-token-distance	0.975	0.979 (-0.3%)	0.977 (-0.1%)

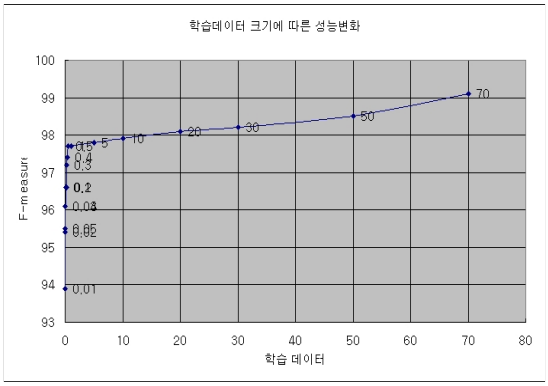
1) 선택된 자질을 제거하고 실험한 결과, 앞/뒤 토큰의 길이 토큰의 유형, 구두점 순으로 긍정적인 영향을 보였다.
2) 반면에 뒤 첫 음절 앞의 토큰 등의 자질은 오히려 정확률을 떨어뜨리는 현상을 보였다

라. 기여도가 떨어지는 자질을 제거하고 실험

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.

마. 학습데이터 크기에 따른 성능 변환



1) 학습 데이터가 1,000 건 정도에 거의 97% 이상의 성능을 보이며, 그 이상의 데이터에서 선형적으로 증가하였다
2) 그래프 상으로는 잘 보이지 않지만, 일부 성능이 튀는 현상도 보였다

7. 결론

본 논문에서는 다양한 기계학습 기법을 사용하여 문장경계 인식방법을 제안하였으며, Decision Tree 와 Random Forest 기계학습 기법을 통하여 실험 및 비교를 하였다. 또한 언어에 종속적이지 않은 독립적인 자질만을 사용하려고 노력하였으며, 기존의 한국어 문장경계 인식의 논문에서 사용했던 특정 구두점 정보에 의존하지 않고서도 보다 나은 성능을 보이고 있으며, 구두점의 발생횟수에 따른 편차 등의 현상이 없고, Spoken Language 와 Written Language 에 따른 정확률이 떨어지는 문제도 많이 보정되었다.

영문 코퍼스에도 동일하게 적용하여 실험한 결과 높은 정확률을 보임으로써 선택한 자질이 보다 범용적인 것임을 확인할 수 있었고 다양한 알고리즘을 통한 실험을 통하여 성능을 향상시킬 수 있었다.

향후 연구로는 정형화 되어있지 않은 문서 즉, 웹문서나 구두점이 생략된 문서 등의 경우에도 얼마나 잘 동작하는 지에 대해서도 해결할 수 있는 새로운 자질에 대한 것들도 고려해보고자 한다.

참고문헌

[1] 1989, Some applications of tree-based modeling to speech and language
[2] 1994, Adaptive Sentence Boundary Disambiguation - Palmer, Hearst
[3] 1997, A Maximum Entropy Approach to Identifying Sentence Boundaries - Reynar, Ratnaparkhi
[4] 1997, Adaptive Multilingual Sentence Boundary Disambiguation - Palmer, Hearst
[5] 2000, Reducing parsing complexity by intra-sentence segmentation based on maximum entropy model
[6] 2000, Tagging sentence boundaries
[7] 2004, Dependency structure analysis and sentence boundary detection in spontaneous Japanese
[8] 2004, Korean Sentence Boundary Detection Using Memory-based Machine Learning - Lim, Han
[9] 2005, Instance-based sentence boundary determination by optimization for natural language generation
[10] 2006, Unsupervised Multilingual Sentence Boundary Detection