

IG back-off 평탄화와 확률 기반 모델을 이용한 한국어 및 영어 단위화

이은지^o 이근배
포항공과대학교 컴퓨터공학과
{juicy, gblee}@postech.ac.kr

Korean and English Text Chunking Using IG Back-off Smoothing and Probabilistic Model

Eunji Yi^o Geunbae Lee
Dept. of Computer Science & Engineering, Pohang University of Science & Technology

요 약

많은 자연언어처리 분야에서 문장의 단위화는 기본적인 처리 단계로서 중요한 위치를 차지하고 있다. 한국어 단위화에 대한 기존 연구들은 규칙 기반 방법이나 기계 학습 기법을 이용한 것이 대부분이었다. 본 논문에서는 통계 기반 방식의 일환으로 순수 확률 기반 모델을 이용한 단위화 방법을 제시한다. 확률 기반 모델은 처리하고자 하는 해당 언어에 대한 깊은 지식 없이도 적용 가능하다는 장점을 가지므로 다양한 언어의 단위화에 대한 기본 모델로서 이용될 수 있다. 또한 자료 부족 문제를 해결하기 위해 메모리 기반 학습 시에 사용하는 IG back-off 평탄화 방식을 시스템에 적용하였다. 본 논문의 모델을 적용한 단위화 시스템을 이용하여 한국어와 영어에 대해 실험한 결과 비교적 작은 규모의 말뭉치를 학습하였음에도 불구하고 각각 90.0%, 90.0%의 정확도를 보였다.

1. 서론

텍스트 단위화(chunking)은 문법적으로 서로 관련된 단어 집합으로 텍스트를 나누는 것을 말한다. 예를 들어, 다음의 예문을 살펴보자.

- 작은 세 마리의 공이 마당에서 놀다가 잠든 듯하다.

이 문장은 다음과 같이 나누어 질 수 있다.

- [NP 작은 세 마리의 공이] [NP 마당에서] [VP 놀다가] [VP 잠든 듯하다].

Abney는 [1]에서 이러한 텍스트 단위화 과정을 구문분석 단계의 전 단계로 이용하는 것이 유용할 수 있음을 제시하였다. 이는 텍스트 단위화가 보다 높은 단계의 텍스트 분석을 위한 기초를 제공하며 동시에 보다 높고 복잡한 수준의 결합 문제 등에 대해서는 나중에 결정할 수 있도록 해주기 때문이다.

텍스트 단위화와 관련된 연구들은 크게 규칙 기반 방법과 통계 기반 방법으로 나눌 수 있다.

규칙 기반 방법은 주어진 텍스트 자료를 유한 상태 오토마타나 그 밖의 제약 규칙들과 함께 이용하는 방식이다. Kupiec은 [2]에서 제시한 영어와 불어의 명사구 인식기에 유한 상태 transducer를 이용하였다. Ramshaw와 Marcus는 [3]에서 태깅된 말뭉치에

대한 단위화를 위해 transformation-based learning을 사용하여 얻어진 규칙들을 이용하였다.

통계 기반 방법은 단어와 품사 태그 등의 정보로부터 확률 모델을 구축하고 이를 단위화에 이용하는 것이다. Skut과 Brants는 [4]에서 제한된 길이의 문법 구조 인식에 HMM 기반의 접근 방식을 제안하였다. 또한 [5]에서는 memory-based learning method를 이용한 단위화 방법이 제안되었다. Ratnaparkhi는 [6]에서 태깅 과정에서의 단위화를 위해 maximum entropy를 이용하는 방법을 제시하였다.

신호필은 한국어에 대한 고찰을 통해 한국어 단위화를 위한 일련의 규칙들을 제시하였다[7]. 김미영은 신호필[7]이 제시한 규칙들을 기반으로 하는 명사구와 동사구의 단위화 규칙들을 만들어 단위화를 수행하였다[8]. 그러나 이와 같은 규칙 기반의 방법은 규칙을 만드는 데 있어 언어의 제반 현상에 대한 이해와 해당 언어의 구조와 특성에 대한 깊이 있는 고찰을 기본 전제로 한다는 어려움을 가진다. 규칙 기반 방법을 이용한 시스템의 성능이 해당 시스템에서 이용된 규칙에 크게 좌우된다는 점을 고려할 때 이는 상당히 큰 제약 조건이 된다.

이러한 문제점은 Ramshaw와 Marcus가 제시한 transformation-based learning[3]과 같이 기계학습을 통

해 규칙을 자동 생성하는 접근 방법을 통해 어느 정도 극복될 수 있다. 그러나 transformation-based learning의 경우 모든 규칙에 대해 greedy search를 하기 때문에 단위화 속도가 느리다는 문제점을 가진다.

본 논문에서는 통계 기반 방법에 초점을 맞춘다. 통계 기반 방법은 규칙 기반 방법에 비해 처리하고자 하는 언어에 대한 지식이 상대적으로 훨씬 적게 필요하다. 구현 및 활용 측면에 있어서도 적절한 모델을 사용할 경우 시스템에 대한 큰 변경 없이 여러 언어를 처리 대상으로 할 수 있다는 장점을 가지며, 처리 속도 면에 있어서도 transformation-based learning에 비해 상대적으로 빠르다는 장점도 가진다.

본문 전체의 구성은 다음과 같다. 먼저 2장에서는 전체 시스템에 대해 설명하고 시스템에 적용된 확률 모델 및 평탄화 방식에 대해서 설명한다. 3장에서는 실제 시스템을 이용하여 한국어와 영어의 단위화 실험을 수행한 결과를 보인다. 마지막으로 4장에서는 결론을 제시한다.

2. 단위화 시스템

2.1. 확률 모델

본 논문에서 사용하는 단위화를 위한 확률 모델은 Zhou가 [9]에서 제시한 확률기반 모델을 기본으로 하여 개발되었다.

단위화(chunking) 과정은 주어진 token sequence $G_1^n = g_1 g_2 \dots g_n$ 에 대해 가장 적절한 단위화 tag sequence $T_1^n = t_1 t_2 \dots t_n$ 을 결정하는 것으로 정의한다. 이 때 가장 적절한 단위화 tag sequence는 $\log P(T_1^n | G_1^n)$ 을 최대화 시키는 단위화 tag sequence로 정의할 수 있다.

$$\log P(T_1^n | G_1^n) = \log P(T_1^n) + \log \frac{P(T_1^n, G_1^n)}{P(T_1^n)P(G_1^n)}$$

위 식에서 두 번째 term은 tag sequence T_1^n 과 주어진 token sequence G_1^n 간의 상호정보(mutual information)이다. T_1^n 과 G_1^n 간의 상호정보가 G_1^n 과 각 단위화 tag t_i 과의 상호정보 값을 전부 합한 것과 같다고 가정하면, 즉 다시 말해 T_1^n 과 G_1^n 간의 상호정보가 각 단위화 tag t_i 에 대해 독립적이라고 가정하면 해당 term을 다음과 같이 바꿀 수 있다.

$$\log \frac{P(T_1^n, G_1^n)}{P(T_1^n)P(G_1^n)} = \sum_{i=1}^n \log \frac{P(t_i, G_1^n)}{P(t_i)P(G_1^n)}$$

$$MI(T_1^n, G_1^n) = \sum_{i=1}^n MI(t_i, G_1^n)$$

위의 내용을 첫 식에 대입하면,

$$\begin{aligned} \log P(T_1^n | G_1^n) &= \log P(T_1^n) + \sum_{i=1}^n \log \frac{P(t_i, G_1^n)}{P(t_i)P(G_1^n)} \\ &= \log P(T_1^n) - \sum_{i=1}^n \log P(t_i) + \sum_{i=1}^n \log P(t_i | G_1^n) \end{aligned}$$

맨 마지막 식은 term들을 적절히 재배열한 것이다.

2.2. Term Estimation

이제 앞에서 얻어진 식을 통해 실제 $\log P(T_1^n | G_1^n)$ 값을 계산하기 위하여 각 term을 어떻게 계산할 것인지 살펴보도록 한다.

먼저 이 식의 첫 번째 term은 back-off bigram model을 이용하여 간단히 구할 수 있다. 두 번째 term은 각 단위화 tag의 확률의 log값을 전부 더한 것이므로 역시 손쉽게 구할 수 있다.

마지막 term의 근사 방법으로서 Zhou[9]는 바로 앞의 품사 태그, 현 위치의 품사 태그를 기본 자질로 간주하고 전치사나 관사의 경우 어휘까지 추가적으로 자질에 포함시키는 방식을 택하였다.

본 논문에서는 마지막 term을 구하기 위한 자질로 바로 앞의 품사 태그 및 어휘, 현 위치의 품사 태그 및 어휘의 총 네 가지 자질을 모두 이용하였다. Zhou[9]의 방식과 비교하면 특정한 경우에만 어휘 정보를 자질로 선택하는 것이 아니라 모든 경우에 어휘를 자질로 선택하여 이용한다는 차이가 있다.

이 경우 마지막 term은 다음과 같이 표현할 수 있다.

$$\begin{aligned} \sum_{i=1}^n \log P(t_i | G_1^n) &= \sum_{i=1}^n \log P(t_i | g_{i-1} g_i) \\ &= \sum_{i=1}^n \log P(t_i | p_{i-1} w_{i-1} p_i w_i) \end{aligned}$$

이 때 p_i, w_i 는 각각 i 번째 위치의 품사 태그, i 번째 위치의 어휘를 나타낸다.

Zhou는 [9]에서 어휘를 한정적인 경우에 대해서만 자질에 포함시킨 것에 대해 메모리 사용량과 계산상의 복잡도를 줄이기 위해서라고 설명하였다. 이러한 문제 외에도 어휘를 모든 경우에 대해 자질에 포함시키는 것은 비교적 작은 학습 말뭉치를 이용할 경우 자료 부족 문제(data sparseness problem)를 야기시킬 수 있다는 점 또한 심각한 문제가 될 수 있다. 본 논문에서는 자료 부족 문제 쪽에 초점을 맞추어 이를 극복하기 위하여 다음에서 설명하는 것과 같은 평탄화 방식을 이용하였다.

2.3. 평탄화 (Smoothing)

학습을 위해 사용한 말뭉치의 크기가 충분히 크지 않을 경우 자료 부족 문제 (data sparseness problem) 가 일어날 수 있다. 이러한 문제를 해결하기 위해 적당한 평탄화 방법을 선택하는 것도 중요하다.

평탄화에는 크게 두 가지 접근 방식이 있다. 하나는 Add-One이나 Good-Turing 방법과 같이 count를 re-estimate하는 방식이고, 다른 하나는 Katz' Back-off smoothing과 같은 Back-off type의 평탄화이다.

본 논문에서는 기본적으로 Back-off type의 평탄화 방법을 채택하였다. Chen 과 Goodman은 [10]에서 Back-off 기반의 평탄화 방식이 count re-estimation 방식보다 우월함을 보였다.

아래는 Collins와 Brooks가 [11]에서 이용한 것과 유사한 방식의 back-off 평탄화 방식이다.

If $f(x_1, \dots, x_n) > 0$:

$$p(c | x_1, \dots, x_n) = \frac{f(c, x_1, \dots, x_n)}{f(x_1, \dots, x_n)}$$

Elseif $f(x_1, \dots, x_{n-1}, *) + \dots + f(*, x_2, \dots, x_n) > 0$:

$$p(c | x_1, \dots, x_n) = \frac{f(c, x_1, \dots, x_{n-1}, *) + \dots + f(c, *, x_2, \dots, x_n)}{f(x_1, \dots, x_{n-1}, *) + \dots + f(*, x_2, \dots, x_n)}$$

Elseif ...:

$$p(c | x_1, \dots, x_n) = \dots$$

Elseif $f(x_1, *, \dots, *) + \dots + f(*, \dots, *, x_n) > 0$:

$$p(c | x_1, \dots, x_n) = \frac{f(c, x_1, *, \dots, *) + \dots + f(c, *, \dots, *, x_n)}{f(x_1, *, \dots, *) + \dots + f(*, \dots, *, x_n)}$$

위의 알고리즘에서 $f(X)$ 는 학습 데이터에서 자질의 패턴 X 가 나타나는 빈도이다. (*) 표시는 모든 패턴, 즉 wild card로 쓰였다.

Zavrel과 Daelemans는 [12]에서 위의 평탄화 방식이 모든 자질들을 동일한 정도로 중요하게 본다는 점에서 문제를 일으킬 수 있음을 언급하고 이러한 관점에서 위의 평탄화 알고리즘을 Naive smoothing이라고 불렀다.

현실적으로는 모든 자질들이 동일한 정도의 중요도를 가진다고 보기 어렵다. Zavrel과 Daelemans는 [12]에서 이러한 문제를 해결하기 위해 각 자질들의 중요도를 평가하기 위한 척도로 information

gain을 이용하는 것을 제안하고 자질 f 의 information gain(IG)을 다음과 같이 정의하였다.

$$w_f = \frac{H(C) - \sum_{v \in V_f} P(v) \times H(C | v)}{si(f)}$$

$$H(C) = - \sum_{c \in C} P(c) \log_2 P(c)$$

$$si(f) = - \sum_{v \in V_f} P(v) \log_2 P(v)$$

위 식에서 C 는 범주 레이블의 집합, V_f 는 자질 f 들의 집합을 나타내며, $H(C)$ 는 범주 레이블의 엔트로피이다. 확률 값들은 학습 데이터 집합에서의 상대 빈도로 계산할 수 있다. $si(f)$ 는 정규화를 위한 값(normalizing factor)이다.

Zavrel과 Daelemans는 위와 같은 식에 의해 계산된 각 자질의 information gain(IG) 값을 해당 자질의 중요도라고 보고, naïve back-off 알고리즘에서의 sequence 중 가장 중요도가 높은 자질이 빠지는 것을 제외시켰다. Collins와 Brooks는 [11]에서 실험적으로 back-off sequence를 정했는데, 그 결과는 Zavrel과 Daelemans가 위의 방법에 의해 결정된 sequence와 동일했다.

본 연구에서는 위에 설명한 메모리 기반 학습에서 사용되는 방식의 IG back-off 평탄화를 이용하여 새로운 확률모델에 대한 실험을 수행한다.

3. 실험

동일한 시스템으로 각각 영어와 한국어에 대한 단위화를 실험하였다. 두 언어 모두에 대해 비슷한 정도의 성능이 나올 보이기 위해 학습과 평가에 이용된 데이터의 크기도 큰 차이가 없도록 조절하였다. 성능 평가는 양 언어에 대한 실험 모두 CoNLL-2000 shared task[13]에 대한 성능 평가 도구로 제시된 Perl script를 이용하였다¹.

3.1. 영어 단위화

영어 단위화를 위한 데이터는 CoNLL-2000 shared task[13]에서 제시된 것을 이용하였다².

학습 말뭉치는 PENN TreeBank의 WSJ section 15에서 18까지의 4개 section으로, 총 211,727개의 단어로 이루어져 있다.

¹

<http://lcg-www.uia.ac.be/conll2000/chunking/conllval.txt>

²Co-NLL 2000에서 제시된 학습 말뭉치와 평가용 데이터, 평가 도구 및 관련 툴, 데이터 설명 등은 <http://lcg-www.uia.ac.be/conll2000/chunking/>에 제시되어 있다.

평가에 이용된 데이터는 PENN TreeBank의 WSJ section 20으로 총 47,377개의 단어로 구성되어 있다.

CoNLL-2000의 학습 말뭉치와 평가 데이터에 제시된 품사 태그들은 모두 Brill Tagger[14]를 통해 생성된 것이다. 이는 현실적으로 말뭉치에 오류가 있는 편이 더 자연스럽다는 점과, 실제 시스템을 통해 단위화를 수행할 때 주어진 문장에 대해서 품사 태거가 태그를 완벽하게 할당할 수 없다는 점을 감안한 것이다.

본 논문에서도 실제 성능 평가에 평가용 데이터에 함께 주어져 있는 Brill Tagger의 결과인 품사 태그를 바로 이용하였다. 입력 문장에 대해 품사 태거를 이용하여 태그를 할당하지 않고 주어진 품사 태그를 바로 이용한 것은 실제 단위화 시스템 자체의 성능을 비교하기 위해서는 Brill Tagger의 오류를 그대로 인정하고 실험하는 편이 좋다고 생각되었기 때문이다.

단위화 태그는 CoNLL-2000의 데이터에 제시된 것을 따라 11개의 태그를 이용하였다.

다음은 마지막 term을 estimation하기 위해 이용된 네 가지 자질에 대해 Naive back-off 방식으로 평탄화를 하여 단위화를 수행한 결과이다. 평가 데이터에 등장하지 않았던 UCP 태그의 경우 성능 평가에 포함시키지 않았다.

단위화 tag	precision	recall	$F_{\beta=1}$
ADJP	67.8%	63.0%	65.3
ADVP	76.7%	70.6%	73.5
CONJP	38.9%	77.8%	51.9
INTJ	12.5%	50.0%	20.0
LST	0.0%	0.0%	0.0
NP	90.0%	92.0%	91.0
PP	94.7%	94.6%	94.7
PRT	71.2%	79.3%	75.0
SBAR	79.6%	74.6%	77.0
VP	89.1%	90.3%	89.7
All	89.5%	90.4%	90.0

결과 중 LST의 경우, 평가 데이터 내에 총 5번 밖에 등장하지 않으며, CoNLL-2000에 제출된 시스템들 대부분이 잘못된 단위화 태그를 부여하였다.

IG back-off 평탄화 방식을 적용하기 위하여 마지막 term 값의 추정에 이용된 네 가지 자질의 information gain을 구한 결과는 $p_{i-1}, w_{i-1}, p_i, w_i$ 에 대해 각각 0.22, 0.15, 0.41, 0.21이었다. 이에 의해 가장 중요한 자질로 현 위치의 품사 태그를 선택하여 back-off sequence를 정하였다. 다음은 그 실험 결과이다.

단위화 tag	precision	recall	$F_{\beta=1}$
ADJP	67.1%	62.8%	64.9
ADVP	76.5%	71.0%	73.6
CONJP	38.9%	77.8%	51.9
INTJ	12.50%	50.0%	20.0
LST	0.00%	0.0%	0.0
NP	89.93%	92.0%	90.9
PP	94.8%	94.6%	94.7
PRT	71.2%	79.3%	75.0
SBAR	79.6%	74.2%	76.8
VP	88.9%	90.3%	89.6
All	89.5%	90.4%	90.0

기본 모델로 이용한 Zhou[9]의 경우 precision, recall, F-value가 각각 89.6%, 89.6%, 89.6이었던 것과 비교하여 성능이 향상된 것을 확인할 수 있다.

Naive back-off 평탄화를 이용한 결과와 IG back-off 평탄화를 이용한 결과는 거의 차이가 없는 것으로 나타났다. 이것은 네 자질의 IG 값 차이가 상대적으로 적기 때문으로 생각된다. 또한 현 위치의 품사 태그를 보지 않는 back-off sequence의 경우 학습 말뭉치 크기가 비교적 작기 때문에 거의 나타나지 않았을 것이므로, 그러한 sequence를 제외시키는 것이 성능에 큰 영향을 미칠 수 없었을 것이라는 점도 생각할 수 있다.

다음은 CoNLL-2000의 상위 5개 시스템들의 실험 결과이다.

System	precision	recall	$F_{\beta=1}$
ZDJ[15]	94.29%	94.01%	94.13
KM1[16]	93.89%	93.92%	93.91
KM0[17]	93.45%	93.51%	93.48
Hal[18]	93.13%	93.51%	93.32
TKS[19]	94.04%	91.00%	92.50

Hal[18]과 TKS[19]의 경우 둘 이상의 시스템의 결과를 놓고 voting하는 방법을 이용하였다. KM1[16]은 KM0[17]을 확장한 것으로 기본적으로는 같은 시스템으로 볼 수 있다. ZDJ[15]의 경우 기본 시스템은 기계 학습 기법을 이용한 것을 이용하며 추가적으로 의존 구조를 도입하여 성능을 향상시킨 것이다.

위에 제시된 시스템들의 경우 주어진 평가 데이터에 대해 튜닝 과정을 거쳤을 가능성이 있다. 이에 비해 본 논문에 제시된 실험 결과들은 튜닝을 하지 않은 확률 기반의 단일 시스템을 그대로 이용한 것이다. 가령 Zhou는 [9]에서 추가적으로 규칙을 통해

오류를 수정하는 방식을 도입하여 실험한 결과도 제시하고 있는데, 약 2.5% 가량의 성능 향상을 보였다.

3.2. 한국어 단위화

한국어의 단위화에 대한 실험은 <http://bi.snu.ac.kr/~sbpark/Step2000/>에 공개되어 있는 STEP 2000 dataset의 일부분을 이용하였다. 전체 데이터는 321,328 단어로 이루어진 것으로, 이 중 앞부분 243,334 형태소를 학습 말뭉치로 이용하고 나머지 중 46,337 형태소는 평가에 이용하였다. 이것은 앞서 영어 단위화 실험에 이용한 것과 학습 및 성능 평가에 이용된 데이터의 크기가 비슷한 정도가 되도록 조절한 것이다. 실험에 이용된 시스템은 한국어 말뭉치에 대해 학습된 확률 사전을 이용하는 점 외에는 영어 단위화에 이용한 것과 완전히 동일하다.

단위화 태그는 말뭉치에 나와 있는 것을 따라 IP, NP, VP, ADVP의 네 가지를 이용하였다.

다음은 마지막 term을 estimation하기 위해 이용된 네 가지 자질에 대해 Naive back-off 방식으로 평탄화를 하여 단위화를 수행한 결과이다.

단위화 tag	precision	recall	$F_{\beta=1}$
IP	83.0%	96.1%	89.1
NP	88.2%	89.5%	88.8
VP	87.6%	93.8%	90.6
ADVP	97.1%	96.7%	96.9
All	88.4%	91.7%	90.0

IG back-off 평탄화 방식을 적용하기 위하여 마지막 term 값의 추정에 이용된 네 가지 자질의 information gain을 구한 결과는 $p_{i-1}, w_{i-1}, p_i, w_i$ 에 대해 각각 0.26, 0.13, 0.33, 0.17이었다. 다음은 이를 적용하여 단위화를 수행한 결과이다.

단위화 tag	precision	recall	$F_{\beta=1}$
IP	82.8%	96.1%	89.0
NP	88.0%	89.5%	88.8
VP	87.5%	93.8%	90.6
ADVP	97.1%	96.7%	96.9
All	88.3%	91.7%	89.9

두 언어의 경우 단위화 태그가 서로 다르므로 위의 결과를 영어 실험 결과와 정확히 비교하기는 어렵지만, 전체 precision, recall, F-value를 보았을 때 대략으로 비슷한 결과를 보인다고 생각할 수 있다.

4. 결론

본 논문에서는 확률 기반 모델을 이용하여 텍스트를 단위화 하는 방식을 제안하였다. 또한 비교적 작은 학습 말뭉치를 이용할 때 야기될 수 있는 자료 부족 문제를 해결하기 위해 메모리 기반 학습에서 사용된 IG back-off 평탄화 방식을 이용하여 성능을 테스트하였다.

한국어, 영어의 두 언어에 대한 실험 결과 거의 비슷한 성능을 보임을 확인할 수 있었다. 이는 본 논문의 모델을 적용한 시스템이 다양한 언어에 대해서 언어 무관하게 고른 성능을 보일 수 있음을 나타낸다고 볼 수 있다.

또한 이와 같은 특징으로 인해 본 모델은 특정 언어 domain에 대한 지식 없이 단위화 시스템을 개발하는 데에 적용할 수 있는 가능성을 보였다는 의의를 가진다.

5. 참고 문헌

- [1]. Steven Abney, "Parsing by Chunks", Kluwer Academic Publishers, 1991.
- [2]. J. Kupiec, "An algorithm for finding noun phrase correspondences in bilingual corpora", *ACL '93*, pp 17-22, 1993.
- [3]. L. A. Ramshaw and M. P. Marcus, "Transformation-based Learning", In *Proceedings of 3rd ACL Workshop on Very Large Corpora at ACL '95*, 1995.
- [4]. W. Skut and T. Brants, "Chunk Tagger: statistical recognition of noun phrases", *ESSLLI-1998 Workshop on Automated Acquisition of Syntax and Parsing*, 1998.
- [5]. J. Veenstra, "Memory-based text chunking", In *Workshop on machine learning in human language technology at ACAI '99*, 1999.
- [7]. 신효필, "최소자원 최대효과의 구문분석", 제 11회 한글 및 한국어 정보 처리 학술 대회, pp. 242-248, 1999.
- [8]. 김미영, 강신재, 이종혁, "단위분석과 의존문법에 기반한 한국어 구문분석", 제 27회 정보과학회 봄 학술대회 논문집, pp. 327-329, 2000.
- [9]. GuoDong Zhou, Jian Su and TongGuan Tey, "Hybrid Text Chunking", In *Proceedings of CoNLL-2000 and LLL-2000*, pp. 163-165, 2000.
- [10]. Stanley F. Chen and Joshua Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", In *Proc. of the 34th Annual Meeting of the ACL*, 1996.
- [11]. M. Collins and J. Brooks, "Prepositional Phrase Attachment through a Backed-Off Model", In *Proceedings of the Third Workshop on very Large Corpora*, 1995.
- [12]. Jakob Zavrel and Walter Daelemans, "Memory-Based Learning: Using Similarity for Smoothing", In *Proc. of the 35th Annual Meeting of the*

ACL, 1994.

[13]. Erik F. Tjong Kim Sang and Sabine Buchholz, "Introduction to the CoNLL-2000 Shared Task: Chunking", In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, 2000.

[14]. Eric Brill, "A corpus-based approach to language learning", Ph.D thesis, Univ. of Penn., 1993.

[15]. Tong Zhang, Fred Damerau and David Johnson, "Text Chunking using Regularized Winnow", In: *Proceedings of ACL-2001*, Toulouse, France, 2001.

[16]. Taku Kudoh and Yuji Matsumoto, "Chunking with Support Vector Machines", In: *Proceedings of NAACL 2001*, 2001.

[17]. Taku Kudoh and Yuji Matsumoto, "Use of Support Vector Learning for Chunk Identification", In: *Proceedings of CoNLL-2000 and LLL-2000*, 2000.

[18]. Hans van Halteren, "Chunking with WPDV Models", In: *Proceedings of CoNLL-2000 and LLL-2000*, 2000.

[19]. Erik F. Tjong Kim Sang, "Text Chunking by System Combination", In: *Proceedings of CoNLL-2000 and LLL-2000*, 2000.