약어 생성 유형을 고려한 한국어 약어 사전 자동 구축

윤여찬, 송영인, 이주영, 임해창
고려대학교 자연어처리 연구실
{ycyoon, sprabbit, jylee, rim}@nlp.korea.ac.kr

Construction of Korean acronym dictionary by considering ways of making acronym from definition

Yeo-Chan Yoon, Young-In Song, Joo-Young Lee, Hae-Chang Lim
Korea University, Seoul, Korea
{ycyoon, sprabbit, jylee, rim}@nlp.korea.ac.kr

요약

본 논문에서는 한국어 고유명사 약어 사전을 자동으로 구축하기 위한 방법론을 제안한다. 본 논문은 원어로부터 약어가 생성되는 방식을 네가지 유형으로 분류 한 후 각 유형에 따라 가능한 약어의 후보들을 생성하여 원어, 약어 후보 쌍을 수집하고, 수집 된 각 쌍에 대하여 확률적모형에 근거, 실제 사용되는 원어, 약어 쌍을 선별하여 사전에 등재함으로써 자동으로 사전을 구축 할 수 있도록 한다.

1. 서론

약어란 본디의 음절이나 형태소 등이 줄어서 된 말로 정의 할 수 있다. 대한민국 나라말 사전

* "이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 "(KRF-2005-041-D00737) 약어는 언어의 효율적 사용을 위해 문서나 일상 언어 등에서 널리 사용되는 언어현상인데 특히 웹 문서, 블로그 등 구어체 환경에서 널리 사용 된다. 가령 '국산품'을 '국산'으로 혹은 '한국전력공사'와 같은 기업명을 '한전'으로 줄여 사용하는 것이 이러한 약어 사용의예라고 할 수 있다.

약어를 인식, 복원하는 문제는 정보검색, 질의응답과 같은 작업의 성능 향상을 위해 요구 되는 일 중 하나라 할 수 있다. 질의에 약어만이 존재하고 문서에 원어만이 존재할 경우, 혹은 그 반대의 경우에 관련문서를 검색하거나 정답을 도출하는 데에 심각한 성능의 저하를 야기 시킬 수 있기 때문이다. 따라서 약어의 인식및 복원 문제를 해결해야 할 필요성이 존재한다.

문서 내에서 약어를 인식하고 복원하기 위해서 크게 두 가지 접근방법을 생각할 수 있다.

첫 번째로, 문서 내의 정보를 이용하여 직접 약어를 인식하는 방법이다. 이는 약어가 다른 일반적인 단어와 쉽게 구별 될 수 있는 특징을 가지고 있을 때 이를 통해서 약어를인식 한 후, 인식된 약어의 원어를 같은 문서 내에서 찾는 방식으로 주로 영어권에서 사용 된다. 이러한 연구로 Taghava[1]의 연구를 들 수 있는데, Taghava는 영어의약어가 주로 대문자로 이루어져 있고, 약어를 생성할때 (International Business Machine, IBM)과 같이 원어를구성하는 각 명사의 첫 음절을 정렬하는 경향이 있다는 것에 기반하여 문서 내에서 3~10음절의 대문자로 구성된 단어를 약어후보로 인식 한 후, 구성 명사의 첫 음절이 약어의 각 음절과 대응되는 명사구를 같은 문서 내에서 찾아 주는 방식으로 복원문제를 해결하였다.

두 번째로, 약어사전 등의 외부자원을 이용하여 문서 내에서 약어의 후보를 인식하고 중의성이 발생할 경우 앞 뒤 문맥 등을 고려하여 이를 해결 한 후, 복원 역시 사전을 통해 해결하는 방법을 생각할 수 있다. 사전을 이용할 경우, 같은 문서 내에 원어가 존재 하지 않아도 약어의 복원이 가능하다는 장점이 있다.

한국어 약어 인식 및 복원의 어려움은, 영어와 달리 약어를 다른 일반적인 미등록어 단어와 구별할 뚜렷한 실마리가 없다는 데 있다. 가령 영어의 경우 "John applied to IBM"에서 약어인 'IBM'과 약어가 아닌 'John'과 같은 단어를 두 개 이상의 대문자 포함여부에 따라 쉽게 구별할 수 있지만 한국어의 경우 "영철이가 한전에 지원했어"와 같은 문장에서 약어인 '한전'과 약어가 아닌 '영철'을 구별할 수 있는 뚜렷한 특징이 존재하지 않는다. 따라서 앞서 말했던 첫 번째 방법을 이용, 영어와 같이 다른 단어와 구별되는 약어의 특

징에 기반하여 약어의 인식 및 복원 문제를 해결하기 어렵다. 반면 약어 사전을 이용하여 약어를 인식하는 방법의 경우, 기구축된 사전이 존재한다면, 문서에 출현하는 약어가 사전에 등재되어 있는지의 여부를 통해 인식, 복원 문제를 해결 할 수 있다. 따라서 한국어약어의 인식 및 복원 문제 해결을 위해서는 기구축된 사전을 이용하는 방법이 적합하다고 할 수 있다.

약어의 특징 중 하나는 필요에 따라 지속적으로 생성 된다는 것이다. 기업명, 영화제목과 같은 고유명사의 경우 지속적으로 새롭게 생성되며, 이러한 고유명사의 상당수가 약어를 가지는 경향이 있기 때문에 약어도 새롭게 계속 생성 된다. 이러한 특성 때문에 약어사전은 지속적으로 갱신되어야

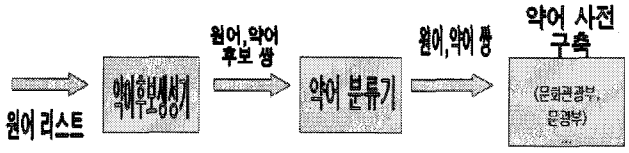

그림 1. 시스템 구성도

하며 이는 수동으로약어의 사전을 구축하는 데 어려움을 준다. 따라서 약어 사전을 효율적으로 자동 구축하는 방안이 필요하다. 본 논문에서는 약어사전을 자동으로 구축하는데 초점을 맞추어, 향후 약어 인식 및 복원 등의 작업에 도움이될 수 있도록 한다.

사전 구축을 위한 기존 연구의 경우, 약어 인식 및 복원 기술을 활용한 경우가 많은데 Leah[2]의 경우, 대용량의 문서에 대하여 인식및 복원을 수행하여 대량의 약어, 원어 쌍을 수집 함으로써 사전을 구축하였다. 하지만 한국어의 경우, 앞에서 언급하였듯이 문서 내에서 인식 및 복원을 수행하기가 어렵기 때문에 이와 같은 방법론을 적용하기가 어렵다.

본 논문에서는 문서에서 약어, 원어를 추출하는 방식 대신, 원어를 통해 약어를 직접 예측, 수집하는 방식을 제안한다. 이를 위해 약어 생성 방식을 고려, 주어진 원어에 대하여 가능한 약어 후보를 직접 예측 하고, 확률적 모형에 기반하여 이들 중, 올바른 원어, 약어 쌍을 분류한 후 사전에 등재한다. 이 때 원어, 약어 쌍을 올바로 분류하기 위하여 약어 생성시의 특성, 약어의 실제 사용 여부 등을 고려하여 자질로 구성한다. 이러한 방식을 사용하여 문서에서 약어 인식을 먼저 수행하지 않고도 약어 사전을 구축할 수 있도록 하였다.

2. 제안하는 방법

그림 1은 본 논문에서 제시하는 약어 사전 구축 방법을 제시하고 있다. 주어진 원어리스트에서 가능한 약어를 예측하여 사전을 구축하기 위해서는 원어의 목록을 얻는 작업이 필수적인데 회사명, 기관명 등과 같이 약어가 빈번히 발생하는 원어들의 목록은 웹 사이트 등을 통해 수집이 용이하다. 다음 절에서는 주어진 원어 리스트를 통해 약어의 후보를 생성하는 방법과 생성된 원어, 약어 후보 쌍에 대해 올바른 쌍을 분류하고, 이를 통해 최종적으로 사전을 구축하는 방법에 대하여 기술한다.

2.1 약어 후보 생성

n개의 음절로 이루어진 원어의 경우, 원어를 구성하는 각 음절의 생략여부를 고려하여 2^n-2개의 가능한 약어의후보를 생각할 수있다. 가령 7음절로 이루어진 약어 "대우자동차판매"의 경우 126개의 약어 후보를 생성하게 되는데 이는 약어 판정에 많은 시간을소모하게 한다. 따라서 약어의 수를 줄여 줄 필요성이 존재한다. 본 논문에서는 약어의 생성 방식을고려하여 효율적으로 약어 후보의 수를 줄여 줄 수 있도록 한다. 약어의 생성 방식은 구성 명사 중 일부를 생략하거나 특정 음절만을 조합하는 방식이라고 볼 수 있는데 가령 한국전력공사의 약어인 한국전력의 경우 원어를 한국, 전력, 공사로 분리하였을 때 공사라는 명사가 탈락한 형태로 약어가 생성되었고, 한전의 경우 한국전력공사의 앞의 두 명사 한국, 전력의 첫 음절을 조합함으로써 약어가 생성되었다고 볼 수 있다. 따라서 한국어 약어의 경우, 원어를 구성하는 명사를 기준으로 약어가 생성 되며 또한 약어의 생성방식이 비교적 전형적이라고 할 수 있다. 이러한 기준에 따라본 논문에서는 약어의 생성유형을 다음과 같이 분류 하였다.

- **명사생략형:** 원어를 구성하는 일부 단일 명사를 생략 함으로서 약어를 생성하는 방식을 본 논문에서는 명사생략형이라 칭하였다. `한국', `전력', `공사'의 세 개의 명사를 결합, 생성한 원어 `한국전력공사'에 대하여, `공사'라는 명사를생략하여 `한국전력'이라는 약어를 생성한 것이 이러한명사생략형 약어의 예라고 할 수 있다.
- **음절조합형:** 명사를 구성하는 단일 명사 중, 일부 명사에 대하여특정 음절을 하나씩 뽑아 배열함으로써 약어를 생성하는 방식이다. `한국전력공사'의 두 단위 명사 `한국', `전력'에서 앞 음절을 따 `한전'으로 줄여 사용하는 것이 음절조합형의 예이다.
- **혼합형:** 명사생략형과 같이 원어를 구성하는 일부 명사가 약어에온전히 출현하며, 음절조합형과 같이 일부 명사에 대해서는 특정 음절만이 출현한 형태가 혼합된 유형을 본 논문에서는 혼합형이라고 칭하였다. `대우자동차판매'(대우+자동차+판매)의 약어인 `대우자판'이 혼합형 약어의 예이다.
- **명사축약형:** 명사축약형은 특정 명사에 대하여 첫 음절부터 차례로두 음절 이상을 뽑아 약어를 구성하는 형태로 일부 명사가 축약 되었다고 볼 수 있다. `이수페타시스(이수+페타시스)'에서 '페타시스'의 뒤의 두 음절을 생략하여 `이수페타'로 축약한 것이 예

이다.

음절조합형과 혼합형의 경우, 약어를 구성할때 사용하는 특정음절은 대부분 단어의 첫 음절이고 그 외의 음절이 사용 된다 해도 대부분 마지막 음절인 경향이 있다. 실제로 실험 데이터로 사용한 원어, 약어 쌍 839개에 대하여 첫 음절이나 끝 음절 이외의 음절이사용된 경우는 한번도 없었다. 명사축약형은 다른 세 유형에 비해 비교적 그 수가 적은 약어유형으로 주로 원어에 외래어와 같은 4음절 이상의 긴 단일명사가 포함된 경우에 발생한다. 따라서 명사축약형의 경우, 4음절 이상의 단일명사가 원어에 포함되어 있을 때 발생할 것이라 예측할 수 있다. 이 같은 특성에 기반하여 아래와 같은 휴리스틱을 사용, 후보 수를 줄여 주는 것이 가능하다.

1. 첫 음절, 혹은 끝 음절이 아닌 음절이 뽑혀 사용된약어 후보 제거
2. 4음절 이상의 단일명사가 포함된 경우에만 명사축약형 약어 생성

생성유형을 고려하여 약어를 생성 후, 위의 두 휴리스틱을 사용하면, 126개의 약어 후보가 생성되던 `대우자동차판매'에 대하여 후보 수를 62개로 줄여 줄 수 있다.

이러한 방법으로 약어의 후보를 생성하기 위해서는 원어를 구성하는 최소명사로 나누는 작업이 필요하다. 이를 위해서 본 논문은 CYK 알고리즘 [3]을 이용하였으며 최소명사가 대부분 2,3음절로 구성된 것을 고려, 음절의 길이에 따라가중치를 적용하여 한 개의 분석 결과만을 사용하였다.

2.2 약어 분류

생성된 원어, 약어 후보 쌍 중, 올바른 쌍을 분류하기 위하여 확률적 모형을 이용한다. 본 논문에서는 비교적 작은 데이터 집합에서도 견고하다고 알려진 Naïve bayes Classifier를 이용하여 약어를일반 단어와 분류할 수 있도록 한다. 원어(d_j)와 약어 후보(w_i)가 주어진 상태에서, 해당 약어 후보가 원어에 대하여 실제 약어인 관계일확률은 아래와 같이 정의 할 수 있다.

$$p(c \mid w_i, d_j)$$

$$= p(c \mid f(w_i, d_j)) \quad \begin{cases} c = acro \\ c \neq acro \end{cases}$$

$$= p(c \mid f_1 f_2 \cdots f_n)$$

위의 식은 w_i가 약어로 분류$(c = acro)$될 확률을 나타낸다. w_i와 d_j는 자질 집합 $f(w_i, d_j)$로 바꾸어표현 가능 하여 이는 다시 자질집합을 구성하는 자질로 나누어 $f_1 f_2 \cdots f_n$으로 나타낼 수 있다. 이를 다시 Naïve bayes Classifier로 아래와 같이 표현, 최종적으로 올바른 원어, 약어 쌍을 분류할 수 있도록 한다.

$$f(w,d) = \arg\max_{c \in \{acro, non-acro\}} p(c \mid f_1 f_2 \cdots f_n)$$

$$= \arg\max_c \frac{p(f_1 f_2 \cdots f_n \mid c)p(c)}{p(f_1 f_2 \cdots f_n)}$$

$$= \arg\max_c p(f_1 f_2 \cdots f_n \mid c)p(c)$$

$$= \arg\max_c p(c) \prod_{i=1}^{n} p(f_i \mid c)$$

3. 자질집합

약어를 구별하기 위한 자질은 축약의 정도, 음절 선택시의 경향 등 약어 생성 특징 및 약어의 실제 여부를 고려하여 구성하였다. 약어의 사용 여부는 실제 사용되는 단어라면 웹 문서에 출현할 것이라는 가정 아래웹 문서에서 해당약어가 사용되는지의 여부에 따라 파악한다. 웹 문서는 정보검색을 이용하여 수집 하되, 약어후보와 원어를 질의로 한 것, 약어후보만을 질의로 한 것의 두 종류 문서를 각 질의에 대하여 50개씩 수집 하여 사용한다. 본 논문에서 사용된 자질은 아래와 같다.

- AbbNum: 약어 생성시 생략되는 명사의 수
- DiffNoun: 약어 생성시 생략된 명사와 생략되지 않은 명사의 수의 차이
- AcroLen: 약어의 음절 수
- DiffLen: 원어와 약어의 음절의 수의 차이

'한국전력'이라는 약어의 경우, 원어 '한국전력공사'에서 한 개의 단위명사 '공사'가 생략되었다 볼 수 있고, 두 개의 음절 '공','사'가 생략되었다고 볼수도 있다. 따라서 축약의 단위로 음절과 단위명사 두 가지를 모두 고려하여 다음과 같이 자질을 구성, 어느 정도의 축약이 발생하는지를 본다.

- **#ofLastChar: 선택된 끝 음절의 개수**

음절조합형 및 혼합형에서 음절을 선택할 경우, 한국전력공사의 약어인 한전과 같이 단위명사의 첫 음절을 선택하는 경우가 흔하고, 끝 음절을 선택하는 경우는 많지않다. 따라서 약어에 포함된 끝 음절의 수를 고려한다.

- **#ofRepChar: 혼합형에서의 조합된 음절 개수**

혼합형에서 단위 명사의 특정 음절을 뽑아 조합할 때, 몇 개의 특정 음절을 사용하는 지를 고려한다.

- **CoOccurFreq: 수집한 문서에서의 공기빈도,**
- **DefFreq: 약어후보를 질의로 하여 검색한 문서에서 원어의 출현 빈도**

약어의 경우, 원어와 동시에 웹 문서에서 출현하는 경향을 찾을 수 있으며, 반면에 약어후보가 대응되는 원어의 약어로 쓰이지 않을 경우 공기 하

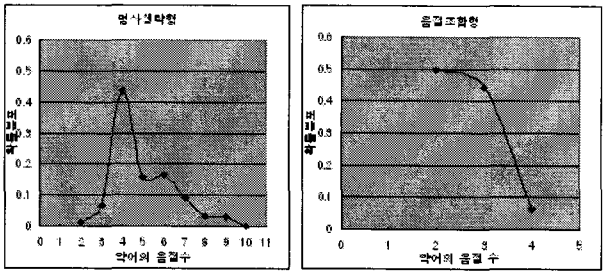

그림 2. 음절 조합형과 명사생략형 약어의 길이의 유형

는 문서를 찾기 어렵다. 다만 약어후보가일상적으로 빈번하게 사용되는 단어와 동음이의어일 경우, 우연히 원어와 공기 할 수 있기에 이를 약어를 질의로 한 문서에서의 원어의 출현 빈도를 통해서 우연성의 여부를 확인 할 수 있도록 한다. 빈번한 단어와 동음이의어인 약어후보를 질의로 하여 검색할 경우, 상대적으로 원어가 검색될 확률이 적고, 빈도도 적을 것이기 때문이다.

4. 생성유형별 특성을 고려한 모형 분리

한국어 약어의 경우, 생성유형별로 길이 등의 특성이 서로 다른데 가령 그림 2에서 나타난 것과 같이 생성유형별로 선호하는 음절의 수가 다른 것을 확인 할 수 있다. 또한 특정 자질의 경우, 특정 유형에만 적용할 수있으며(e.g. #OfRepChar) 따라서 생성유형별로 학습집합 분리하고, 고려하는 자질을 달리하여 모형을 구성하여 성능의 향상을 기대할 수 있다. 본 논문에서는 생성유형을 고려하여 모형을 분리한 경우와 유형을 고려하지 않
는 모형 두 가지를 적용, 각 성능을 평가하였다.

5. 실험 및 평가

5.1 실험 환경

실험을 위해 기업명, 국가단체명, 대학명으로 구성된 4음절 이상의 314개의 원어가 사용되었다. 314개의 원어 중, 307개의 원어에 대한 835개의 원어, 약어 쌍을 Positive Example로 사용하였고,
314개의 원어로 생성 가능 한 50139개의 부적절

자질	정확률	재현율	F maesure
Baseline	47.91%	68.60%	56.41%
AbbNum	52.71%	67.20%	59.08%
DiffLen	50.15%	67.70%	57.62%
#OfLastChar	58.78%	69.60%	63.74%
all	58.68%	73.00%	65.06%

표 1. 생성유형별로 모형을 나누지 않았을 때, 각 자질 추가시의 성능

유형	자질	정확률	재현율	F-Measur e
명사생략형	DiffNoun	72.58%	69.33%	70.92%
음절조합형	#OfLastChar+DefFreq	67.47%	68.29%	67.88%
혼합형	DiffLen+#OfLastChar+#OfRepChar	51.67%	79.45%	62.62%
통합 성능		64.52%	72.00%	68.05%

표 2. 생성유형을 고려하였을 때의 성능

한 원어, 약어 쌍을 Negative Example로 구성하여 학습하였다.
평가와 학습을 위하여 5 Fold Cross Validation을 이용하였고, 각 결과 값의 평균을 통해 성능을 측정하였다. 평가를 위한 정확률과 재현율은 다음과 같이 계산하였다.

$$\text{정확률} = \frac{\text{시스템이 찾아낸 올바른 약어, 원어 쌍의 수}}{\text{시스템이 찾아낸 전체 약어, 원어 쌍의 수}}$$

$$\text{재현율} = \frac{\text{시스템이 찾아낸 올바른 약어, 원어 쌍의 수}}{\text{정답 집합에 존재하는 약어, 원어쌍의 수}}$$

5.2 결과 및 분석

Baseline으로는 다른 특성을 고려하지 않고, 웹 문서에서 원어와 약어의 공기 빈도만을 자질로 사용한 분류기의 성능을 사용 한다. 표1은 Baseline의 성능과 생성유형별로 모형을 나누지 않은 경우에 대하여 Baseline에 성능을 향상시키는 유용한 자질을 추가하였을 때의 성능을 나타낸다. Baseline에 비해 약 8.4%의 성능이 향상됨을 알 수 있다. 표2는 약어의 생성유형에 따라 자질 및 학습 집합을 달리 하였을 때의 성능 및 유용한 자질과 전체 학습 집합에 대한 통합 성능을 보여준다. 명사축약형의 경우, 의미 있는 수준의 학습집합이 없어 유형별로 모형을 나누지 않은 경우에만 성능을 측정하도록 한다. 생성 유형을 고려하지 않았을 때에 비하여 약 3%의 성능이 향상 됨을 확인 할 수 있다.

분류기가 실제 약어가 아닌 단어를 약어로 분류한 경우 중 상당수는, 철자 오류가 발생한 단어가 웹에 출현하였을 경우이다. 이러한 단어의 상당수가 (대우자동차판매, 대우자동판매)와 같이 중간의 한글자가 탈락하는 경우임을 감안하여 이러한 후보를 제거함으로써 70.51%의 정확률과 71%의
재현율, 70.75%의 F-measure 값을 얻을 수 있었다.

약어 사전의 경우, 목적에 따라 재현율 보다는 정확률이 중요 할 수 있고, 또한 그 반대의 경우가 중요할 수도 있다. 따라서 이를 조정할 필요성이 요구 된다. 본 논문에서는 분류기의 확률 결과

값을 정규화 한 후, 약어라고 판단 한 후보 중 약어일 확률과 약어가 아닐 확률의값의 차이가 일정이상 되지 않을 경우 약어라 판단 하지 않는 방법으로 정확률을 높이고 재현율을 낮추었다. 아래는 이 같은 방식으로 약어를 판단하기 위한 식을 나타낸다.

$$f'(w,d)=\begin{cases} acronym & \text{if } f(w,d)=acro \text{ and } f_{rel}(f_1 f_2 \cdots f_n) > treshold \\ non-acronym & \text{else} \end{cases}$$

$$where\ f_{rel}=\frac{p(c=acro)\prod_{i=1}^{n} p(f_i \mid c=acro) - p(c=non-acro)\prod_{i=1}^{n} p(f_i \mid c=non-acro)}{\sum_{c} p(c)\prod_{i=1}^{n} p(f_i \mid c)}$$

정확률을 낮추고 재현율을 높이기 위해서는 이와 반대로 약어라고 판단하지 않은 후보에 대하여확률 값의 차이가일정 값 이상이 될 경우, 약어라고 판단 하였다. 표 3은 이와 같은 방법으로 조정하였을 때의 정확률과 재현율 및 F-measure 값을 나타낸다.

	정확률	재현율	F-Measure
정확률 상승	82.30%	54.40%	65.50%
재현율 상승	53.38%	83.00%	64.97%

표 3. 정확률 및 재현율 조정시의 성능

6. 결론

본 논문은 제한된 문서 내에서 후보를 검색할 때와 달리 주어진 원어에 대하여 가능한 약어를 찾는 방식으로 한국어에 적합한 구축방식을 제안하였다. 또한 이러한 방식을 통해 약어의 인식 및 복원 작업을 선행하지 않고 약어 사전을 자동으로 구축 하였다.

본 논문에서는 확률 모형에 기반하여 약어 여부
를 판정하였는데, 이 때 판정을 위한 유용한 자질을 발견하여 성능의 향상을 가져왔다. 약어 후보 생성시에는 약어의 생성유형별로 약어 후보를 생성 하여 가능한 약어의 후보를 감소시킴으로써 판정을 위한 시간을 절약하였다.

또한 생성유형별로 약어의 특징에 차이가 있다는 점을 감안하여 모형을 분리하였고, 실험을 통해 이 같은 방식을 사용할 경우 성능이 향상됨을 보였다.

참조문헌

1. Kazem Taghva and Jeff Gilbreth. Recognizing acronyms and their definitions. technical report Taghva95-03, ISRI ISRI, November 1995
2. Larkey, L., Ogilvie, P., Price, A. and Tamilio, B. (2000) Acrophile: An Automated Acronym Extractor and Server, In Proceedings of the ACM Digital Libraries conference, pp. 205-214.
3. A.V. Aho and J.D. Ullman. Parsing, volume 1 of The Theroy of Parsing, Translation and Compiling. Prentice-Hall,1972