

OCR (Optical Character Recognition)

1. OCR Technology

OCR 이란 광학 문자 인식(Optical Character Recognition)을 말하는 것으로, 이미지를 기계가 읽을 수 있는 텍스트 포맷으로 변환하는 과정을 말한다. 1920 Emanuel Goldberg 의 글자 판독 및 전신신호 변환 장치를 시초로 이러한 자동 인식 기술은 계속 발전해왔고, 이제는 생활 어디서나 찾아볼 수 있는 기술로, 자동차 번호판 인식, 영수증 인식 등이 OCR 에 기반해 있다. 이전에는 원형정합(template matching) 방법을 사용했다면 이제는 구조 분석적(structural analysis) 방법과 인공신경망(neural network) 방법이 주로 사용된다. 최근 딥러닝에 대한 높은 관심과 발전 덕분에 OCR 기술도 거의 다 딥러닝 인공신경망을 통해 이뤄진다.

2. OCR Transformation

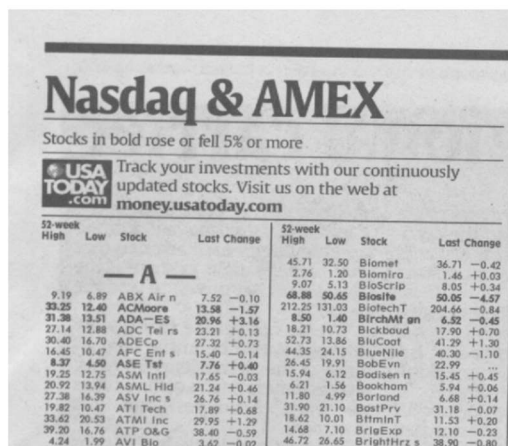
먼저, 변환하려는 인풋 미지를 받으면 전처리(pre-processing) 과정을 거친다. 전처리는 기울기 보정, 명암대비 향상, 수평 맞춤 등이 있다. 이 과정을 거치면 수평을 맞춘 흑과 백의 명확한 이미지로 바뀌고 segmentation 작업을 통해 백은 0, 혹은 1 로 변환된다. 예전에는 각각 Unsharp Mask, HE, Median 등의 통계에 기반한 머신러닝 기법을 사용했지만 최근에는 이 전처리 과정에도 딥러닝 알고리즘 SRN, LLCNN, DnCNN 등을 사용하기 시작했다. 이후에 머신러닝 또는 딥러닝 알고리즘을 접목해 특징 추출을 함, 특징들로는 선, 닫힌 고리, 선 교차 등이 있다. 그 후 가장 비슷한 폰트 구하고 가장 비슷한 글자를 추출해 기본적인 변환은 완성된다. 이 텍스트들을 구한 다음에 semantic learning 을 통해 문맥을 파악해 더 정교하게 완성하는 후처리 과정 등도 있다.

3. Public OCR API

OCR 을 이용해 하나의 이미지를 변환하는 것은 금방 하지만 대용량의 파일을 실시간으로 병렬로 변환하는 데에는 더 좋은 컴퓨팅과 클라우드 서버가 필요하다. 직접 만들어 사용하는 방법도 있지만, 많은 사용자들은 대기업에서 제공하는 API 를 사용해 대용량의 데이터를 변환한다. 대표적인 대기업의 API 예시는 구글의 Google Cloud OCR, 네이버의 CLOVA OCR 등이 있다.

4. OCR Transformation examples

OCR 은 입력으로 이미지 데이터 및 pdf 파일을 받고, OCR 의 output 은 txt 또는 json 파일이며 예시는 아래와 같다.



52-week High Low Stock Last Change				52-week High Low Stock Last Change			
9.19	6.89	ABX Air n	7.52 -0.10	45.71	32.50	Biomet	36.71 -0.42
33.25	12.40	ACMoore	13.98 -1.57	2.76	1.20	Blomira	1.46 +0.03
31.38	13.51	ADA-ES	20.96 +3.16	9.07	5.13	BloScip	8.05 +0.34
27.14	12.88	ADC Tel rs	23.21 +0.13	68.88	50.65	BloSite	50.05 -4.57
30.40	16.70	ADECo	27.32 +0.73	212.25	131.03	BiotechT	204.66 -0.84
16.45	10.47	AFC Ent s	15.40 -0.14	8.50	1.40	BirchMit an	6.52 -0.45
8.37	4.50	ASE Tst	7.76 +0.40	18.21	10.73	Blackboud	17.90 +0.70
19.25	12.75	ASML Intl	17.65 -0.03	52.73	13.86	BluCoat	41.29 +1.30
20.92	13.94	ASML Hld	21.24 +0.46	44.35	24.15	BlueNile	40.30 -1.10
27.38	16.39	ASV Inc s	26.76 +0.14	26.45	19.91	BobEv n	22.99 ...
19.82	10.47	ATI Tech	17.89 +0.68	15.94	6.12	Bodisen n	15.45 -0.45
33.62	20.53	ATMI Inc	29.95 +1.29	6.21	1.56	Bookham	5.94 -0.01
39.20	16.76	ATP O&G	38.40 -0.59	11.80	4.99	Borland	6.68 +0.14
4.24	1.99	AVI Bio	3.62 -0.02	31.90	21.10	BtPry	31.18 -0.07
				14.68	7.10	BrigExp	12.10 -0.23
				46.72	26.65	BrightHz s	38.90 -0.80

Nasdaq & AMEX									
Stocks in bold rose or fell 5% or more									
4 USA Track your investments with our continuously									
1 X AY updated stocks. Visit us on the web at									
.com money.usatoday.com									
52-week									
High	Low	Stock	Last Change	High	Low	Stock	Last Change		
				45.71	32.50	Biomet	36.71 -0.42		
				2.76	1.20	Blomiro	1.46 +0.03		
				9.07	5.13	BloScrio	8.05 +0.34		
9 19 689	ABX Air n			66.88	50.65	BloSite	50.05 -4.57		
			7.52 -0.10	212.25	131.03	BiotechT	204.66 -0.84		
33.25	12.40	ACMoore	13.98 -1.57	8.50	1.40	Birchht an	6.52 -0.45		
31.36	13.51	ADA-ES	20.96 +3.16	16.21	10.73	Blackboud	17.90		
27.14	12.88	ADC Tel rs	23.21 -0.13				+0.70		
30.40	16.70	ADECo	27.32 -0.73	52 73	13.86	BluCoat	41.29 +1.30		
16.15	10.47	AFC Ent s	15.40 -0.14	44.35	24.15	BlueNile	40.30 -1.10		
5.37 4.50	ASE Tst		7.76 4-0.40	26.45	19.91	BobEv n	22.99 ...		
19 25 12_75	ASM Intl		17.65 -0.03	15.94	6.12	Bodisen n	15.45 -0.45		
				6.21	1.56	Bookham	5.94 -0.01		
20.92 13.94	ASML Hld		21.24 +0.46	11.80	4.99	Borland	6.68 +0.14		
27.38 16.39	ASV Inc s		26.76 +0.14	31.90	21.10	BtPry	31.18 -0.07		
19 82 10.47	ATI Tech		17.89 +0.68	18.62	10.01	131minT	11.53 +0.20		
33.62 20.53	ATMI Inc		29.95 -1.29	14.68	7.10	BrigExp	12.10 -0.23		
39.20 16.76	ATP O&G		38.40 -0.59				-0.80		
4.24 1.99	AVI Bio		3.62 -0.02						
				3.62 -0.02					

< 영수증 OCR 변환 예시 >

Automatic Survey Generation Based on Position Closeness of Key Words

Xiaoping Sun and Hai Zhuge*

Key Lab of Intelligent Information Processing of the Institute of Computing Technology, Chinese Academy of Sciences, China.
The Great Bay University, China.

Abstract: Survey enables researchers to know a research area quickly. Automatic survey generation is a challenge to text summarization research because a survey consists of many sections, each of which is composed by the sentences selected from the papers most relevant to its titles. The key is to rank papers, determine top-k papers and select sentences from the top-k papers according to section title. This paper proposes a measure called key word position closeness to rank papers by measuring how closely two neighboring key words of a section title are distributed within a paper. The basic rationale is that the positions of the neighboring key words of a section title should be closer in more relevant papers to be surveyed. To select the top-k papers for each section, an unsupervised method is proposed to predict the top-k value based on the shape of the curve of the sorted ranking scores. Based on the duality property of the closeness, sentence ranking scores of a selected paper with respect to the key words of section title can be directly obtained when the paper ranking score is computed using the closeness, both the importance and coherence of selected sentences can be reflected without extra sentence ranking calculation. Experiments and manual evaluation show that the

proposed methods achieve significant improvements compared with term frequency-based approaches to generating scientific survey.

Keywords: Text summarization, Natural Language Processing, Text processing, survey generation

I. INTRODUCTION

AUTOMATICALLY generating survey from scientific papers of a research area can help researchers to know the area quickly. A basic approach is to assume that the user of the summarization system provides a general structure of the survey, for example, the titles of sections of the survey, according to which, appropriate papers can be selected and sentences can be extracted from the papers for composing the sections of the survey. Implementing the approach needs to solve the following three closely coupled problems.

(1) Ranking papers according to a section title for selecting papers relevant to the section. Classical term frequency-based

Automatic Survey Generation Based on Position

Closeness of Key Words

Xiaoping Sun and Hai Zhuge*

Key Lab of Intelligent Information Processing of the Institute of Computing Technology, Chinese Academy of Sciences, China.

The Great Bay University, China.

Abstract: Survey enables researchers to know a research area quickly. Automatic survey generation is a challenge to text summarization research because a survey consists of many sections, each of which is composed by the sentences selected from the papers most relevant to its titles. The key is to rank papers, determine top-k papers and select sentences from the top-k papers according to section title. This paper proposes a measure called key word position closeness to rank papers by measuring how closely two neighboring key words of a section title are distributed within a paper. The basic rationale is that the positions of the neighboring key words of a section title should be closer in more relevant papers to be surveyed. To select the top-k papers for each section, an unsupervised method is proposed to predict the top-k value based on the shape of the curve of the sorted ranking scores. Based on the duality property of the closeness, sentence ranking scores of a selected paper with respect to the key words of section title can be directly obtained when the paper ranking score is computed using the closeness, both the importance and coherence of selected sentences can be reflected without extra sentence ranking calculation. Experiments and manual evaluation show that the

proposed methods achieve significant improvements compared with term frequency-based approaches to generating scientific survey.

Keywords: Text summarization, Natural Language Processing, Text processing, survey generation

I. INTRODUCTION

AUTOMATICALLY generating survey from scientific papers of a research area can help researchers to know the area quickly. A basic approach is to assume that the user of the summarization system provides a general structure of the survey, for example, the titles of sections of the survey, according to which, appropriate papers can be selected and sentences can be extracted from the papers for composing the sections of the survey. Implementing the approach needs to solve the following three closely coupled problems.

(1) Ranking papers according to a section title for selecting

papers relevant to the section. Classical term frequency-based

< 논문 OCR 변환 예시 >

Keyword Extraction

Step 1. Pre-processing

문서 전처리 과정은 sentence segmentation, word tokenization, Part-of-Speech (PoS) tagging 의 세 단계로 구성된다. Sentence segmentation 을 통해 문서를 문장 단위로 분할하고 word tokenization 을 통해 문장을 단어 단위로 분할한다. 분할된 단어 각각에 PoS tag 를 사용해 각 word 의 품사가 명사인지, 형용사인지에 대한 정보를 추가한다.

Step 2. Candidate extraction

일반적인 경우 keyword 의 품사는 명사 혹은 형용사와 명사의 sequence 로 구성 되어있고, keyword 의 길이는 두 단어 이상이라는 기존 연구 결과에 따라 형용사와 명사로 이루어진 가장 긴 길이의 phrase 로 candidate keyword set 을 구성한다. candidate keyword 간의 중복을 줄이기 위해 각 candidate 에 대해 stemming 을 적용한다.

Step 3. Candidate clustering

Candidate keyword 들에 대해서 Hierarchical Agglomerative Clustering (HAC) algorithm 을 사용해 자동적으로 cluster 를 구성하고, cluster 내의 maximum distance 를 설정하여 두 keyword 가 얼마나 비슷해야 같은 cluster 로 간주할 것인지 정한다. (관련 논문에서는 적어도 25%가 중복되어야 같다고 판단했다.)

HAC algorithm 으로 clustering 을 진행할 때, distance 는 각 keyword 의 term frequency vector 간 거리를 사용하고, cluster 와 keyword 의 거리는 average, single, complete 등의 다양한 방법이 있지만 single 과 complete 의 타협점인 average linkage 를 사용한다.

* term frequency vector: $(\# \text{ of occurrence of candidate } i \text{ in context } c) \in \mathbb{R}^{(\# \text{ of context})}$

context는 문장, 문단, 문서, 도메인으로 occurrence를 판단할 window를 의미한다.

다른 방법으로는

Step 4. Graph construction

Step 3 의 clustering 의 각 cluster 를 topic 이라고 하고 이를 graph $G = (V, E)$ 의 vertex 로 설정한다. Edge 에 대응하는 두 topic 간의 weight 는 다음과 같이 구한다.

$$w_{i,j} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} \text{dist}(c_i, c_j)$$
$$\text{dist}(c_i, c_j) = \sum_{p_i \in \text{pos}(c_i)} \sum_{p_j \in \text{pos}(c_j)} \frac{1}{|p_i - p_j|}$$

t_i, t_j 는 topic 이고, c_i, c_j 는 각 topic 에 포함된 candidate keyword, $\text{pos}(c_i), \text{pos}(c_j)$ 는 각 candidate 의 등장 위치를 의미한다.

Step 5. Graph-based Ranking

먼저 각 topic 에 부여되는 score 를 같은 값으로 initialize 하고, edge 의 weight 에 기반하여 converge 할 때까지 아래의 계산을 반복한다.

$$S(t_i) = (1 - \lambda) + \lambda \times \sum_{t_j \in V_i} \frac{w_{j,i} \times S(t_j)}{\sum_{t_k \in V_j} w_{j,k}}$$

V_i 는 t_i 로 연결된 topic 들의 set 이고, λ 는 damping factor 로 한 topic 에서 다른 topic 으로 연결될 확률을 의미한다.

Step 6. Keyword Selection

Step 5 에서 각 topic 에 부여된 점수로 top-k 개의 topic 을 선택할 수 있고 선택된 topic 내에서 keyword 를 선택하는 방법은 세 가지가 제시되었다. Candidate keyword 의 position 에

기반하여 가장 먼저 등장한 candidate 를 선택하거나, frequency 에 기반하여 가장 많이 등장한 candidate 를 선택하거나, cluster 의 특징을 반영하여 centroid 에 가까운 candidate 를 선택하는 방법을 사용할 수 있다.

해당 연구에서는 여러 view point 에서의 서로 다른 기준으로 keyword 를 추출할 계획이다. 논문 하나에 대해서, 저자가 작성한 keyword 나 abstract, introduction, conclusion 과 같은 특수 section 을 기준으로 introduction 에서 더 중요한 keyword 와 같이 다양한 관점에서 keyword 를 얻을 수 있다. 저자가 작성한 keyword 의 경우는 OCR 에서 논문의 feature 를 구하는 과정에서 구할 수 있다. 특정 section 의 keyword 의 경우, 앞서 소개한 method 를 사용하지만 document collection(corpus)를 전체 논문 셋이 아닌 논문 하나로 생각하고, 그 corpus 내에서 keyword 를 추출해야 하는 document 를 section 으로 설정하여 얻을 수 있다.

Academic document 의 특성상 수학, 물리학, 언어학 등 다양한 큰 주제(domain)가 있고 그 아래에 다양한 작은 단위의 주제가 존재한다. 예를 들어, 수학이라는 domain 에서는 일반적으로 사용되는 단어가 언어학의 domain 에서는 거의 사용되지 않는 경우에, 이 단어가 수학 domain 의 한 문서의 keyword 로 설정된다면 domain-level 에서는 구분할 수 있지만 같은 domain 내의 다른 논문들과 변별력은 줄어들게 된다. 이를 해결하기 위해 domain 관점에서의 keyword 와 논문 관점에서의 keyword 를 구분할 수 있다. Domain-level 의 keyword 는 corpus 를 전체 논문 셋으로, keyword 추출 단위를 domain 으로 설정하여 얻을 수 있고, 한 domain 내 document 의 keyword 는 corpus 를 그 domain 에 포함되는 논문 셋으로 설정하고, keyword 추출 단위를 논문으로 설정하여 구할 수 있다.

쿼리와 사전 생성된 Feature 를 이용한 Scoring 및 요약 생성

Step 1. 후보군 생성

DB에 저장되어 있는 논문은 40만개 이상이다. 검색어(쿼리)가 들어올 때 마다 모든 논문에 대해 연관성 순위를 매기는 것은 비효율적이다. 따라서 top-k 논문을 선정하기 이전에 연관성이 없거나 적은 논문들을 후보군에서 제외하는 것이 효율적이다.

각 논문에 대해서 서지정보&본문 외에도, 추출한 Feature(논문 제목, 섹션 제목, 초록 전문, Keyword(n-gram 형태로 제공됨))등이 DB에 미리 저장되어 있으므로 이들을 이용하면 아래와 같은 과정을 통해 관련된 논문만 후보군에 추가하는 것이 가능하다.

1. 사용자가 n-gram 형태의 쿼리(예: 사회과학에서 네트워크 이론의 응용)을 제시한다. 이 쿼리에서 조사 등 문법적 요소를 제거하고 여러 단어로 이루어진 전문 용어가 있다면 클러스터링 한다. 이를 쿼리 (예: 사회과학, 네트워크 이론, 응용)
2. 논문의 개수가 많기 때문에 본문은 step1. 에서 확인할 수 없다. 따라서 해당 쿼리의 키워드가 모두 Feature 에 나오는 논문을 선택한다.
3. 만약 선정된 논문이 부족하다면, 쿼리의 키워드 중 한 개까지 연관 검색어로 대체하는 것을 허용한다. 연관검색어는 외부사전& 단어 그래프를 이용한 쿼리확장 기법을 사용한다
4. 3 번을 실행한 이후에도 선정된 논문이 부족하다면 핵심단어중 하나를 제외하는 것을 허용한다.

이 과정이 마무리 되면 검색어와 관련이 있는 논문만 다음 단계에서 고려 대상이 된다.

Step 2. 논문 Scoring

논문의 저자가 말하고자 하는 바와 내용은 초록에 요약되어 있으므로 초록을 이용하여 Top-K 논문을 선정하는 것은 합리적이다. Top-k 논문의 선정은 후보군의 각 논문에 점수를 매기는 것으로 진행된다.

각 논문의 점수는 다음과 같이 제목, 키워드, 초록의 내용을 이용하여 계산한다.

$$\text{논문 점수} = 0.3 * \text{제목 점수} + 0.4 * \text{키워드 점수} + 0.3 * \text{초록점수}$$

- 제목 점수 = 0.5*논문 제목 점수 + 0.5*섹션 제목 점수
논문제목 점수 & 섹션 제목 점수

검색어 $Q = [q_1, \dots, q_n]$ 와 문장 $S = [w_1, \dots, w_m]$ (q, w 는 형태소)의 코사인 유사도는 다음과 같은 과정을 통해 구한다.

형태소 목록 $M_{Q,S} = \{m | m \in (Q, S)\} = \{z_1, \dots, z_o\}$

Q 의 빈도 벡터 표현 $V_Q = [v_{Q,1}, \dots, v_{Q,o}]$, $v_{Q,i} = \# \text{ of } z_k \text{ in } Q$ (S 도 같은 방식으로 벡터화

Q 와 S 의 코사인 유사도 $CS(Q, S) = \frac{V_Q \cdot V_S}{\|V_Q\| \|V_S\|}$

이를 이용하여 제목 점수를 다음과 같이 정의한다.

DT = 논문의 제목, ST_1, ST_2, \dots, ST_S = 섹션 제목, Q = 검색어

논문 제목 점수 = $CS(Q, DT)$

각 섹션의 섹션제목 부분점수 = $CS(Q, ST_s)$

섹션 제목 점수 = $\frac{1}{S} \sum_{s=1}^S CS(Q, ST_s)$

제목 점수 = 0.5*논문 제목 점수 + 0.5*섹션 제목 점수

키워드 점수

키워드 점수는 쿼리와 각각의 키워드 사이에 일치도를 구하고 그 중 최댓값을 논문의 키워드 점수로 한다.

쿼리 $Q = [q_1, q_2, \dots, q_N]$ 과 키워드 $K = [k_1, \dots, k_M]$ 에 대해

$A_n = \{q_n \text{의 외부사전을 이용한 연관검색어}\}$, $Q' = (\cup_{n=1}^N A_n) - Q = (Q \text{의 확장})$

$B_m = \{k_m \text{의 외부사전을 이용한 연관검색어}\}$, $K' = (\cup_{m=1}^M B_m) - K = (K \text{의 확장})$

K 의 정방향 키워드 유사도 $KS(Q, K) = \frac{(|Q \cap K| + 0.5 \sum_{n=1}^N |\{q_n\} \cap K'|)}{M}$

K 의 역방향 키워드 유사도 $KS(K, Q) = \frac{(|K \cap Q| + 0.5 \sum_{m=1}^M |\{k_m\} \cap Q'|)}{N}$

K 의 대칭 키워드 유사도 $KS_{sym}(Q, K) = \frac{KS(Q, K) + KS(K, Q)}{2}$

논문 D 의 키워드가 K_1, K_2, \dots, K_L 일 때

D 의 키워드 점수 = $\max_{l \in \{1, 2, \dots, L\}} KS_{sym}(Q, K_l)$

초록 점수

초록점수는 초록 내 문장들의 평균적인 코사인 유사도로 정의된다

초록 $Abs = [abs_1, abs_2, \dots, abs_L]$, abs_l = 초록의 l 번째 문장

초록 점수 = $\frac{1}{L} \sum_{l=1}^L CS(Q, ABS_l)$

Top - k 선정

해당 알고리즘을 통해 각 논문 D_1, D_2, \dots, D_P 에 점수 R_1, R_2, \dots, R_P 가 부여되었다. (인덱스는 점수 내 랭크순으로 정렬했을 때 기준)

$p = 2, 3, \dots, P-1$ 에 대해 유사 미분계수 $\tilde{R}_p^{(1)} = \frac{R_{p-1} - R_{p+1}}{2}$

$p = 3, 4, \dots, P-2$ 에 대해 유사 이계 미분계수 $\tilde{R}_p^{(2)} = \frac{\tilde{R}_{p-1}^{(1)} - \tilde{R}_{p+1}^{(1)}}{2}$

$\tilde{R}_k^{(2)} \geq 0, \tilde{R}_{k+1}^{(2)} < 0, k \geq 5$ 인 k 중 최솟값을 top-k의 파라미터 k 로 사용한다. (만약 이러한 k 가 없다면 $k = \min(5, P)$ 를 사용한다.

Step3. 문장 scoring & 요약생성

요약은 다음과 같은 과정으로 생산된다.

1. 허용된 요약의 문장 수에 대해 top-k 논문에서 논문 점수에 비례하도록 문장을 배분한다.

2. 각 논문에서 문장/섹션 별로 점수를 매긴다.
3. 섹션 점수에 비례하여 각 논문에 배분된 문장을 다시 섹션별로 배분하고 (이때 사용자의 하이퍼 파라미터를 통해 특정 섹션에 가중치를 더 두도록 설정할 수 있다.) 각 섹션마다 배분된 문장은 문장점수가 높은 순서대로 선정한다.
4. 이를 모아서 요약을 형성한다. 요약에 사용되는 문장의 순서는 원문의 순서를 유지한다.
5. 요약에서 figure 를 인용한다면 figure 또한 요약에 첨부된다.

논문에서 문장별로 그리고 섹션별로 점수를 매기는 과정은 다음과 같다.

- a. 문장 점수 = 0.5*검색어 문장점수 + 0.5*섹션 제목 문장점수
- b. 쿼리 혹은 섹션제목에서 형태소 단위로 분리 조사들을 제거하여 키워드 목록 $K = [k_1, k_2, \dots, k_T]$ 를 구한다.
- c. 각 k_t 에 대해 위치벡터 $v_t = [v_{1,t}, v_{2,t}, \dots, v_{N_t,t}]$ 를 구한다 이때 $v_{n,t}$ 는 k_t 가 n 번째로 나타난 문장의 번호를 의미한다.

- d. v_t, v_{t+1} 에 의한 문장 s(문장의 인덱스를 의미)의 점수

$$r(s|v_t, v_{t+1}) = \sum_{i=1}^{N_t} \left[\frac{\delta(s-v_{i,t})}{|v_{i,t}-M(v_{i,t}|v_{t+1})|+1} \right] + \sum_{i=1}^{N_{t+1}} \left[\frac{\delta(s-M(v_{i,t}|v_{t+1}))}{|v_{i,t}-M(v_{i,t}|v_{t+1})|+1} \right]$$

이때 $M(v_{i,t}|v_{t+1}) = (v_{t+1} \text{의 원소중 } v_{i,t} \text{와 가장 가까운 값})$

$\delta(s-x) = (1 \text{ if } s=x, \text{ else } 0)$

- e. 키워드 목록 K에 대한 문장 s의 점수 $SR(K, s) = \frac{1}{T-1} \sum_{t=1}^{T-1} r(s|k_t, k_{t-1})$
- f. 검색어 Q에 대응하는 키워드 목록 K_Q 섹션 $P = [s_1, \dots, s_{N_P}]$ 섹션제목 T_P 에 대응하는 키워드 목록 K_{T_P} 에 대해

$$\text{문장 } s_i \in P \text{의 최종점수 } SR(s_i) = \frac{(SR(K_Q, s_i) + SR(K_{T_P}, s_i))}{2}$$

$$\text{섹션 P의 점수 } PR_P = \frac{1}{N_P} \sum_{i=1}^{N_P} SR(s_i)$$