의미적으로 확장된 문장 간 유사도를 이용한 한국어 텍스트 자동 요약

김희찬, 이수원
숭실대학교 컴퓨터학과
e-mail:eriz0xt@gmail.co.kr

Korean Text Automatic Summarization using Semantically Expanded Sentence Similarity

Heechan Kim, Soowon Lee
Dept. of Computer Science, Soongsil University

요 약

텍스트 자동 요약은 수많은 텍스트 데이터를 처리함에 있어 중요한 연구 분야이다. 이중 추출요약은 현재 가장 많이 연구가 되고 있는 자동 요약 분야이다. 본 논문은 추출 요약의 선두 연구인 TextRank는 문장 간 유사도를 계산할 때 문장 내 단어 간의 의미적 유사성을 충분히 고려하지 못하였다. 본 연구에서는 의미적 유사성을 고려한 새로운 단어 간 유사도 측정 방법을 제안한다. 추출된 문장 간 유사도는 그래프로 표현되며, TextRank의 랭킹 알고리즘과 동일한 랭킹 알고리즘을 사용하여 실험적으로 평가하였다. 그 결과 문장 간 유사성을 고려할 때 단어의 의미적 요소를 충분히 고려하여 정보의 유실을 최소화하여야 한다는 것을 실험 결과로써 확인할 수 있었다.

1. 서론

텍스트는 인간의 삶에 매우 많은 부분을 차지한다. 21세기에 들어서 인터넷의 세상이 도래하고, 누구나 어디에서나 인터넷에 접속할 수 있는 기기를 가지고 다님에 따라 여가 시간에 텍스트를 접하는 시간이 더욱 더 늘어났다. 이러한 텍스트의 범람 속에서 사용자에게 가장 흥미로운 글이 무엇일지 제목만 보고 판단하는 것은 상당히 어려운 일이다. 누군가가 그 글들의 줄거리를 요약해준다면 흥미로운 글을 찾는데 들이는 시간이나 읽어보니 흥미롭지 않은 글일 수도 있는 위험성을 줄일 수 있을 것이다.

본 논문에서는 문서 자동 요약(Automatic Summarization) 중 자율 학습(Unsupervised Learning) 방법의 추출 요약(Extractive Summarization)에 대한 새로운 연구 방법을 제시한다. 추출 요약에서의 첨단 기술인 TextRank[1]는 문장 간 유사도를 계산할 때 문장 내 단어 간의 의미적 유사성을 충분히 고려하지 못한다는 문제점이 있다. 이러한 문제점을 해결하기 위하여, 본 연구에서는 TextRank에서 충분히 고려하지 못한 문장 간 의미적 유사성을 고려하기 위하여 문장 내 동시 출현 단어와 그 단어의 유의어 관계를 이용한다.

추출 요약 분야의 관련 연구에 대해서는 2장에서 다루고, 이를 해결하는 방법을 3장에서 제안한다. 4장에서는 기존 연구의 방법과 본 논문에서 제시하는 방법을 실험을 통하여 비교하고, 마지막 5장에서 본 연구의 결론 및 향후 연구를 기술한다.

2. 관련 연구

추출 요약 관련 연구들은 문서에서 등장한 문장들 중 가장 영향력이 있는 상위 N개의 문장을 추출하는 방법인 랭킹 알고리즘에 초점을 맞추고 있다. 대부분의 랭킹 알고리즘은 문장 간의 관계를 이용하여 가장 영향력 있는 문장을 추출하는 것이기 때문에 그래프 방식의 접근 방법을 많이 택한다.

그 중 사람들에게 가장 많이 알려진 랭킹 알고리즘인 PageRank[2]를 추출 요약에 접목시킨 TextRank는 특정 문서의 각 문장들을 하나의 노드로, 문장 간 유사도를 엣지로 표현하여 문서를 그래프화하고, 이 그래프로부터 가장 중요한 문장들을 선택한다. TextRank에서 사용하는 PageRank 알고리즘은 기존 PageRank의 랭킹 알고리즘에 엣지의 가중치를 반영하여 노드의 영향력을 계산한다.

TextRank에서는 두 문장 간에 어휘적으로 동일한 단어를 이용한 관계에 초점이 맞추어져 있어 의미적 관계가 고려되지 않는 경향이 있다 ([식 1]).

$$milarity_{xtRank}(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \quad w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)}$$

[식 1] TextRank에서의 문장 간 유사도

문서를 그래프 형태로 변환할 때 TextRank와 같이 두 문장 간에 어휘적으로 동일한 단어들의 수로 유사성을 계산하여 변환하면 문서에 포함된 의미적 요소들이 무시되

기 때문에 문장 간 정보를 온전히 담지 못한다고 할 수 있다. 이러한 의미적 정보의 누락을 해결하기 위한 방법으로 WordNet을 활용한 TakeLab[3]에서 제안한 방법이 있다([식 2]).

$$_N\ S_1, S_2) = \frac{1}{|S_2|} \sum_{w \in S_1} score(w_1, S_2)$$

$$score(w, S) = \begin{cases} 1 & \text{if } w \in S \\ \max_{w' \in S_1} sim(w, w') & otherwise \end{cases}$$

[식 2] WordNet을 이용한 TakeLab의 단어 간 유사도

[식 2]에서 sim(w, w')는 WordNet에서 두 단어 사이의 최단 경로 길이이다. 이 방법은 두 단어가 동일한 단어는 아니지만, WordNet 상에서 두 단어의 의미적 거리를 두 단어의 유사도로 표현하여 동일한 단어가 아니더라도 유사성을 계산할 수 있다. TakeLab은 WordNet의 유의어 관계를 이용하여 의미적 관계를 고려하였지만, 입력된 특정 문서의 주제에 대한 의미적 관계는 WordNet의 일반적인 유의어 단어 관계에서는 확인하기 어렵기 때문에, 해당 문서의 주제와 관련된 정보는 고려되지 않는다고 볼 수 있다.

3. 제안 방법

TakeLab에서 제시한 방법은 WordNet과 같이 일반적인 상황에서의 유의어 단어 관계를 이용하므로 입력된 문서의 주제에서만 성립되는 유의어 간 유사성을 충분히 고려할 수 없다는 점이다.

예를 들어, 두 문장 S_1과 S_2가 다음과 같다고 가정하자[4].

S_1="의존 관계를 가지는 어절은 다음 단계에서는 삭제되어 더 이상 다른 어절의 의존구조에 영향을 미치지 못한다."

S_2="성능 측정은 의존구조와 의존 관계명이 모두 일치하는 경우만 정답으로 하였다."

이 두 문장을 단어 차원의 벡터로 표현하면 [표 1]과 같다.

[표 1] 예시로 든 두 문장의 문장 벡터

	의존	관계	가지다	어절	다음	단계	삭제되다
S_1	1	1	1	2	1	1	1
S_2	1	0	0	0	0	0	0
	이상	다르다	의존구조	영향	미치다	못하다	성능
S_1	1	1	1	1	1	1	0
S_2	0	0	1	0	0	0	1
	측정	관계명	모두	정답	일치하다	하다	
S_1	0	0	0	0	0	0	
S_2	1	1	1	1	1	1	

[표 1]의 두 문장 벡터를 보면 같은 동일한 단어가 많지 않기 때문에 $Similarity_{cosine}(S_1,S_2)$로 계산을 하면 유사도가 0.167로 유사도가 높지 않다. 그러나 위 두 문장 벡터의 차원을 보면 서로 자주 사용되는 단어들이기 때문에 문장 간 유사도는 이를 고려하여 이보다 높은 수치로 추출되어야한다.

본 연구에서는 다음과 같은 점에 착안하여 단어 간 유사도를 새로이 정의함으로써 이러한 문제를 해결한다.

첫째, 한 문장 내에 같이 등장하는 단어들은 상호 연관 관계가 존재한다.

둘째, 특정 단어의 유의어들은 같은 의미는 아니지만 상호 유사한 의미를 가진다.

이 두 가지 가설에 따라 새롭게 정의된 문장 벡터를 통해 문서의 요약을 추출하는 방법은 [그림 1]과 같다. 먼저 두 가지 가설을 이용하여 단어 간 유사도를 계산하기 위한 단어 동시 출현 관계 그래프G_{word}를 생성하고, 단어의 유의어를 추출하여 G_{word}에 추가한다. 유의어로 확장된 G_{word}를 이용하여 문장 관계 그래프 $G_{sentence}$를 생성한 후 $G_{sentence}$에 랭크 알고리즘을 적용하여 중요 문장을 추출하는 것으로 추출 요약을 진행한다.

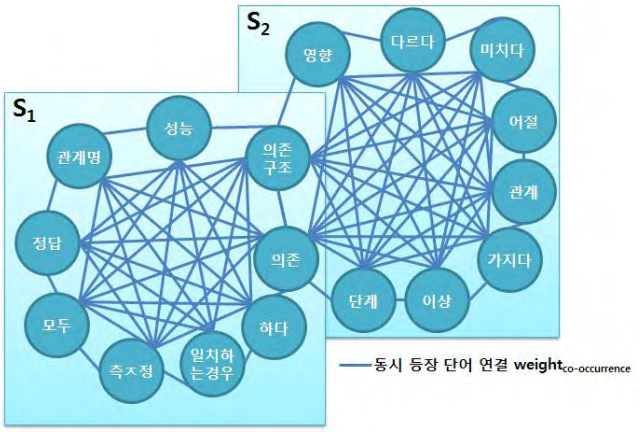
[그림 1] 제안 요약 시스템 구조도

3-1. 단어 동시 출현 관계 그래프 생성

단어 동시 출현 관계 그래프 생성 단계는 요약할 문서의 특정 문서 내의 단어 간 유사성을 고려하는 단계와 단어의 유의어와의 유사성을 고려하기 위한 단계로 구성된다.

먼저 문서 D를 문장으로 나누는데, 각 문장 별로 형태소 분석을 한 후 명사, 형용사, 동사만을 추출하여 단어 w_i로 정의한다. 이때 사용한 형태소 분석기는 창원대 형태소 분석기를 사용하였다. 다음 각 문장에서 추출한 단어를 노드로 하고 문장 내에서 동시 출현 관계를 나타내는 가중치 $weight_{co\text{-}occurrence}$를 가지는 엣지로 단어 간 관계를 표현한다. 이때 생성한 그래프를 단어 동시 출현 관계 그래프 G_{word}라 정의한다.

앞서 예시로 들었던 문장들로 단어 동시 출현 관계 그래프를 생성하면 [그림 2]와 같은 그래프 형태를 이루게 된다.

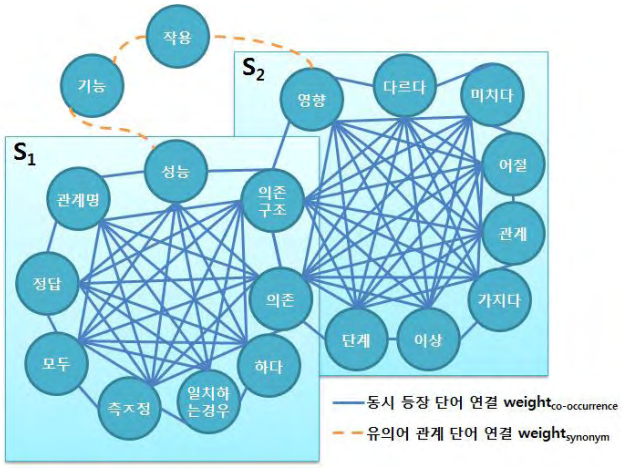


[그림 2] 동시 출현 단어를 표현한 G_{word} 예시

3-2. 단어의 유의어 추출

이 단계에서는 G_{word}에 등장하는 모든 단어에 대해 유의어를 검색하여 추가한다. 이때 추가된 유의어의 유의어 또한 추가하는데 유의어 관계가 계속해서 연속될 경우 실제 의미가 전혀 다른 경우가 많기 때문에 본 논문에서는 깊이를 3까지만 하여 유의어를 추가한다. 특정 단어를 나타내는 노드와 유의어로 추가되는 노드와는 유의정도를 나타내는 가중치 $weight_{synonym}$를 가지는 엣지로 연결한다. 이때 사용한 유의어 정보는 네이버 국어사전에 ㈜낱말이 제공하는 유의어 단어 정보를 이용하였다.

[그림 2]의 G_{word}에 유의어를 추가하면 [그림 3]과 같은 그래프 형태를 이루게 된다. [그림 3]은 S_1과 S_2사이의 연관성이 있는 유의어만 도식화 한 것으로 실제 그래프는 이 보다 더 노드가 많은 방대한 그래프이다.



[그림 3] 유의어를 추가한 G_{word} 예시

위 두 단계를 통해 G_{word}을 완성하고 이 그래프 내에서 두 단어 간의 유사도를 계산하게 된다. 두 단어 간 유사도를 계산하는 방법은 G_{word} 내의 두 단어를 표현하고 있는 노드 간의 Shortest Path를 이용하여 계산한다. 그러나 G_{word}의 엣지의 가중치는 단어 사이의 유사도를 나타내는 가중치이기 때문에 거리로 정의할 수가 없다. 따라서 최단 경로를 구하기 위해 G_{word}에서의 노드 간 거리는 1 - weight로 정의한다.

두 단어 w_e와 w_n의 의미적 유사도 $Sim_{word}(w_i, w_j)$은 G_{word} 내에서 두 단어를 나타내는 노드 간의 최단 경로를 구한 후 엣지 e_k의 가중치들을 곱하여 계산하며 다음과 같이 식으로 표현할 수 있다.

$$m_{ord}(w_i, w_j) = \prod_{\substack{e \in hortestPath(w_i, w_j) \\ e_k \in E(G_{word})}} weight(e_k)$$

[식 3] 두 단어 간의 의미적 유사도

예를 들어 실험 데이터로 사용된 입력으로부터 생성한 G_{word}에서 가중치 $weight_{co\text{-}occurrence}$와 $weight_{synonym}$가 각각 0.2와 0.7이라고 가정할 때 특정 두 단어의 의미적 유사도는 아래와 같다.

Sim_{word}(관계명, 성능) = 0.2
Sim_{word}(성능, 기능) = 0.7
Sim_{word}(성능, 영향) = 0.7 * 0.7 * 0.7 = 0.343

3-3. 문장 간 관계 그래프 생성

문서 간 관계 그래프 생성 단계에서는 문서를 그래프화하는 단계로, 앞서 정의한 단어 간 유사도를 문장 간 유사도에 적용하여 엣지를 표현함으로써 문서의 의미적 정보의 손실을 최소화하며 그래프화한다.

문서에서 가장 영향력 있는 문장을 추출하기 위해 생성하는 그래프는 각 문장 S_i를 노드로 하며, 문장 그래프 $G_{sentence}$라고 칭한다. 이때 노드 간의 엣지는 [식 4]의 문장 간 유사도를 이용하여 엣지를 표현한다.

본 연구에서는 문장 간 유사도를 TextRank에서의 $Similarity_{TextRank}$와 다르게 코사인 유사도 $Similarity_{cosine}$를 이용하여 계산한다.

$$Similarity_{cosine}(S_i, S_j) = \frac{\vec{S_i}(S_j) \cdot \vec{S_j}(S_i)}{\|\vec{S_i}(S_j)\| \ \|\vec{S_j}(S_i)\|}$$

[식 4] 두 문장 간의 코사인 유사도

[식 3]에서 정의한 단어 간 유사도를 문장 벡터에 반영하기 위해 문장 벡터를 벡터 함수로 새로 정의하였다. 두 문장 S_i와 S_j의 유사도 계산을 위한 벡터 함수 (S_j)는 두 단계로 계산된다. 먼저 벡터 함수의 반환된 벡터의 단어 차원은 S_i와 S_j에 등장한 단어들의 합집합의 크기 n개의 차원을 가지고, 각 단어 차원에 S_i에 등장한 단어의 등장 빈도를 채워 넣는다. 두 번째로 등장 빈도를 채우고 난 벡터에서 값이 0으로 채워진 차원의 단어와 S_i에 출현한 단어와 유사도를 계산하여 유사도의 평균을 구하고 문장의 길이로 표준화한 값을 해당 차원의 값으로 지정한다.

위에서 설명한 벡터함수 $\vec{s_i}(s_j)$를 만드는 방법은 아래의 식으로 표현할 수 있다.

$$\vec{s_i}(S_j) = (|\ w_1|w_1 \in S_i\}|,\ \ldots\ ,|\{w_n|w_n \in S_i\}|) + \frac{1}{|S_i| - |S_i \cap S_j|}\Big(\sum_{\substack{w \in i,\\ w_1 \notin S_i,\\ w_1 \in S_j}} |\ w_1|w_1 \in S_j\}| \cdot Sim_{word}(w_1, w_k), \ldots, \sum_{\substack{w_k \in S_i,\\ w_n \notin S_i,\\ w_n \in S_j}} |\{w_n|w_n \in S_j\}| \cdot Sim_{word}(w_n, w_k)\Big)$$

[식 5] 두 문장 사이의 단어 간 유사도를 적용하기 위한 벡터 함수

예시에 적용한 가중치 $weight_{co\text{-}occurrence}$와 $weight_{synonym}$를 각각 0.2와 0.7이라 할 때 [식 5]로 처음 예시로 들었던 두 문장 S_1, 및 S_2의 문장 벡터를 구하면 아래 표와 같고, 그 둘의 $Simialrity_{cosine}(S_1,S_2)$는 0.295가 된다.

[표 2] 제안 방법을 적용한 두 문장의 문장 벡터

	의존	관계	가지다	어절	다음	단계	삭제되다
(S_2)	1	1	1	2	1	1	1
$\vec{S_2}(S_1)$	1	0.109	0.063	0.263	0.063	0.109	0.04
	이상	다르다	의존구조	영향	미치다	못하다	성능
$\vec{S_1}(S_2)$	1	1	1	1	1	1	0.069
$\vec{S_2}(S_1)$	0.04	0.04	1	0.04	0.04	0.04	1
	측정	관계명	모두	정답	일치하다	하다	
$\vec{S_1}(S_2)$	0.040	0.113	0.069	0.040	0.055	0.069	
$\vec{S_2}(S_1)$	1	1	1	1	1	1	

3-4. 랭킹 알고리즘

TextRank에서 사용하는 랭킹 알고리즘은 PageRank의 랭킹 알고리즘에 문장 간의 유사성, 즉 노드 간의 엣지의 가중치를 반영하여 해당 문장의 랭크 점수를 계산한다[식 6].

$$WS(V_i) = (1-d) + d^* \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} wk} WS(V_j)$$

[식 6] TextRank의 랭킹 알고리즘

문장 그래프$G_{sentence}$에 [식 6]의 알고리즘을 적용한 후 문장들의 랭크 점수를 내림차순 정렬하여 상위 N개의 문장을 뽑아 문서 D의 요약으로 제시한다.

4. 실험 및 결과

실험은 정답 셋을 비교적 명확히 할 수 있는 '논문'을 문서 셋으로 하고 입력된 논문의 제목과 요약 부분이 논문을 요약한 결과라고 가정한 후 진행하였다.

본 연구에서 제시하는 방법으로 추출한 상위 10개의 문장 중에 입력된 논문의 제목과 요약의 문장이 얼마나 추출되는지를 측정하여 F-measure로 평가하였다. 사용한 논문은 정보과학회 논문 중 최신 논문 중 시간 순서로 내림차순 정렬 후 상위 30개의 논문을 데이터로 사용하였다. 실험에서 사용한 $weight_{co\text{-}occurrence}$, $weight_{synonym}$은 본 논문에서 예시로 들었던 0.2, 0.7이다.

TextRank의 문장 간 유사도를 사용한 실험에서는 0.152를 기록하였고, 본 연구에서 제시한 유사도를 사용한 실험에서는 0.208를 기록하였다. 이는 문장 간 유사도를 계산할 때 문장의 정보를 충분히 추출하는 것이 문서를 그래프화할 때 중요한 점임을 다시 한 번 시사한다.

5. 결론

본 연구에서 제안한 의미적으로 확장된 문장 간 유사도 측정 방법은 간단한 형태의 문장 간 유사도 계산 방식을 가진 TextRank와의 비교 실험을 통하여 랭킹 알고리즘도 중요하지만, 문서를 의미의 누락 없이 그래프화하는 것 또한 중요하다는 결과를 얻을 수 있었다.

그러나 제안 방법은 최단 경로 검색과 코사인 유사도 계산과 같이 계산복잡도가 높은 방법을 사용하므로 기존 TextRank에 비해서 매우 느리기 때문에 더 빠르지만 비슷한 효력을 가지는 다른 Sentence Similarity Measure에 대한 연구가 필요하다.

Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업[13-912-03-003, 사회문제에 관한 도메인 별 이벤트 추출 및 예측 기술 개발], 중소기업청에서 지원하는 2014년도 산학연협력 기술개발사업(No. C0221419) 및 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2014030008).

참고문헌

[1] Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. In Proceedings of EMNLP (Vol. 4, No. 4, p. 275)

[2] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web.

[3] Šarić, F., Glavaš, G., Karan, M., Šnajder, J., & Bašić, B. D. (2012, June). Takelab: Systems for measuring semantic text similarity. In Proceedings of SemEval-2012, pages 441 - 448, Montreal, Canada.

[4] 최맹식, 정석원, & 김학수. (2014). CRFs를 이용한 의존구조 분석 및 의존 관계명 부착. 정보과학회논문지: 소프트웨어 및 응용, 41(4), 302-308.