

감정 기반 블로그 문서 분류를 위한 부정어 처리 및 단어 가중치 적용 기법의 효과에 대한 연구*

정 유 철 최 윤 정 맹 성 현[†]

한국정보통신대학교 공학부

일상생활에서 많이 쓰이는 블로그 문서를 분석하는 것은 다양한 웹 응용서비스를 연결할 수 있는 중요한 단초를 제시하므로, 블로그 문서에 담긴 감정을 파악하는 것을 매우 유용한 일이다. 본 논문에서는 블로그 문서에 존재하는 감정을 보다 정확하게 분류하기 위해 부정어 처리와 새로운 단어 가중치의 적용이 성능에 미치는 영향에 대해 탐구한다. 특히, 감정 단서(clue)가 내재된 정규화된 부정어 n-gram을 통해 부정어 처리를 고도화하고 말뭉치기반 단어 가중치 계산법(Corpus-specific Term Weighting, CSTW)을 통해 감정 분류 성능향상을 살펴보기로 한다. 검증을 위해 블로그 문서들로 정답 말뭉치를 구축하고 감정 흐름 분석(Enhanced Mood Flow Analysis, EMFA)과 지지벡터기계기반 감정 분류(Support Vector Machine based Mood Classification, SVMCM)의 두 가지 분류기법에 대해 실험을 하였다. 정규화된 부정어 n-gram의 적용은 EMFA에서 점진적인 감정 분류 성능 향상을 보여주었으며, CSTW의 적용은 TF*IDF나 TF에 비해 보다 높은 감정 분류 성능을 나타내었다.

주제어 : 감정 분류, 정규화된 부정어 n-gram, 단어 가중치 계산법, 감정흐름분석, 지지벡터기계기반 감정 분류

* 본 연구는 지식경제부 및 정보통신연구진흥원의 IT핵심기술개발사업의 일환으로 수행 [2008-F-047-01, Urban Computing Middleware 기술 개발] 하고, 과학기술부(K20711000007-07A0100-00710) 및 한국과학기술협력재단의 지원을 받았으며, 2008년도 2단계 BK 21 사업의 일환으로 수행되었습니다.

† 교신저자: 맹성현, 한국정보통신대학교 공학부 교수, 연구세부분야: 정보검색, 자연어처리, 오피니언 마이닝, Email: myaeng@icu.ac.kr

서 론

블로그 문서에는 작성자의 감정이 내재되어 있는 경우가 있어 이러한 감정을 기반으로 블로그 문서를 분류하는 것은 웹과 관련된 다양한 추천서비스와 연계하여 활용될 수 있어 최근 활발한 연구가 이루어지고 있다. 예를 들어 다양한 언어 자질들(단어 빈도, 문서 길이, PMI-IR 등)을 기반으로 지지벡터기계(SVM) 기법을 활용하여 블로그 문서의 감정 분류를 시도하는 연구가 있었다[1]. 그에 이어, 블로그 문서의 복잡하고 다양한 특성을 파악하기 위해 복수개의 분류기를 채택하여 해당 분류 결과를 선택적으로 합성하는 연구도 이미 시도되었다[2].

본 연구에서는 감정 분류의 성능을 향상시키기 위해 다음과 같은 새로운 두 가지 방법론을 제안하고 분석 평가한다.

첫째, 감정 분류 향상을 위해 부정어 처리를 고도화 한다. 이전 sentiment 분석에 관한 연구들[3, 4, 5, 6]은 주로 영화 리뷰 혹은 상품 리뷰 등의 문서 내에서 긍정/부정을 정확하게 판단하는데 중점을 두었는데, 부정어 처리에 대해서는 세밀한 연구가 진행되지 못했으며 부정어 처리가 감정 분류 성능 향상에 미치는 영향에 대해서는 정확하게 검증된 바가 없다.

둘째, 블로그 문서 내에서 두드러지게 나타나는 언어자질들을 탐지해내기에 적합한 새로운 단어 가중치 계산법을 적용한다. 기존 sentiment 분석 연구[5, 7]에서는 단어의 출현빈도를 긍정/부정의 분류를 위한 단어 가중치 계산에 이용하였지만, 4가지 정도의 감정이 혼재하는 블로그 문서에서의 감정 분류에는 좋은 성능을 나타내지 못했다.

본 연구는 감정 말뭉치(mood corpus)에서 나타나는 다양한 부정어의 패턴에 대한 조사를 기반으로 확장된 감정 단서(clue)가 포함된 정규화된 부정어 n-gram과 새로운 단어 가중치 계산법이 어떻게 감정 분류의 성능 향상에 효과적으로 적용될 수 있는지 알아본다. 특히, 서로 다른 처리 방식의 감정 분류 기법으로 제안된 강화된 감정 흐름 분석(Enhanced Mood Flow Analysis, 이하 EMFA) 방법과 지지벡터기계 기반의 감정 분류(Support Vector Machine based Mood Classification, 이하 SVMCM)에 제안한 기법들을 적용하여 그 효용성을 검증하였다.

관련 연구

몇몇의 sentiment 분석 연구[5, 6]에서는 영화 리뷰 문서들을 uni-gram과 간단한 부정어 처리를 포함한 bi-gram등의 자질을 써서 3가지 범주(긍정, 부정, 중립) 중 하나로 분류하는 실험을 하였다. 이러한 연구의 결과로 전통적인 장르(혹은 토픽) 분류에서는 효과적인 기계학습기반 분류 기법이 sentiment 분석에는 상대적으로 좋은 성능을 얻기 힘들다는 결론을 내렸다.

다른 연구[7]에서는 영화 리뷰 문서에서 부정, 강조, 감쇄 등을 고려하여 보다 세밀한 sentiment 분석을 시도하였는데, 특히 긍정/부정어들의 동의어, uni-gram 및 bi-gram에 걸친 다양한 자질들을 기반으로 수행한 실험 결과를 제공하였다. 단어빈도수만을 고려한 기법은 성능이 좋지 않았지만, 부정, 강조, 감쇄를 고려한 bi-gram을 결합한 방법은 sentiment 분석기의 성능을 향상시킬 수 있다고 기술하고 있다. 비록 uni-gram자질만 사용한 [5, 6]의 결과를 반박하며 특수한 bi-gram을 사용한 지지벡터기계기반 sentiment 분석 방법의 성능이 더 높다고 주장하지만, 그 차이는 0.01% 포인트로 미미한 수준이다.

기존연구들과 비교하여 본 연구에서는 단순히 긍정/부정을 판단하는 sentiment 분석에서 벗어나 4가지의 감정(즉, 행복, 슬픔, 분노, 두려움)의 감정 분류를 목표로 한다. 또한 감정 분류 기법의 성능을 고도화하기 위해 부정어 처리 및 새로운 단어 가중치 계산법을 채택하여 감정 분류 성능 향상을 모색한 것이 차별점이다.

감정 말뭉치 (Mood Corpus)

본 연구에서는 실제의 블로그 말뭉치를 구축하기 위해 영어권에서 매우 많이 쓰이고 있는 “LiveJournal.com”의 블로그 문서들을 수집하여 말뭉치를 구축하였다. 이 블로그 문서들은 작성자들이 각자의 블로그를 작성하기 전 132개의 감정 중에 하나를 선택하게 있기 때문에, 감정을 담고 있는 문서 수집에 있어서 매우 유용하다. 실험을 위해 4가지 감정(행복, 슬픔, 분노, 두려움)에 대응되는 유사 감정을 LiveJournal의 감정 계층구조(Hierarchy)¹⁾를 참조하여 추출하고 이에 해당하는 약

50GB에 달하는 문서를 수집하였다.

이를 위해 우선 주요 감정 자질들을 포함하고 있는 14,000여 개의 문서를 기계적으로 추출한 후, 영어에 능숙한 대학원생 2명이 공통으로 동의하는 문서들을 최종 말뭉치에 포함시켰다. 이 과정에서 문서의 길이가 매우 짧아서 기계적 판단이 애매하거나, 무의미하게 길어서 말뭉치에 포함시키기 어려운 블로그 문서들을 배제하여, 말뭉치에는 5~40문장으로 이뤄진 블로그 문서만을 포함시켰다. 표 1은 목표하는 4가지 감정에서 각각 1000건 이상의 문서를 확보하기 위해, 약 4개월 동안 20회에 걸쳐 LiveJournal.com에서 블로그 문서를 수집 및 검증하여 최종 구축한 감정 말뭉치의 누적 통계를 보여준다.

표 1. 감정 말뭉치 구축의 통계

	1st	2nd	3rd	4th	...	13th	14th	...	20th	total
Happy	877	1,111	2,017			4,005
Sad	50	93	53	128	...		145	...		1,032
Angry	22	33	27	37	...	280	85	...		1,139
Fear	8	5	2	16	...	202	67	...	33	1,032
Total	957	1,242	2,099	181	...	482	297	...	33	7,189

부정어 패턴 분석 및 정규화된 부정어 n-gram

우리가 일상적인 영어 문장에서 발견할 수 있는 대표적인 부정문에는 크게 3가지 종류(not 부정문, no 부정문, never 부정문)가 있는데, 간단한 예는 아래 (1)~(3)과 같다.

- (1) I'm not happy right now.
- (2) I have no dream and no hope.
- (3) I am never good enough.

1) <http://www.livejournal.com/moodlist.bml?moodtheme=140&mode=true>

부정어의 패턴 및 패턴 별 출현 빈도의 정도를 구체적으로 알아보기 위해 자체 구축한 감정 말뭉치의 영문 블로그 문서들에 대해 POS태깅을 수행하고 부정어 전후에 나타나는 패턴들을 유사한 것들끼리 모아서 표 2와 같은 부정문의 패턴 통계를 획득하였다. 총 7,189건의 문서(17,285개의 부정문 포함 문장)에서 3가지 부정어 유형에 관해 12가지의 대표적인 패턴이 관찰되었지만, 5개의 패턴(패턴1, 2, 3, 6,

표 2. 감정 말뭉치에서 획득한 부정문의 패턴 통계

	패턴들	빈도	점유율
not	1. AV or V + not + (ADV) + V e.g., I do not like Mare's friends.	9075	56.51%
	2. AV or V + not + (ADV) + A e.g., I'm not happy right now.	1344	8.37%
	3. AV or V + not + (DET) + (ADV) + (A) + N e.g., This is not a good thing.	1711	10.65%
	4. AV or V + not + PREP	887	5.52%
	5. AV or V + not + PRO	3	0.02%
no	6. no + (DET) + (ADV) + (A) + N e.g., I have no good grade.	1349	8.4%
	7. no + (DET) + A	167	1.04%
	8. no + COM	71	0.44%
	9. no + V-ing	55	0.34%
never	10. never + (ADV) + V e.g., I have never felt good.	1090	6.79%
	11. never + (ADV) + A	36	0.22%
	12. never + AV	29	0.18%
	기타	243	1.51%
	총계	16060	100%

AV(Auxiliary Verb), V(Verb), A(Adjective), N(Noun), ADV(ADVerb),
PREP(Preposition), PRO(Pronoun), COM(COMParative degree).

10)이 90% 이상을 차지하였다.

그리고 부정문의 단어들을 조사해본 결과 16,060 문장 중 약 17% 정도만이 감정적 단어(affective words)를 포함하고 있었는데, 각 패턴마다 부정문 처리 규정을 만드는 것은 여러 가지 예외상황들(문법에 맞지 않은 문장들이 많으며, 부사, 형용사와 같은 수식어가 한 문장에 빈번히 사용되는 등 문장 내에서 정확하게 일치하는 패턴을 찾기가 쉽지 않음)로 인해 현실적이지 못하다.

따라서 정확한 문법, 순서 등을 고려하는 것 보다, N개의 단어 중에 Affective Norms English Keywords(이하, ANEW)[8] 리스트 단어와 부정어가 포함된 여부를 고려하여 정규화된 표현을 추출하는 것이 더 효율적인 것으로 분석되었다.

다양한 부정문 패턴을 다루기 위해, 각 부정어(not, no, never)와 인접한 단어들을 결합하여 새로운 합성어를 만들고 정규화하여, 이를 "정규화된 부정어 n-gram"으로 명명하였다. 정규화된 부정어 n-gram을 구성하기 위해 불용어(stop words)를 제거하여 의미 없는 단어에 의한 방해를 방지하고, 의미 있는 감정적 단어들을 모아놓은 ANEW 리스트에 나오는 단어가 포함된 표현들을 주로 추출하였는데, 표 3은 출현 빈도 1이상의 정규화된 부정어 n-gram들의 예시이다.

표 3. 정규화된 부정어 n-gram의 예시

n-gram	Example Features
n=2	Not hurt, not pain, never happy, not sad, no money, no success, ...
n=3	Not lot fun, no fun sick, not admit sad, not optimistic happy, ...
n=4	Not found makes happy, never feel bitter lost, ...

말뭉치 기반 단어 가중치 계산법

본 연구에서는 주어진 블로그 문서들을 4가지의 감정(행복, 슬픔, 분노, 두려움)으로 효율적으로 분류하기 위한 새로운 단어 가중치 계산법을 고안하였는데, 말뭉치를 기반으로 확률계산을 한 값을 사용하므로 CSTW(Corpus-Specific Term

Weighting)라 명명하였다.

주어진 단어에 대해 단어가 각각 4종류의 감정을 얼마나 잘 나타내는지의 의미를 4가지의 가중치 값[happy, sad, angry, fear]을 고려할 수 있다. 각 감정에 대한

W={w1,w2, ..., wj}: 단어 리스트
 M={m1,m2, ..., mj}: 감정 범주 (본 논문에서 j=4)

단어(w_i)가 감정 범주(m_j)에 나타날 상대적 성향을 P(m_j|w_i)/P(m_j)와 같은 확률의 비율로 표현할 수 있다. 여기서 P(m_j|w)는 단어 w를 포함하는 문서가 감정 범주 m에서 나타날 조건부 확률을 나타내고, p(m)은 전체 말뭉치 중에서 감정 범주 m을 가지는 문서의 비율을 의미한다. 본 논문에서는 한 단어가 특정 감정 범주에 대해 월등히 우세한 성향을 보이는 경우와 그렇지 않은 경우를 구별해서 각 감정범주 별 가중치를 결정한다. 특히, 전자의 경우, 우세한 성향의 값만을 극단적으로 강조하는 정책을 채택하기로 한다.

1) 예를 들어 단어 w가 감정 범주 m₃와 높은 상관관계를 가져서 임계치(실험적 경험치)를 초과한다면, 즉 다음이 성립하면, $\frac{P(m_3|w)}{P(m_3)} > threshold$

계산된 가중치는 감정 범주 m₃에 대해서만 할당하고, 나머지 감정 범주에는 0의 값을 설정한다 즉, 말뭉치기반 단어 가중치(Corpus-specific Term Weighting, CSTW)는 다음과 같이 계산한다. SVMMC는 최대 가중치만 적용 한다

a) EMFA 경우: $\overline{CSTW}(w) = \frac{1}{N} \times \left[0.0, \frac{P(m_3|w)}{P(m_3)}, 0 \right]$

b) SVMMC 경우: $\overline{CSTW}(w) = TF_w \times \frac{1}{N} \times \max(0.0, \frac{P(m_3|w)}{P(m_3)}, 0)$

2) 특정 감정 범주를 향한 강한 성향을 나타나지 않은 경우에는 모든 감정 범주에 대하여 다음과 같이 가중치를 할당한다.

a) EMFA 경우: $\overline{CSTW}(w) = \frac{1}{N} \times \left[\frac{P(m_1|w)}{P(m_1)}, \frac{P(m_2|w)}{P(m_2)}, \frac{P(m_3|w)}{P(m_3)}, \frac{P(m_4|w)}{P(m_4)} \right]$

b) SVMMC 경우: $\overline{CSTW}(w) = TF_w \times \frac{1}{N} \times \max\left(\frac{P(m_1|w)}{P(m_1)}, \frac{P(m_2|w)}{P(m_2)}, \frac{P(m_3|w)}{P(m_3)}, \frac{P(m_4|w)}{P(m_4)}\right)$

이때, N은 $\sum_{j=1}^4 \frac{P(m_j|w)}{P(m_j)}$ 이고, 두개 이상의 범주에서 임계치를 초과하는 경우, 최댓값을 취한다.

그림 1. 말뭉치 기반 단어 가중치(CSTW) 알고리즘

상대적 가중치는 주어진 단어를 포함하는 전체 블로그 문서 수에 대비 각 감정을 나타내는 블로그 문서의 수의 비율로 계산할 수 있다. ($N(k)$ 와 $N_m(k)$ 는 각각 단어 k 를 포함하는 전체 블로그 문서의 수와 단어 k 를 포함하고 감정 m 을 나타내는 블로그 문서의 수를 의미한다.)

$$TermWeight_k = \left[\frac{N_{happy}(k)}{N(k)}, \frac{N_{sad}(k)}{N(k)}, \frac{N_{angry}(k)}{N(k)}, \frac{N_{fear}(k)}{N(k)} \right]$$

하지만, 이 방법만으로는 4가지의 감정 범주 중 가장 두드러지는 한 개의 감정을 추출하기가 어렵다. 감정이 두드러지는 단어를 찾고 단어의 정확한 가중치를 결정하기 위하여, 본 논문에서는 2가지 감정 범주(negative와 non-negative)를 위하여 개발된 emotional salience[10]의 개념을 4가지 감정 범주에 확대·수정 적용한 CSTW를 제안 한다(그림 1). 이는 특정 감정 범주에서 월등히 우세한 경향을 보이는 경우와 그렇지 않은 경우로 나누어 EMFA와 SVMMC에 차별적으로 적용한다.

시스템 구성

그림 2에서 볼 수 있듯이, 전체 과정은 크게 3가지 단계로 나눌 수 있다. 블로그 문서가 들어오면, 시스템은 전처리 과정을 통하여 두 종류의 분류(classification) 방법에 쓰일 자질들(features)을 생성한다. 전처리 과정에는 오타자 처리, 불용어 제거, 의성어 처리뿐만 아니라 이모티콘(emoticon), 약어 등과 같이 블로그에서 자주 쓰이는 특정 표현 처리 등이 포함된다. 부정문 처리 단계에서는 아래에 상세히 기술되는 2가지의 부정문 처리 기술(PAD에 의한 부정문 처리와 CSTW 방법에 적용하는 정규화된 부정어 n-gram 기반 처리)이 쓰인다. 마지막 단계에서는 2개의 다른 분류 모듈이 각각 주어진 블로그 문서의 감정을 분류하기 위하여 사용된다. 아래에서는 사용된 부정문 처리 기술 및 EMFA와 SVMMC에 대해 자세히 설명한다.

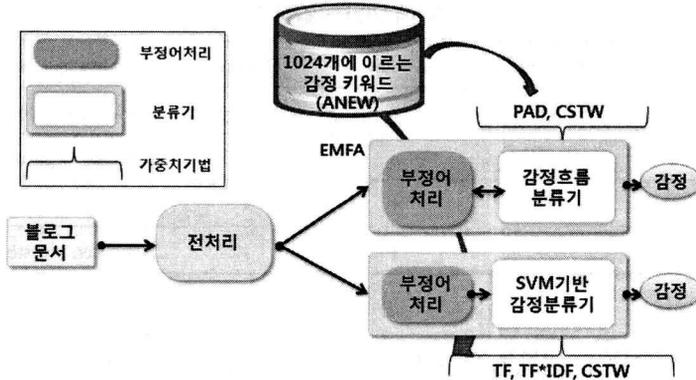


그림 2. 시스템 구조

부정문 처리

본 시스템의 부정문 처리는 PAD에 의한 부정문 처리와 정규화된 부정어 n-gram 방법 2가지를 적용하며 각각에 대한 상세한 설명은 다음과 같다.

PAD에 의한 부정문 처리

ANEW 리스트에는 1000여 개의 주요 감정단어가 포함되어 있는데 각각에 대해 P(pleasure-displeasure), A(arousal-nonarousal), D(dominance-submissiveness)의 세 가지 척도에 대한 값이 달려 있다. 이 값들은 1~10 사이에 있는데, PAD기반의 감정 분류를 위해 -1~1 스케일로 변환시킬 수 있다. 따라서 문서에 부정어와 함께 출현하는 단어가 ANEW 리스트에 존재하는 경우, 이 PAD 값의 일부 극성(polarity)을 변화시키면 부정어 처리가 가능하게 된다.

PAD값에 기반한 감정 분류를 위해 [14]를 참조하여 표 4를 구성할 수 있는데, 이는 4가지 감정범주 각각에 대해 PAD값을 할당한 결과이다. 이를 기반으로 특정 감정 범주에 속하는 단어 PAD값들의 극성 반전을 각 감정 범주 별로 실험해 본 결과, 행복(happy)과 슬픔(sad)에 해당하는 단어들의 극성 반전만이 유효했으며, 다른 감정 분류에서는 많은 오류를 발생시켰다. 분노(angry)와 두려움(fear)에 해당하는 단어들의 경우는 극성 반전 후 PAD값을 합산시켰을 때 다른 감정 단어들의 지지도를 떨어뜨

표 4. PAD값에 기반한 감정 분류 기준

Mood	P value	A value	D value	Example words
Happy	+	+ or -	- or +	Hope, joy, love, etc.
Sad	-	-	-	Alone, depressed, etc.
Angry	-	+	+	Anger, dirty, etc.
Fear	-	+	-	Cancer, disaster, etc.

러 오류를 확산시키는 결과를 낳아 부정어 처리를 위한 극성 반전에서 배제한다.

행복-슬픔 간의 감정 변화에서 여러 조합의 PAD극성 반전 실험을 볼 때, P와 A의 극성 반전이 유효한 결과를 가져왔으며, D의 극성을 변경하는 것은 기존의 감정 분류 성능을 전반적으로 떨어뜨리는 결과를 초래하였다. 따라서 PAD기반 부정어 처리에서는 ANEW리스트에서 행복과 슬픔에 해당하는 단어들 앞에 no/never/not 등의 부정어 나타날 경우 그들의 P와 A의 값만 일괄적으로 반전시켰다(그림 4). 이렇게 필요한 반전을 시킨 후 주어진 단락에서 가장 높은 가중치를 갖는 감정을 최종 값으로 결정하였다.

정규화된 부정어 n-gram을 통한 부정문 처리

감정 단어를 포함하는 모든 부정문 패턴은 복잡한 문법처리 없이 정규화된 부정어 n-gram으로 나타낼 수 있다. 부정어 not을 포함한 bi-gram인 “not good”과 같은 경우 “not_good”으로 정규화 하여 하나의 독립된 자질로 취급할 수 있다. 따라서 정규화된 부정어 n-gram은 TF, TF*IDF, 그리고 앞에서 설명한 CSTW와 같은 단어 가중치 계산법에서 하나의 자질로서의 역할을 한다. PAD에 의한 부정문처리 방법이 EMFA에만 사용되는데 반해, 정규화된 n-gram 기반 부정문 처리 방법은 CSTW에 적용되어 EMFA와 SVMCMC에 같이 사용된다.

강화된 감정 흐름 분석(Enhanced Mood Flow Analysis - EMFA) 기법

기존 연구에서는 블로그 문서에 존재하는 여러 단락 사이의 감정 변화를 관찰

하기 위하여, ConceptNet[11]에 있는 GuessMood의 수정 버전과 ANEW 리스트를 이용한 감정 흐름 분석기(Mood Flow Analyzer-MFA)[2]가 개발되었다. 강화된 감정 흐름 분석기(EMFA)에서는 부정문을 다루기 위하여 PAD에 의한 부정어 처리와 정규화된 부정어 n-gram을 통한 부정어 처리 방법을 CSTW를 통해 추가하였다. 즉, EMFA를 기반으로 2개의 서로 다른 부정어 처리 방법을 테스트 하였고, 그 결과는 7절에서 상세히 설명한다.

감정 흐름 분석기(MFA)

MFA[2]는 블로그 문서를 여러 개의 단락으로 나눈 후 단락 내 존재하는 단어들의 특성에 따라 세부적인 절차를 결정한다. 각 문단 분석 단계에서는, PAD 모듈과 수정된 GuessMood 중 한 개의 방법을 선택하기 위하여 감정적 단어의 개수를 센다. 만약 감정적 단어의 개수가 임계값(즉, 3)을 넘으면 PAD 모듈을 선택하고, 넘지 않으면 GuessMood 방법을 선택한다. 각 단락의 감정분석결과는 감정의 변이를 고려하여 해당 블로그 문서의 최종 감정을 결정하는데 사용된다.

그림 3은 블로그의 단락을 입력으로 하는 PAD기반의 감정 판단 모듈로써 행복 및 슬픔에 해당하는 ANEW리스트의 단어에 대해서만 P와 A의 극성을 반전시켜 부정문을 처리한다. 그림 4는 CSTW 기반 단어 가중치를 계산하는데 있어 정규화된 부정어 n-gram기법과 PAD기반의 감정 판단 모듈의 가중치 계산 부분이 어떻게 포함되는지 보여준다. 이것들의 실험결과는 표 5에서 살펴볼 수 있다.

수정된 GuessMood

기존의 통계기반 자연어처리 기법을 사용하는 경우 “*We are stuck in the dormitory because of big snow*”와 같이 감정 단어가 없는 문장에 대해서는 감정을 판단하기 어렵다. 이러한 문제를 다루기 위해, ConceptNet의 상식 네트워크를 이용하는 GuessMood 함수를 사용할 수 있다. 본 논문에서는 기 구축된 GuessMood 함수의 계산 효율과 성능의 향상을 위해, Open Mind Common Sense(OMCS)[12] 말뭉치로부터 감정을 포함하는 문장을 추출하여, 이를 기반으로 ConceptNet 데이터를 재구성하였다. 이를 위해, 먼저 Ekman의 감정 단어 리스트[13]나 정제된 ANEW 리스트에 있는 감정 단어를 포함하는 문장을 추출하였고, 감정적 단어를 직접 포함하지는 않

입력: 블로그 문서의 단락
출력: happy, sad, angry, fear 중 하나의 감정
절차:

각 감정 가중치 변수 W_H, W_S, W_A, W_F 을 선언
PAD 가중치 변수 SUM_P, SUM_A, SUM_D 을 선언
단락상에 나타난 ANEW리스트의 개수: C_{ANEW} 선언

- $SUM_P, SUM_A, SUM_D, C_{ANEW}, W_H, W_S, W_A, W_F$ 초기화
- 단락내 각 문장에 대해
 - 각 문장 내 단어에 대해
 - If (현재 단어가 ANEW리스트에 존재)
 - $SUM_P +=$ 현 단어의 Pleasure값
 - $SUM_A +=$ 현 단어의 Arousal 값
 - $SUM_D +=$ 현 단어의 Dominance 값
 - Else If (현재 단어 앞에 부정어) and
(현재단어가 ANEW리스트에 존재) and
(단어가 행복 or 슬픈인 경우)
 - $SUM_P +=$ 현 단어의 Pleasure값 * (-1.0)
 - $SUM_A +=$ 현 단어의 Arousal 값 * (-1.0)
 - $SUM_D +=$ 현 단어의 Dominance 값

$$SUM_P \leftarrow \frac{SUM_P}{C_{ANEW}}, SUM_A \leftarrow \frac{SUM_A}{C_{ANEW}}, SUM_D \leftarrow \frac{SUM_D}{C_{ANEW}}$$

If ($SUM_P \geq 0$ and $SUM_A \geq 0$)
 $W_H += 1$

Else If ($SUM_P < 0$ and $SUM_A < 0$ and $SUM_D < 0$)
 $W_S += 1$

Else If ($SUM_P < 0$ and $SUM_A \geq 0$ and $SUM_D \geq 0$)
 $W_A += 1$

Else If ($SUM_P < 0$ and $SUM_A \geq 0$ and $SUM_D < 0$)
 $W_F += 1$

- W_H, W_S, W_A, W_F 중 최대값을 지닌 감정 반환

그림 3. PAD기반 감정 판단 및 부정문 처리

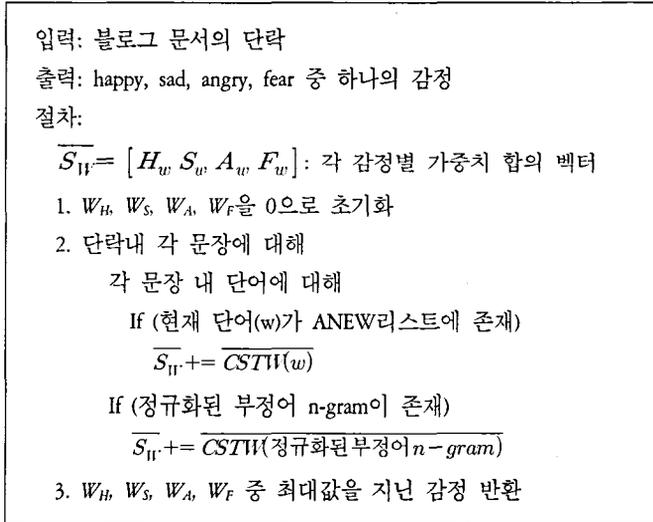


그림 4. 부정어 처리를 포함한 CSTW 계산

지만 잠재적으로 감정을 나타내고 있는 문장도 포함하기 위하여, ConceptNet의 상식 네트워크를 통해 감정 단어와 간접적으로 연결된 문장을 추출하였다. 이렇게 재구성된 ConceptNet 데이터를 이용하여 GuessMood의 성능을 극대화하기 위해 상식 네트워크의 내부 연결 가중치(linkage weight)를 휴리스틱하게 재설정하였다. 단, 여기서는 부정문 처리는 고려하지 않는다.

감정 결정기(Mood Resolver)

만약 블로그 문서 내에서 감정이 변화 없이 유지된다면, 감정 결정기 모듈은 일관성을 체크하고 최종 결과로 유지된 감정을 선정한다. 하지만 단락내에서 감정의 변이가 생기는 경우, 단락의 위치를 고려하여 휴리스틱한 가중치를 부여하게 된다. 블로그 말뭉치에 기반한 수차례의 실험을 통하여, 문단을 강조해야 할 때(예. 첫 문단 혹은 마지막 문단)의 휴리스틱 가중치($\alpha=2, \beta=7, \gamma=8, \delta=5$)를 얻었다(그림 5).

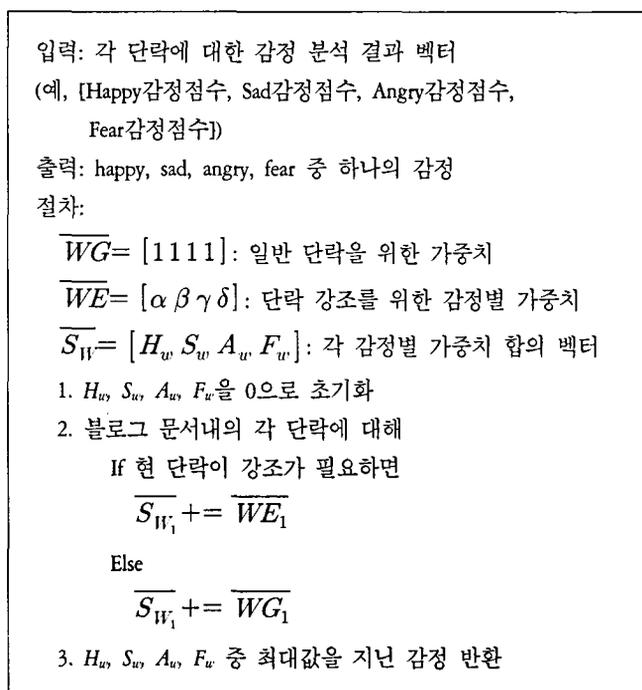


그림 5. Mood Resolver 함수

SVM 기반 감정 분류(SVM based Mood Classification-SVMCMC)

SVM 기반 감정 분류 모듈의 구현을 위해 본 실험에서는 SVM 툴킷의 하나인 SVM light²⁾를 사용하며, 블로그의 특성을 보다 면밀히 반영하기 위해 크게 5가지 종류의 자질들(감정이 분명히 나타나는 감정 단어, 강조된 단어, 이모티콘, 특수 기호, 그리고 정규화된 부정어 n-gram)을 사용하였다. 또한, 가중치 계산법에 따른 유용성을 알아보기 위해, TF, TF*IDF, 그리고 본 논문에서 제안한 CSTW 방법을 각각 적용하여 SVMCMC를 테스트하였다. 결과는 SVMCMC의 실험에서 설명한다.

2) <http://svmlight.joachims.org/>

실험 및 토론

본 장에서는 앞서 제안한 부정어 처리기법을 TF*IDF, TF 그리고 CSTW 가중치 계산법 등에 선택적으로 적용한 실험의 결과를 설명한다. 실험을 위해서 사용한 문서는 3장에서 기술한 감정 말뭉치 7000여 개의 문서 중 선택한 총 4,303개이다 (happy: 1,125, sad: 1,031, angry: 1,036, 그리고 fear: 1,011). 각 감정 범주 별로 1000여 개 이상의 문서를 갖추기 위해, 4000여 개가 넘는 행복 범주에 대해서만 1000여 개를 임의로 선택하고, 나머지 감정 범주들에 대해서는 감정 말뭉치의 것을 그대로 사용기로 하였다. 블로그 문서의 감정 분류는 블로그 기반으로 적절한 음악을 추천하거나, 감정에 맞는 이미지 혹은 문맥광고 등의 서비스에 응용가능하다.

EMFA의 실험

CSTW를 적용한 EMFA의 경우에는, 2-gram에서 4-gram으로 점차 영역을 확대 적용하여 실험을 하였을 경우 점진적인 감정 분류 성능의 향상을 확인할 수 있었다 (표 5). 베이스라인인 MFA(부정어 처리 모듈 없음)에 비해서 CSTW를 적용한 EMFA의 기본 설정은 약 10% 높은 성능을 보였다. 이 결과는 부정어 처리는 단순히 바로 다음에 오는 단어의 감정 범주를 고려하여 극성을 단순 반전 시키기 보다는, 문장 내 존재하는 부정어와 인접한 감정 단어 그리고 감정 단어가 영향을 줄 것으로 예상 되는 인접 단어들까지 확장 고려하는 것이 감정 분류를 위한 하나의 자질로 고려하는 것이 유용함을 보여준다. 즉, 정규화된 부정어 n-gram은 ANEW리스트의 단어들이 부정어와 결합하여 해당 문서의 감정 범주를 나타내는 주요 자질로서 감정 분류의 성능에 영향을 미친 것이다.

하지만, 부정어 처리가 항상 도움이 되었던 것은 아니다. PAD기반의 감정 분류 기법에서 단순히 극성을 변화시킨 경우는 전반적으로 약 2.81% 포인트의 성능 향상을 보이긴 했지만, 이는 4가지 감정들 모두에 있어서 향상을 보이지는 못했고 단지 ‘슬픔’에 대해서만 향상을 보였다. 이는 PAD기반의 감정 분류 기법에서 적용한 단순 극성 반전의 한계로 다른 감정들의 경우에 있어서는 오히려 오류를 유

표 5. EMFA 실험 통계

Mood	MFA (baseline)	EMFA (PAD기반 부정어처리)	EMFA (CSTW) 'default'	EMFA (CSTW -2gram)	EMFA (CSTW -3gram)	EMFA (CSTW -4gram)
Happy (1125)	974 (86.56%)	951 (84.51%)	1117 (99.26%)	1116 (99.17%)	1119 (99.44%)	1119 (99.44%)
Sad (1031)	448 (43.45%)	605 (58.68%)	517 (50.15%)	552 (53.54%)	658 (63.82%)	738 (71.58%)
Angry (1136)	718 (63.20%)	718 (63.20%)	954 (83.98%)	968 (85.21%)	1020 (89.79%)	1046 (92.08%)
Fear (1011)	659 (65.18%)	646 (63.90%)	636 (62.91%)	635 (64.59%)	709 (70.13%)	762 (75.37%)
Total (4303)	2788 (65.04%)	2920 (67.95%)	3224 (74.92%)	3289 (76.43%)	3506 (81.47%)	3665 (85.17%)

발하였다.

SVMCMC의 실험

정규화된 부정어 n-gram을 적용한 SVMCMC의 경우, 실험에서 확인한 정확도의 향상은 0.08~0.53%정도로, n-gram의 coverage를 증가시켰을 때 EMFA의 경우처럼 점진적인 성능 향상을 가져오지는 못 했다. 이 미미한 성능 향상은 [7]의 성능 향상(0.01% 포인트)과 매우 유사하다. 비록 1%의 향상에도 미치지 못한 저조한 성능 향상의 이유를 파악하기 위해 면밀한 추가적 실험이 필요하지만, 현재의 분석으로는 SVMCMC에서 사용된(6.3에서 언급) 다양한 자질들로 인해 정규화된 부정어 n-gram이 감정 분류에 영향을 미치는 추가적인 자질로서의 효과가 미미한 것으로 추정된다.

다양한 자질을 사용하는 SVMCMC에서 이미 어느 정도 안정적인 성능이 4가지의 감정 범주에서 나타나고 있기 때문에, 추가적 분류 성능 향상을 꾀할 수 있는 방

표 6. 다른 단어 가중치 계산법 적용시의 효과

Mood	SVMMC (TF*IDF)	SVMMC (TF)	SVMMC (CSTW)
Happy (1125)	1075 (96.56%)	1086 (96.88%)	1091 (97.12%)
Sad (1031)	810 (78.56%)	849 (82.35%)	854 (82.83%)
Angry (1136)	1036 (91.20%)	1067 (93.93%)	1092 (96.13%)
Fear (1011)	839 (82.99%)	853 (84.37%)	842 (83.28%)
Total (4303)	3760 (87.40%)	3855 (89.62%)	3879 (90.18%)

법은 다른 단어 가중치 계산법을 적용하는 것이다.

표 6(5-fold cross evaluation의 결과)에서 보듯이 부정어처리를 하지 않고, 서로 다른 단어 가중치 계산법을 적용한 경우, 분류성능에 평균 1~3%정도로 영향을 미치고 있는데 그만큼 적절한 단어 가중치의 선택이 중요함을 알 수 있다. 현재 실험으로는 CSTW 단어 가중치 계산법이 4 가지 감정 범주들에 대해 평균 90.17%의 정확도로 TF*IDF나 TF보다 우수한 성능을 보이고 있다.

좀 더 자세히 보면 TF 단어 가중치 계산법은 TF*IDF 단어 가중치 계산법에 비해 평균 2.2% 포인트의 성능 향상을 보였는데, 4가지 감정 범주 모두에 대해 전반적인 성능 향상을 가져왔다. 이에 반해 CSTW 단어 가중치 계산법의 성능을 TF 단어 가중치 계산법의 성능과 비교하면 평균 약 0.46% 포인트의 정확도 향상을 보였는데, 두려움을 제외한 나머지 3가지의 감정 범주들에 대해서는 TF 단어 가중치 계산법보다 높은 성능을 보였다. CSTW 단어 가중치 계산법에 의한 성능 향상이 다소 미미해 보이는 것 같지만 [7]의 연구에서 특수한 2-gram을 도입하여 기존의 1-gram 접근법보다 약 0.008% 포인트의 성능 향상을 보인 것과 비교하면 상당히 높은 수치이다.

토론

EMFA와 SVMMC의 실험에서 볼 때, 감정 말뭉치에서 다양한 자질을 모두 사용하기보다는 특정 감정에서 자주 나타나는 단어들만을 사용해서 감정을 분류하는

는 EMFA의 경우 정규화된 부정어 n-gram을 자질로 추가하는 것이 감정 분류에 있어서 긍정적으로 작용하여 성능 향상에 도움이 되었지만, 이미 다양한 자질들을 지지벡터기반의 학습데이터로 사용한 SVMMC의 경우 그 성능 향상이 미미하였다.

하지만, CSTW는 단어가중치 계산법을 통해 감정을 담고 있는 단어들과 이런 단어들이 포함된 감정 단서(mood clue)들이 보다 높은 가중치를 부여 받아서 EMFA 및 SVMMC 모두에서 감정 분류에 도움을 주었다.

추가적으로, 위 EMFA와 SVMMC의 결과물을 [2]처럼 조합하여 보다 높은 성능을 기대할 수 있는데, 보다 콘텐츠의 내용을 보다 면밀히 파악하여 적절한 분류결과를 선택하는 것이 필요하다.

결론

본 논문에서는 감정 분류의 성능 향상을 위해 정규화된 부정어 n-gram 및 말뭉치 의존 단어 가중치 계산법을 제안하고 4가지의 감정 범주에 대한 분류 성능 향상을 자체 구축한 약 4000여 개의 블로그 문서를 통해 검증하였다. 특히, 감정 분류에 있어 부정어 처리가 어떤 영향을 미칠 수 있는지에 대한 실험을 통해 고찰하였으며, 3가지의 서로 다른 단어 가중치 계산법을 적용 및 실험하여 감정 분류 시에 단어 가중치 계산법이 성능에 미치는 영향을 살펴보았다.

본 연구의 가장 큰 시사점은 부정어 처리와 단어 가중치의 적절한 선택이 감정 분류 성능 향상에 직접적으로 상당한 영향을 미치는 것을 입증한 것이다. 정규화된 부정어 n-gram에서는 2-gram에서 4-gram까지 그 영역을 확대하면서 CSTW 단어 가중치 계산법을 함께 적용하였을 경우, 부정어 처리를 하지 않았을 경우와 대비하여 약 10%정도 성능이 향상되었다. SVMMC의 경우도 TF에 비해 CSTW 적용하였을 때 0.46% 높은 성능 향상을 보였는데, SVMMC에서는 자질의 추가적 선택보다는 적절한 단어 가중치 기법의 선택이 필요함을 알 수 있었다.

학문적인 측면에서 블로그의 감정분류 분야에서 최적의 단어가중치 모델이 나온다면 CSTW를 뛰어 넘어 보다 높은 성능향상을 실현할 수 있겠지만, 본 논문에서는 TF, TF*IDF와는 진화된 모델을 기용하여 그 가능성을 제시했다고 할 수 있

다. 또한, 실용적인 측면에서, light-weight한 감정분류기를 구현코자할 때는 EMFAR을, 고성능의 감정분류기를 구현코자하고 적절한 학습데이터가 가용할 때에는 SVM을 쓰는 것이 유용하다.

향후 연구방향으로 인간 생활 속의 행위, 사물 등에서 감정과 관련된 추가적인 단서(clue)를 지도 및 비지도 학습기반 bootstrapping기법을 통해 웹 말뭉치로부터 확장하여 블로그 뿐만 아니라 다른 문서에서의 감정 분류에도 활용될 수 있는 방법론을 개발하고자 한다.

참고문헌

- [1] Gilad Mishne, "Experiments with Mood Classification in Blog Posts", Style2005 at SIGIR 2005.
- [2] Yuchul Jung, Yoonjung Choi and Sung-Hyon Myaeng, "Determining Mood for a Web Blog by Combining Multiple Sources of Evidence", Proc. of IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI-07), pp. 271-274, 2007.
- [3] Kushal Dave, Steve Lawrence and David M. Pennock, "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews", Proc. of the 20th Int. World Wide Web Conf. (WWW-03), pp. 519-528, 2003.
- [4] Jin-Cheon Na, Christopher Khoo and Paul Horng Jyh We, "Use of negation phrases in automatic sentiment classification of product reviews", Library collections, Acquisitions, & Technical Services 29: 180-191, 2005.
- [5] Bo Pang, Lillian Lee and Sivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques", Proc. of the ACL-02 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 79-86, 2002.
- [6] Bo Pang and Lillian Lee, "A Sentiment Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts", Proc. of the 42nd Annual Meeting on Association for Computational Linguistics (ACL), no (271), 2004.
- [7] Alistair Kennedy and Diana Inkpen, "Sentiment Classification of Movie Reviews Using

- Contextual Valence Shifters”, Computational Intelligence, vol(22), no(2), pp. 110-125, 2006.
- [8] Bradley, M. M. and Lang, P. J. Affective norms for English words (ANEW). Gainesville, FL. The NIMH Center for the Study of Emotion and Attention, University of Florida, 1999.
- [9] Rada Mihalcea and Hugo Liu, “A Corpus-based approach to finding happiness”, Proc. of AAAI Spring Symposium on Computational Approaches to Weblogs, 2006.
- [10] Chul Min Lee and Shrikanth S. Narayanan, “Toward Detecting Emotions in Spoken Dialogs”, IEEE Transaction on Speech and Audio Processing, 13(2): 293-303, 2005.
- [11] Hugo Liu and Push Singh, “ConceptNet - a practical commonsense reasoning tool-kit”, BT Technology Journal: 211-226, 2004.
- [12] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins and Wan Li Zhu, “Open Mind Common Sense: Knowledge acquisition from the general public”, Proc. of the 1st Int. Conf. on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, 2002.
- [13] Ekman, P. Facial expression of emotion. American Psychologies 48:384-392, 1993.
- [14] Mehrabian, A. Framework for a comprehensive description and measurement of emotional states. Genetic, Social, and General Psychology Monographs. 121(3): 339-361, 1995

1 차원고접수 : 2008. 10. 27

2 차원고접수 : 2008. 12. 3

최종게재승인 : 2008. 12. 11

(Abstract)

A Study on Negation Handling and Term Weighting Schemes and Their Effects on Mood-based Text Classification

Yuchul Jung

Yoonjung Choi

Sung-Hyon Myaeng

Information and Communications University

Mood classification of blog text is an interesting problem, with a potential for a variety of services involving the Web. This paper introduces an approach to mood classification enhancements through the normalized negation n-grams which contain mood clues and corpus-specific term weighting (CSTW). We've done experiments on blog texts with two different classification methods: Enhanced Mood Flow Analysis (EMFA) and Support Vector Machine based Mood Classification (SVMMC). It proves that the normalized negation n-gram method is quite effective in dealing with negations and gave gradual improvements in mood classification with EMFA. From the selection of CSTW, we noticed that the appropriate weighting scheme is important for supporting adequate levels of mood classification performance because it outperforms the result of TF*IDF and TF.

Keywords : Mood Classification, Normalized Negation N-gram, Weighting Scheme, EMFA, SVMMC