문서 분류 알고리즘을 이용한 한국어 스팸 문서 분류 성능 비교

송철환[O] 유성준
세종대학교 컴퓨터 소프트웨어 공학과
peternara[O]@naver.com, sjyoo@sejong.ac.kr

Comparing Korean Spam Document Classification Using Document Classification Algorithms

Chull Hwan Song[O], Seong Joon Yoo
School of Computer Engineering, Sejong University, 98 Gunja, Gwangjin
Seoul, Korea 143-747

요 약

한국은 다른 나라에 비해 많은 인터넷 사용자를 가지고 있다. 이에 비례해서 한국의 인터넷 유저들은 Spam Mail에 대해 많은 불편함을 호소하고 있다. 이러한 문제를 해결하기 위해 본 논문은 다양한 Feature Weighting, Feature Selection 그리고 문서 분류 알고리즘들을 이용한 한국어 스팸 문서 Filtering연구에 대해 기술한다. 그리고 한국어 문서(Spam/Non-Spam 문서)로부터 명사를 추출하고 이를 각 분류 알고리즘의 Input Feature로써 이용한다. 그리고 우리는 Feature weighting 에 대해 기존의 전통적인 방법이 아니라 각 Feature에 대해 Variance 값을 구하고 Global Feature를 선택하기 위해 Max Value Selection 방법에 적용 후에 전통적인 Feature Selection 방법인 MI, IG, CHI 들을 적용하여 Feature들을 추출한다. 이렇게 추출된 Feature들을 Naive Bayes, Support Vector Machine과 같은 분류 알고리즘에 적용한다. Vector Space Model의 경우에는 전통적인 방법 그대로 사용한다. 그 결과 우리는 Support Vector Machine Classifier, TF-IDF Variance Weighting(Combined Max Value Selection), CHI Feature Selection 방법을 사용할 경우 Recall(99.4%), Precision(97.4%), F-Measure(98.39%)의 성능을 보였다.

1. 서 론

현재 한국은 고속 인터넷 보급으로 인해 많은 인터넷 사용자를 보유하고 있다. 동시에, Spam mail, Adult multimedia data, Internet 범죄와 같은 많은 문제점들이 계속 증가하고 있다. 그 중 가장 심각한 문제가 바로 Spam mail이다. 더구나 한국은 다른 영어권 국가와 달리 고유의 언어를 사용하고 있다. 특히, 많은 인터넷 보급에도 불구하고 문서 분류에 대한 선진 기술을 가지고 있는 미국과는 달리 이에 대한 연구가 미비하다. 따라서 본 논문은 한국어 스팸 문서 Filtering연구에 대해 기술한다. 한국어 스팸 문서를 분류하기 위해서 먼저 한국어에 맞는 Document Preprocessing을 거쳐야 한다. 이 과정을 통해 한국어 명사를 추출한 후, 다양한 Feature Weighting, Feature Selection 그리고 Classification Algorithm들을 이용하여 적용한다.

2. 이전 연구

Spam 분류 연구는 다양한 형태의 Feature구성과 이를 다양한 문서 분류 알고리즘에 적용하는 형태로 이루어져 왔다. 먼저 Fu et al.[1]는 Multi-grams과 Bayesian Learning 을 이용하여 분류하였다. 이때의 연구는 97%의 정확성과 2% False Positive를 보여주었다. 또한 Shrestha et al.[2]의 연구는 [1]의 연구와 마찬가지로 Bayesian Learning을 이용하고 Co-Weighting을 적용하여 Spam 문서를 분류하였다. Iwanaga et al.[3]의 연구 또한 [1]과 [2]의 연구와 마찬가지로 Bayesian Learning을 이용하고 bi-gram을 적용하여 일본어 Spam 문서를 분류하였다. Stuart et al.[4]의 연구는 Neural Network 이용하여 Junk Mail을 분류하였다. 특히 Hidden Node를 12개 사용했을 때 Spam Precision(약 92%), Spam Recall(약 92%), Legitimate Precision(약 91%) 그리고 Legitimate Recall(92%)의 성능을 보였다. 아주 Simple한 Binary Feature, TF를 적용함에도 불구하고 높은 성능을 보인다. 우리도 이러한 Feature를 포함하여 한국어 문서에 적용하고 이를 다양한 분류 알고리즘에 적용하여 그 성능 비교를 한다. 우리의 연구는 다양한 Feature Weighting, Feature Selection 그리고 Classification Algorithm을 적용하여 한국어 Spam 문서를 분류한다.

3. Feature Extraction 과 Feature Selection

우리는 한국어 문서에서 전처리 과정에서 기본적인 Feature로써 명사를 추출한다. 명사는 그 문장에 대해 중요한 의미를 함축하고 있다. 그러나 선택된 모든 명사는 문서 분류에 있어서 모두 필요로 하지 않는다. 즉, 보다 좋은 Feature의 선택은 그 분류 성능을 향상시키는 중요한 요인이다. 이를 위해 우리는 다양한 Feature Weighting 과 Feature Selection 알고리즘들을 이용한다. 특히 우리는 Variance Feature Weighting과 전통적인 Feature Selection방법[5]인 Mutual Information (MI), Information Gain(IG) 그리고 Chi Square Statistic(CHI)과 결합하여 보다 좋은 Feature들을 선택하고 이를 다양한 분류 알고리즘에 적용한다.

3.1 Feature Selection

본 논문은 MI, CHI 그리고 IG와 같은 Feature Selection 알고리즘을 이용한다. 또한 우리는 MI, CHI 알고리즘과 함께 비슷하게 이용하고 있는Max Value Selection(MVS)를 Table 1의 식(1)~(4)에 적용하였다. 다음은 이러한 식에 대한 설명이다.

$$MSV(T_j) = Max(TermWeighting(Spam, j), TermWeighting(NonSpam, j)) \quad \cdots\cdots(1)$$

식(1)에서 *TermWeighting*은 Table 1의 식(1)~(5), (7)의 Term Weighting 식들이 올 수 있다. 즉, 어떠한 Category에 상관없이 적용할 수 있는Global Feature들을 선택하게 한다. 이러한 Feature Selection에 있어서 장점은 좋은 Feature를 찾기 위한 과정일 뿐만 아니라 각 분류 알고리즘의 Input Feature 차원을 줄이는 장점도 가진다.

4. Term Weighting based on Variance

기본적으로 Term Weighting은 Classification 알고리즘의 Input Value이다. 즉, Term Weighting은 각 Document의 Term(Korean Noun)과 Weighting Value로 대응된다. Term Weighting의 대표적인 예는 Vector Space Model(VSM)의 TF*IDF이다. 그러나 이들은 기본적으로 각 카테고리에서만 국한된 Local Term Weighting이다. 이를 해결하기 위해 식(1)과 같이 우리는MI와 CHI의 Feature Selection 방법에서 적용한 것을 응용하여 대표적인 Term Weighting인 TF와 TF*IDF에 적용한다. 또한 우리는 TF 와 TF*IDF를 이용한 새로운 Feature Weighting을 구성한다. 다음은 이에 관한 Variance Weighting에 대한 것이다.

$$TermWeightingVariance(j) = TermWeighting_{newQuery}(j) - GlobalMeanTermWeighting(j) \quad \cdots\cdots(2)$$

새로운 쿼리가 들어왔을 경우에 Global Mean Term Weighting을 이용하여 식(2)와 같이 Term Weighting Variance 를 구한다. 우리는 이러한 Term Weighting Variance와 Feature Selection 방법인 MI, IG 그리고 CHI와 결합하여 분류 알고리즘에 적용한다. 그리고 우리는 이러한 Term Weighting Variance를 적용한 값이 기존 Term Weighting보다 좋은 성능 결과를 보인다.

본 논문은 다양한 Feature를 각 분류 알고리즘에 적용한다. Table 1은 우리가 적용한 8개의 Feature Weighting에 대해 기술하고 있다.

Table 1의 식(1)과 식(8)은 Drucker et al.[6]에 적용되었던 Feature이다. 우리는 이를 우리의 Data Set과 다양한 알고리즘을 적용하여 그 성능 비교를 보여준다.

표 1 Term Weighting

no	Feature	Weighting Formula
(1)	tf_{ij}	$\frac{Tij}{totalOfTermNumber(i)}$
(2)	$tf-Variance$	$tf_j - Max(Mean(tf_{spamj}, tf_{nonSpamj}))$
(3)	$tf*idf$	$\frac{tf_{ij}\log(N/n_j)}{\sqrt{\sum'_{j=1}(tf_{ij})^2[\log(N/n_j)]^2}}$
(4)	$tf*idf-Variance$	$tf*idf-Max(Mean(tf*idf_{spamj}, tf*idf_{nonSpamj}))$
(5)	$I(j,i)$	$\log\frac{A+N}{(A+C)*(A+B)}$
(6)	$G(j)$	$-\sum_{i=1}^{m}\Pr(C_i)\log\Pr(C_i)+\Pr(t)+\Pr(t)\sum_{i=1}^{m}\Pr(C_i\mid t)\log(C_i\mid t)+\Pr(\bar{t})\sum_{i=1}^{m}\Pr(C_i\mid\bar{t})\log\Pr(C_i\mid\bar{t})$
(7)	$\chi^2(j,i)$	$\frac{N*(AD-CB)^2}{(A+C)*(B+D)*(A+B)*(C*D)}$
(8)	$BinaryFeaure$	*1(Yes), 0(NO)*

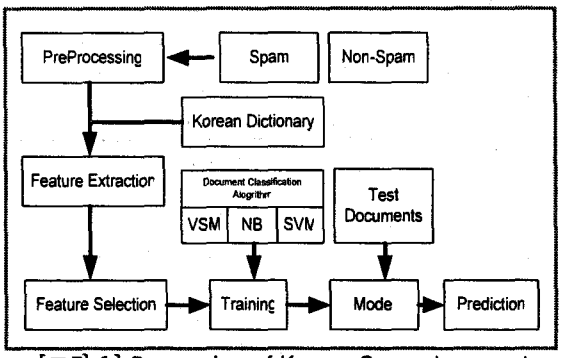
Table 1의 식(5)~식(7)은 Feature Selection 알고리즘이다. 그리고 이들은 또한 Feature Weighting으로도 사용할 수 있다. Table 1의 식(3)은 Vector Space Model의 Term Weighting이다. 특히, Table 1의 식(8)은 Binary Feature 로써 Term의 존재 여부에 따라 Yes/No로 나타낸다. Table 1의 식(2)와 식(4)는 위의 Term Weighting Variance 알고리즘을 적용한 Term Weighting이다. 우리는 Table 1의 8개 Term Weighting 을 Vector Space Model[12], Naïve Bayes[7] 그리고 Support Vector

Machine[8]에 적용하여 그 성능 결과를 보여준다.

6. 한국어 스팸 문서 분류

[그림 1]은 한국어 문서 분류 과정을 보여준다. 그림에서 우리는 3장에서 설명했었던 것처럼 한국어 Spam/Non-Spam 문서에 대한 전처리 과정을 진행하고 한국어 사전을 이용하여 명사를 추출한다. Feature Extraction은 추출된 명사에 대하여 [표 1]과 같이 Term Weighting 을 계산한 후, 식(1)과 결합하여 3.1장의 Feature Selection 알고리즘을 적용하여 Feature를 선택한다.

이제까지의 작업과정을 Training set에 적용하여 각 분류 알고리즘에 적용하여 학습을 시킨다. 그리고 새로운 쿼리가 들어왔을 때, 이를 이용하여 분류한다.



[그림 1] Processing of Korean Spam document Filtering

7. 실험 결과

기본적으로 우리는 학습과 테스트에 대해 5-fold cross validation을 이용하고 총 2000개의 Data Set을 이용하여 각 분류 알고리즘, Feature Selection, feature weighting에 대한 결과(Recall, Precision, F-Measure)를 보여준다. 먼저 우리는 Vector Space Model의 경우에는 MSV의 Global Feature Selection을 사용할 때 Recall(90%), Precision(90.7%), F-Measure(90.3%)의 결과를 보였다. Vector Space Model은 그러나 다른 두 개의 분류 알고리즘 보다 낮은 성능을 보였다. 그리고 Table 1의 여러 Feature를 적용할 수 있는 Naïve Bayes와 Support Vector Machine과는 달리 Table 1의 식3만 적용할 수 있다. 따라서 우리는 다음부터 다양한 Feature Weighting과 Feature Selection에 대한 Naïve Bayes와 Support Vector Machine의 다양한 결과를 보여준다.

[표 2]의 결과를 보면, Naïve Bayes 알고리즘을 적용할 때 가장 좋은 성능을 보여준다. 즉, TF*IDF Feature Weighting 과 IG를 적용하고 Feature Number 2000개일 때, Recall(98.8%), Precision(88.2%) 그리고 F-Measure(93.2%)의 성능 결과를 보였다.

표 2. Naive Bayes Classifier, TF*IDF Feature Weighting, Information Gain Feature Selection을 사용할 때의 결과

Feature Number	Recall	Precision	F-Measure
All	86	97.3	91.3
5000	86	97.3	91.3
4000	86	97.3	91.3
3000	86	97.3	91.3
2000	**98.8**	**88.2**	**93.2**

[표 3]의 결과를 보면 Feature Number 4000개 일 때, Recall(99.4%), Precision(97.4%) 그리고 F-Measure(98.4%)의 성능 결과를 보였다. 이때의 결과는 Support Vector Machine을 사용할 때 가장 좋은 성능이다.

표 3. Support Vector Machine Classifier(RBF Kernel), TF*IDF Variance Feature Weighting, CHI Feature Selection을 사용할 때의 결과

Feature Number	Recall	Precision	F-Measure
All	88.2	94.4	91.2
5000	97.8	89.7	93.6
4000	**99.4**	**97.4**	**98.4**
3000	97.8	90.6	94.1
2000	97.2	89.3	93.1

이제까지 우리는 한국어 스팸 문서 성능 결과에 대해 알아 보았다. 그 결과 Naïve Bayes Classifier사용할 때는 Information Gain Feature Selection(Feature Number=4000), TF*IDF Variance Feature Weighting을 이용할 경우 가장 높은 성능을 보였다. 그 때의 성능 결과는 Recall(98.6%), Precision(89%) 그리고 F-Measure(93.6%)를 보였다. 반면에Support Vector Machine Classifier를 이용할 때는 CHI Feature Selection(Feature Number=4000)과 TF*IDF Variance의 경우에 가장 높은 성능을 보였다. 그때의 성능 결과는 Recall(99.4%), Precision(97.4%) 그리고 F-Measure(98.4%)의 결과를 보였다. 또한 이때의 결과는 3개의 Classifier중에서 가장 높은 성능을 보였다. 즉, Support Vector Machine>Naïve Bayes> Vector Space Model순의 결과를 보였다.

8. 결론

우리는 이제까지 Korean Spam Document Filtering에 대한 연구에 대해 기술하였다. 이를 위해 다양한 Feature

Weighting, Feature Selection과 3가지의 분류 알고리즘을 이용하였다. 그 결과 우리는 Variance기반 Feature Weighting과 Support Vector Machine Classifier를 이용할 때, Recall(99.4%), Precision(97.4%) 그리고 F-Measure(98.4%)의 성능 결과를 보였다. 또한 다른 연구[9]의 Feature들을 이용할 때 그 결과를 비교한 결과 우리의 연구가 좀더 좋은 성능을 보였음을 증명하였다.

9. 참고 문헌

[1]Fu, Z. and Sarac, I.: A Computational Study of Naive Bayesian Learning in Anti-spam Management. SSPR&SPR 2004, LNCS 3138, pp. 824-830 (2004)

[2]Shrestha, R. and Yaping Lin, Y.: Improved Bayesian Spam Filtering Based on Co-weighted Multi-area Information. PAKDD 2005, LNAI 3518, pp. 650-660 (2005)

[3]Iwanaga, M., Tabata, T. and Sakurai, K.: Some Fitting of Naive Bayesian Spam Filtering for Japanese Environment. WISA 2004, LNCS 3325, pp. 135-143 (2004)

[4] Stuart, I., Cha, S.H. and Tappert, C.: A Neural Network Classifier for Junk E-Mail. DAS 2004, LNCS 3163, pp. 442-450 (2004)

[5]Salton, G.,Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing & Management 24(5), pp.513-523 (1988)

[6]Drucker, H., Wu, D. and Vapnik, V.: Support Vector Machines for Spam Categorization.. IEEE Trans. on Neural Networks , vol 10, number 5, pp. 1048-1054 (1999)

[7]McCallum, A., Nigam, K.: A Comparison of Even Models for Naïve Bayes Text Classifica-tion: in AAAI 98 Workshop on Learning for Text Categorization (1998)

[8]Vanpik,V.: Statistical learning theory. Willy, New York (1998)