3-бөлім Раздел 3 Section 3

Информатика Информатика Computer science

UDC 004.622, 004.623, 004.855.5

Aubakirov S.S.[1*], Akhmed-Zaki D.Zh.[1], Trigo P.S.[2]

[1]Al-Farabi Kazakh National University,
Republic of Kazakhstan, Almaty
[2]Instituto Superior de Engenharia de Lisboa, Biosystems and Integrative Sciences Institute
Agent and Systems Modeling, Lisbon, Portugal
* E-mail: aubakirov.sanzhar@gmail.com

News Classification using Apache Lucene

In this paper, we describe the binary classification of textual messages using Apache Lucene. We first describe the problem of "news classification" and the method to gather the dataset samples used for models' training and testing. We focus on creating and training classifiers based on different indexes and models' evaluation metrics. We choose three main attributes that affects Apache Lucene indexes, and as a consequence affects accuracy of classifiers. One of the attributes are words on which it is built. We use Ngram concept instead of words, where different N varies from one to five. Second attribute is text preprocessing method, such as stemming and stop words filtering. Third attribute is classification method. In this research we choose k-nearest neighbors and naive Bayes classification methods. Variation of this attributes produces various index types. As a result, we provide analysis of the correlation between index type and the accuracy of classifiers.
Key words: Binary classification, learning algorithms, Apache Lucene.

Аубакиров С.С., Ахмед-Заки Д.Ж., Триго П.С.

Apache Lucene көмегімен жаңалықтар классификациясы

Бұл мақалада Apache Lucene қалыптамасы көрсеткіштерінің негізінде мәтіндік хабарламаларды бинарлық топтау есебі қарастырылған. Нақты уақыт бойынша түсетін мәтіндік жаңалықтарды топтау есебі қойылған. Оқытуды талдап және тестілеп алудың әдістемелері, сондай-ақ топтау дәлдігін бағалайтын әдістемелер өркендетілген. Зерттеу барысында Apache Lucene көрсеткіштеріне және топтау дәлдігіне салдар ретінде ықпал ететін негізгі үш атрибут таңдап алынған. Көрсеткіштердің өзі сөздерден құрылатындықтан атибуттың бәрі сөз болып табылады. Біз сөздердің орнына Ngram ұғымын пайдаланамыз, мұндағы N бірден беске дейін ауытқиды. Екінші атрибут – мәтінді алдын ала өңдеу әдісі, дәлірек айтсақ мәтінді қалыптау және тоқтау-сөздерді сұрыптау. Ал үшінші атрибут – топтастыру пішінін құру алгоритмі. Зерттеу кезінде біз төмендегі екі алгоритмді таңдап алдық: «K – ең жақын көршілер» және «Ангалдық байестік топтаушы» әдістерімен топтау. Алғашқы екі атрибуттың өзгерісі көрсеткіш қасиеттерінің өзгерісіне әкеледі, нәтижесінде әр түрлі көрсеткіш түрлері қалыптасады. Жұмыста топтаушыларды оқытуда олардың көрсеткіштерге байланысы мен құрастырудың практикалық жағынан жүзеге асыру жағдайлары қарастырылған. Көрсеткіш түрлерінің топтаушылар дәлдігіне ықпал етуіне талдау жүргізілген.
Түйінді сөздер: Бинарлы классификация, оқыту алгоритмдер, Apache Lucene.

Аубакиров С.С., Ахмед-Заки Д.Ж., Триго П.С.

Классификация новостей при помощи Apache Lucene

В данной статье рассматривается задача бинарной классификации текстовых сообщений на базе индексов платформы Apache Lucene. Сформулирована задача классификации текстовых новостей, поступающих в режиме реального времени. Разработаны методы получения тестовой и обучающей выборки, а также методы оценки точности классификации. Для исследования были выбраны три основных атрибута, влияющих на индексы Apache Lucene и, как следствие, на точность классификаторов. Одним из атрибутов индексов являются слова, на основе которых они построены. Мы используем понятие Ngram вместо слов, где число N варьируются от одного до пяти. Второй атрибут - метод предварительной обработки текста, а именно нормализация текста и фильтрация стоп-слов. Третий атрибут - алгоритм построения модели классификации. Для данного исследования мы выбрали два алгоритма: классификация методом “К-ближайших соседей” и “наивный байесовский классификатор”. Изменение первых двух атрибутов приводит к изменению свойств индекса и, как результат, к формированию различных типов индексов. В работе рассмотрена практическая реализация создания и обучения классификаторов в зависимости от типа индексов. Проведен анализ влияния типа индексов на точность классификаторов.

Ключевые слова: Бинарная классификация, алгоритмы обучения, Apache Lucene.

1 Introduction

In this paper we address the task of evaluating binary classification of social media messages and analyzing the relationship between classification quality and classifier parameters.

We consider two classes of messages: NEWS and NOTIFICATION. We consider that the NOTIFICATION class is applied to messages with information about emergencies and hazards. Those messages are so important that people should be notified about them.

In order to collect messages and build datasets (for training and testing) we designed a web crawler with a simple choice mechanism based on a decision tree over the notion of "strong keywords". The decision tree classifies each message using three classes: NEWS, NOTIFICATION and NOT_SURE. The NOT_SURE class is applied to messages that are neither classified as NEWS nor as NOTIFACTION. This approach guarantees high Precision and low Recall but it is essential to collect the training dataset. The classifiers investigated in this paper are trained with a dataset built from the NEWS and NOTIFICATION classes. The NOT_SURE class is used as the testing dataset.

We are using Apache Lucene with the classification package as the main framework. The Lucene classifiers works with vector models and the TF-IDF metric. The evaluation resorts to the k-fold cross validation method. The Lucene classification based on indexes, in order to configure classifier Lucene allows changing index parameters. All indexes has Analyzer, it is consists of Tokenizer and set of Filters. Tokenizer extract tokens from text stream, Filters processes tokens, normalize and do stemming. Classifiers also depends on Analyzer that is why it is very import to have similar Analyzer for both classifier and index.

Analyzers combined using different Filters and Tokenizers. We are using Analyzers configured using minShingleSize, maxShingleSize, SnowballFilter and LengthFilter in ShingleAnalyzerWrapper. Parameters minShingleSize and maxShingleSize adjust the minimum and maximum value of N, where N is size of Ngrams. In our research, we are using up to fivegram indexes and one index with all Ngrams from one to five. Additionally

Analyzers combined by turning on and off SnowballFilter and LengthFilter. As a result, we are using 24 Analyzers.

Furthermore, each classifier has parameters that affects classification quality. KNN can be configured by varying K. Our experiments shows that KNN classifiers quality dramatically low for K>200, that is why we decide to bound K from 1 to 200 [1]. The last parameter is message fields. There is two fields available for indexing: title and body. Based on fields we can combine three different indexes: title, body and both.

Totally, we have combination of 24 Analyzers, 201 classifiers and 3 fields and it is 14472 different classifiers totally. In the next chapter, we showing quality of learning and analyzing relations between Ngrams, indexes and quality of classification. This paper describes how we evaluate 14472 classifiers and provide detailed analysis of correlation between index type and learning accuracy.

2 Method

According to researches [2, 3] we choose Kappa and MCC coefficient measures to evaluate binary classification learning. Both, Kappa and MCC, take into account true and false positives and negatives and well balance. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and −1 indicates total disagreement between prediction and observation. Both coefficients perfectly fit our needs because we consider NOTIFICATION true-positives more relevant than false-negatives; i.e., we care more about correctly classifying messages as NOTIFICATION than badly as NEWS.

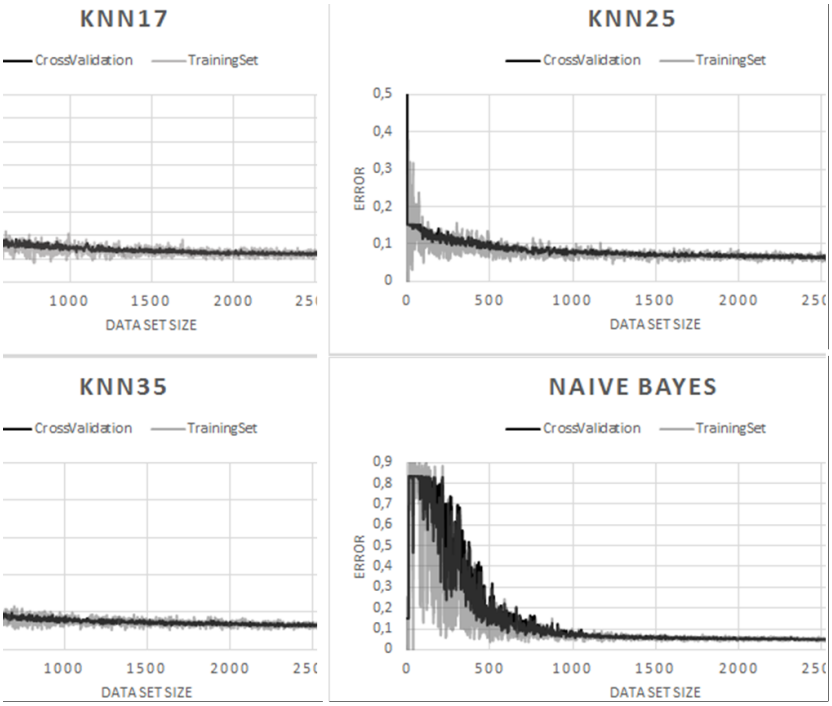

Figure 1 – Correlation between error rate and dataset size for KNN and Naive Bayes classifiers

Researches [4, 5] shows that the "Learning Curve"(LC) methodology is a good diagnosing tool, telling you how fast your model learns and whether your whole analysis is, or not, biased

by the dataset cardinality (e.g., overfitting a too small dataset). We use the LC to determine the smallest size of training that shows reasonable results. The learning curves are depicted in the Figure 1. The error rate is validated using k-fold cross validation method.

The graph shows that the accuracy of learning algorithms is aligned on the data set size on the interval 1500-2000. We choose 2270 news messages as a training set, all messages was classified using a decision tree described above. Finally, classified messages were manually checked and corrected by the experts. It is important to mention that the dataset consists of 84% NEWS messages and 16% NOTIFICATION messages. The k-fold cross validation follows that same stratification.

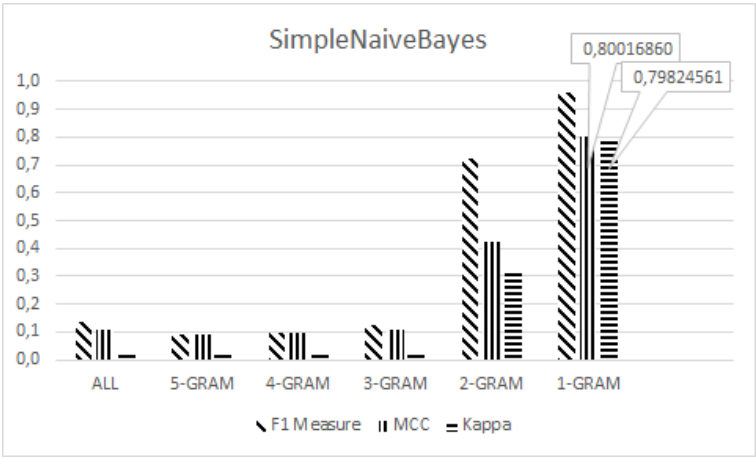

Figure 2 – Naive Bayes classifier using only body field, Snowball stemmer and short tokens filtering

3 Results

We measure each classifier using k-fold cross validation method. Our experiments show that 87% of the learned models exhibit a Kappa and a MCC less than 0.5, which, according to Landis and Koch [2] can be considered as "poor"models.

The best results shows classifiers that is using unigrams tokenizer, only body field, Snowball stemmer and with short tokens filtering. Best results had shown on the Figure 2, 3 and 4.

Top five classifiers use unigram indexes, only body field, Snowball stemmer and short token filtering. Naive Bayes classifier best results: MCC=0.8, Kappa=0.79, Figure 5 shows its confusion matrix. There are 93 NEWS messages classified as NOTIFICATION (False Negatives - FN) and 45 NOTIFICATION messages classified as NEWS (False Positives - FP). F-measure often shows values close to 1, this is because F-measure calculation ignores FN errors. That is why we do not use it as evaluation measure, because our goal main is to minimize both FP and FN. Figure 3 shows that KNN algorithm, with K=17, best result is MCC=0.68, Kappa=0.67. Furthermore, increasing number of neighborhoods leads to decreasing accuracy of classifier.

Our experiments shows that classifiers based on indexes with Ngrams with N>1 shows low accuracy: MCC<0.5, Kappa<0.5.

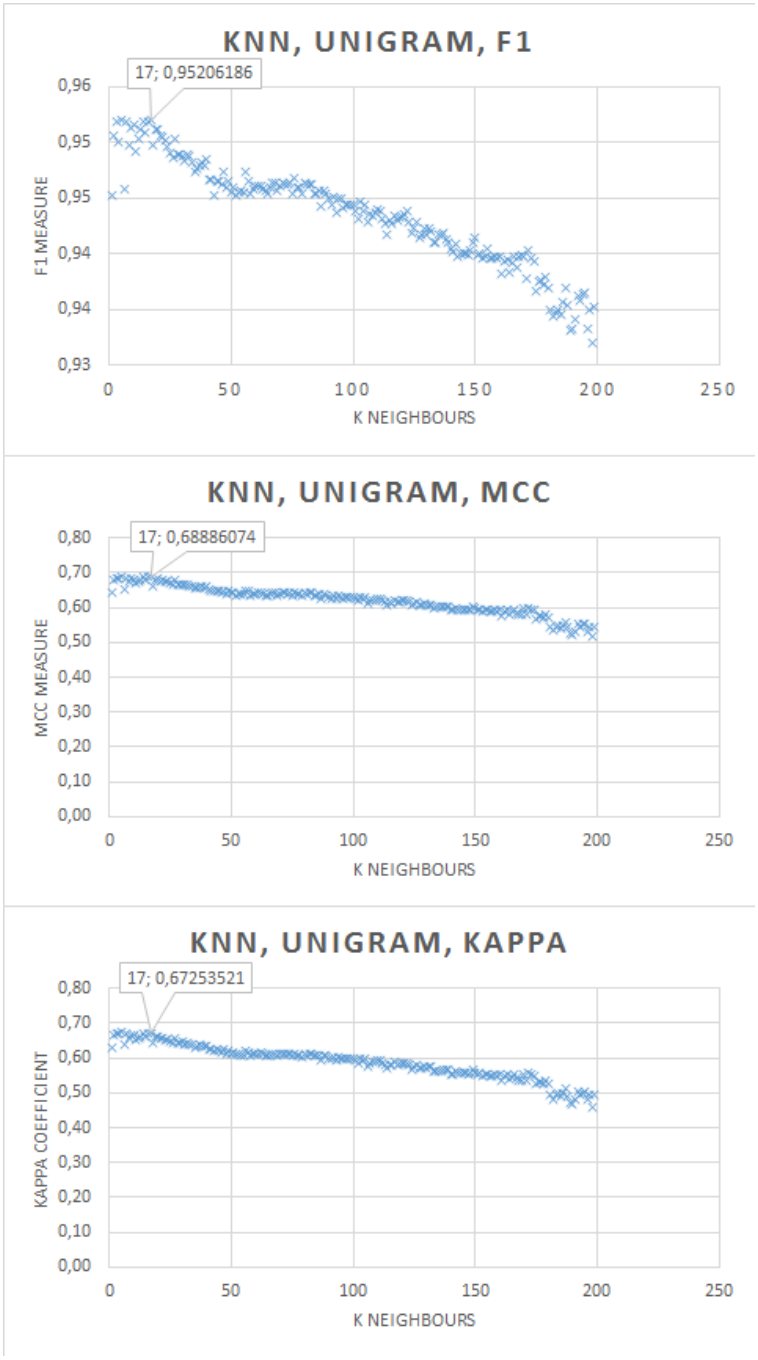

Figure 3 – Results for KNN classifier with varying K, unigram indexes, only body field, Snowball stemmer and short token filtering

4 Discussion

Our research shows that classification quality depends on Apache Lucene index type and n-gram size. The best classifier is based on unigrams as shown by the Kappa coefficient of 0.79 and the MCC coefficient of 0.8. It is expectable that a single classifier cannot achieve higher accuracy. As a future work, we are planning to implement a Voting classifier that combines

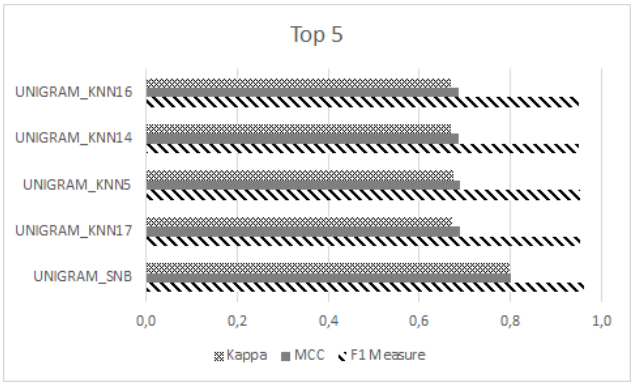

Figure 4 – The best five classifiers. All classifiers use Snowball stemmer, unigram indexes, only body field and short token filtering

Class names was shortened:
C0 NEWS
C1 NOTIFICATION

Confusion matrix:

	C0	C1
C0	350	45
C1	93	1782

Observed Accuracy:	0.9392070484581497
Expected Accuracy:	0.6986784140969163
Error rate:	0.06079295154185022
Kappa statistics:	0.7982456140350875
Precision:	0.8860759493670886
Recall:	0.7900677200902935
F measure:	0.8353221957040573
MCC :	0.8001686011904481
Total Docs #	2270

Figure 5 – The best classifiers confusion matrix. Naive Bayes based on unigram indexes

an odd number of classifiers in order to obtain a joint decision-making algorithm. To combine classifiers and evaluate all possible combinations we are planning to use distributed computing technologies, such as Message Passing Interface or Apache Spark. We also intend to explore the Ada Boost meta-algorithm in conjunction with other learning algorithms. The goal of this approach is too combine several weak classifiers into a weighted sum that represents one boosted classifier.

This work was partially supported by the grant of the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan (project 3350/GF4 MON RK).

References

[1] *Markus M., Matthias H., Ulrike H.* Optimal construction of k-nearest neighbor graphs for identifying noisy clusters. // Theoretical Computer Science. – 2009. – № 410(19). – P. 1749–1764.

[2] *David M W Powers.* Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. // Journal of Machine Learning Technologies. – 2011. – Vol. 2 (1). – P. 37–63.

[3] *Landis J.R., Koch G.G.* The Measurement of Observer Agreement for Categorical Data. // Biometrics. – 1977. – Vol. 33. – № 1. – P. 159–174.

[4] *Saman M., Robert S.* The Learning Curve and Optimal Production under Uncertainty. // Rand Journal of Economics. – 1987. – Vol. 20. – № 3. – P. 331–343.

[5] *Christopher M., Bo T., David H.* The Learning Curve Sampling Method Applied to Method Based Clustering. // Journal of Machine Learning Research 2S. – 2002. – P. 397–418.