

NAMED ENTITY RECOGNITION ON A COLLECTION OF RESEARCH TITLES

Siti Mariyah

The Center of Computational Statistics Study, Institute of Statistics, Jakarta-Indonesia 13330
e-mail: sitimariyah@stis.ac.id

Abstrak

Judul dapat membantu pembaca untuk mendapatkan sudut pandang universal dari artikel tersebut sebagai pemahaman awal sebelum membaca konten secara keseluruhan. Pada penelitian teknis, judul memuat informasi penting. Dalam penelitian ini, kami mengembangkan teknik ekstraksi informasi untuk mengenali dan mengekstrak masalah, metode, dan domain penelitian yang terdapat dalam judul. Kami menerapkan pendekatan supervised learning pada 671 judul penelitian dalam bidang ilmu komputer dari beragam jurnal online dan prosiding seminar internasional. Kami melakukan beberapa percobaan dengan skema yang berbeda untuk mempelajari pengaruh fitur dan kinerja algoritma. Kami menguji fitur kontekstual, fitur sintaksis, dan fitur bag of words menggunakan Naïve Bayes dan Maximum Entropy. Classifier Naïve Bayes yang belajar dari kelompok set fitur pertama berhasil memprediksi kategori masing-masing token dalam dataset judul. Keakuratan dan nilai f1-score untuk setiap kelas lebih dari 0,80 karena kelompok pertama set fitur mempertimbangkan lokasi token dalam sebuah kalimat, memperhatikan token sekitar dan tag POS dari beberapa token sebelum dan sesudah. Sementara classifier Naïve Bayes yang dipelajari dari kelompok kedua dari rangkaian fitur lebih tepat mengklasifikasikan token frase daripada token kata.

Kata Kunci: *research titles, named entity recognition, information extraction, contextual features, naïve bayes classifier*

Abstract

The title can help the reader to get the universal point of view of the article as the initial understanding before reading the content as a whole. On technical research papers, the title states essential information. In this study, we aim to develop information extraction techniques to recognize and extract problem, method, and domain of research contained in a title. We apply supervised learning on 671 research titles in computer science from various online journals and international conference proceedings. We conducted some experiments with different schemas to discover the influence of features and the performance of the algorithm. We examined contextual, syntactic, and the bag of words feature sets using Naïve Bayes and Maximum Entropy. The Naïve Bayes classifier learned from the first group of the feature set is successful in predicting category of each token in title dataset. The accuracy and f1-score for each class are more than 0.80 since the first group of feature sets considers the location of a token within a sentence, considers the token and POS tag of some tokens before and after and deliberates the rules of a token. While the Naïve Bayes classifier learned from the second group of the feature set is more appropriate classifying a phrase token than a word token.

Keywords: *research titles, named entity recognition, information extraction, contextual features, naïve bayes classifier*

INTRODUCTION

Research title is a short sentence which can help the reader to get the main or universal point of view of the article as the initial understanding before reading the content as a whole. The title is also commonly used as a filter in a search engine when there is a retrieval query against a research paper in online journals or online archives. On technical research papers such as in computer science or engineering, the title states essential information. That information consists of the research problem, the method used or method proposed, and the specific research domain. A reader or a researcher should know the problem, method, and domain of research regarding the topic she/he is studying or focusing.

On the other side, information extraction opens the opportunity to extract words or phrases that are regarded as informative words or phrases. Informative means that the word or phrase describes the information a reader want to know. Information extraction technique involves a collection of natural language processing (NLP) tasks. Each method may include different NLP task which depends on the complexity of information, the format of the document, and the task itself, etc. There are three approaches to build information extraction technique, i.e., rule-based extraction, statistical or machine learning-based extraction, or hybrid approach.

In this study, we aimed to develop information extraction techniques to recognize and extract problem, method, and domain of research contained in a title. We apply supervised learning as a part of statistical or machine learning-based approach on 671 research titles in computer science from ACM Digital Library, IEEE, and some international conference proceedings. By using some learning algorithms, we constructed some named entity recognition (NER) models. Machine learning based extraction can handle the knowledge acquisition bottleneck since, in rule-based extraction, we need to construct extraction rules which requires the domain experts. The NER model identifies the

property of each word in the title then classify it into some defined categories. We conducted some experiments with different schemas to learn the influence of features and the performance of the algorithm. In this paper, we technically describe how we built the information extraction techniques in detail and suggest some recommendations which one is the best feature and model.

LITERATURE REVIEW

NER was first introduced in the Sixth Message Understanding Conference (MUC-6) held in November 1995. Two of four goals are named entity recognition and scenario templates (traditional information extraction). NER task comprises the recognition of entity names of people, names of company or organization, place names, temporal expressions and a particular type of numerical expressions.

Suakkaphong et al. (2009) built disease named entity recognizer. They used three feature sets. The first feature set is a morphological-pattern feature since biomedical terms commonly have unique prefixes and suffixes. The remaining features are word appearance and chunking and POS tag features. Then, They combined conditional random field (CRF) with bootstrapping and feature sampling. CRFs with bootstrapping implemented sequentially is more accurate than supervised CRFs.

Biomedical named entity recognition was also done by Saha et al. (2009) and Bodenreider et al. (2000). They hypothesized that the appropriate feature templates affect the performance of NER models. They conducted word clustering and selection based feature reduction approaches for NER using Maximum Entropy algorithm. The feature sets are generated without involving profound biomedical knowledge such as word feature, previous NE tags, capitalization and digit information, unique character, word normalization, prefix and suffix information, Part of Speech (POS) tags, and trigger words. They proved that the use of

dimensionality reduction techniques could increase the performance substantially.

Bodenreider, Olivier, and Pierre Zweigenbaum (2000) developed methods to collect proper names used in biomedical terminology. The task is recognizing a word that is the appropriate name by using individual criteria owned by that word and some combination of these different criteria (capitalization, invariant words, and patterns).

Another relevant work was done by Ek et al. (2011) who conducted NER for short text messages. The characteristics of the short text message are similar to title sentence which has small windows (a few of words). They constructed regular expression and complemented with logistic regression classifier. Wu et al. (2005) used POS tag as feature set. Researches of McKenzie (2013), Mao, Xinnian et al. (2007) and Qin et al. (2008) utilized the contextual feature sets to either improve the NER results in the large-scale corpus or to reduce the noise introduced into aggregated features from disparate and generic training data. They proved that the missed entities occur when their contextual surroundings are not identified well. NER using machine learning approach are more frequent conducted than other methods. There are learning algorithms applied for NER or text classification tasks such Naïve Bayes or Multinomial Naïve Bayes performed by Fabrizio Sebastiani (2001) and Amarappa S, and Sathyaranayana S.V. (2015), Maximum Entropy applied by Ayan et al. (2006), Conditional Random Fields performed by Mao, Xinnian et al. (2007), Qin et al. (2008), and Chodey et al. (2016), Support Vector Machines applied by Fabrizio Sebastiani (2001), Thorsten Joachims (1998), and Rafi et al. (2012).

METHODS

Extraction technique was developed by involving some tasks depicted by this following diagram:

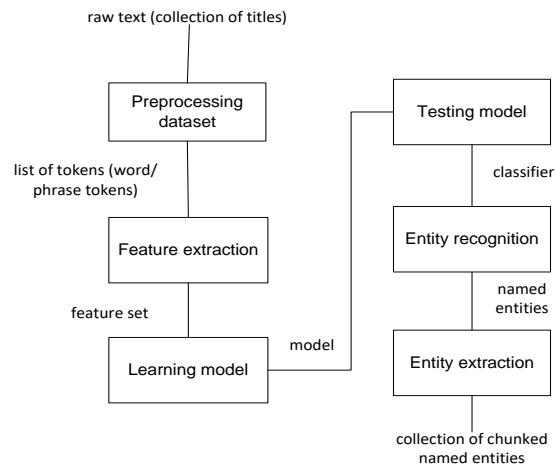


Figure 1. Extraction techniques

It starts from collecting dataset. We gathered 671 research titles in computer science fields from some online journals or online archives. Then the dataset will be processed in some following tasks:

1. Preprocessing dataset

The dataset was validated to ensure there were no double titles. Then we conducted annotation on the dataset to tag the words or the phrases which explain problem, method, and domain of research. Annotation was done by humans who are familiar with computer science research. We tagged <m>...</m> for words explaining method, <p>...</p> for words explaining problem, and <d>...</d> for words explaining domain. The annotated dataset was then validated to make sure that there was no missed annotation or wrong annotation. The missed annotation means that there is a token that is not annotated. The wrong annotation means that there is a token annotated by the wrong label. By using the regular expression, we split the annotated dataset into four files. Each file contains 671 lines where each line contains the words in one category. It aimed to check whether every title contains full information (problem, method, and domain) or not.

Then, we tokenized every title sentence, made part-of-speech-tag (POS tag) for each token and mapped token with the label it owns. We labeled m for tokens flanked by <m>...</m> tag, p for tokens flanked by <p>...</p> tag, d for tokens flanked by <d>...</d> tag, and none for

tokens not flanked by any tag. Output in this step is collection of tokens per sentence who have its each label. We focused and used the word tokens only rather than the phrase tokens.

2. Feature extraction

The output from processing dataset stage is the input for this feature extraction step. The feature is information which characterizes a token. The features used significantly affect the accuracy of the classification model. We were curious which features accurately differentiate each category. We extracted some features and grouped it into two groups of the feature set. Then, these two groups would be tested with some experiments to know which group is the most relevant.

The first group of feature set:

1. Feature word: the token itself
2. Feature POS tag
3. Feature prevWord: one token before
4. Feature prevTag: POS tag of one token before
5. Feature prevBigram: two tokens before
6. Feature prevBigramTag: POS tag of two tokens before
7. Feature nextWord: one token after
8. Feature nextBigram: two tokens after
9. Feature nextTag: POS tag of one token after
10. Feature nextBigram: POS tag of two tokens after

The second group of feature set was the list resulted matching the existence of a token in a collection of the method, problem, and domain tokens. If a token exists in that collection, then the value is true. Otherwise, the value is false. The number of extracted features equals the number of tokens owned in 671 research titles. This is the example of how to extract this feature set:

The first title:

```
<m>simple algorithms</m> for <p>complex relation extraction</p> with applications to <d>biomedical ie</d>
```

The second title:

```
<m>a seed-driven bottom-up machine learning</m> framework for <p>extracting relations of various complexity</p>
```

Therefore, The method, problem, domain and none tokens are:

Method tokens: simple algorithms, a seed-driven bottom-up machine learning
Problem tokens: complex relation extraction, extracting relations of various complexity
Domain tokens: biomedical ie
None tokens: for with application to, framework for

If want to extract feature from phrase “extracting relations of various complexity”, the extracted feature is:

```
{ contain(simple): False, contain(algorithms): False, contain(a): False, contain(seed-driven): False, contain(bottom-up): False, contain(machine): False, contain(learning): False, contain(complex): False, contain(relation): False, contain(extraction): False, contain(extracting): True, contain(relations): True, contain(of): True, contain(various): True, contain(complexity): True, contain(biomedical): False, contain(ie): False, contain(for): False, contain(with): False, contain(application): False, contain(to): False, contain(framework): False, contain(for): False }  
If want to extract feature from phrase “biomedical ie”, the extracted feature is:  
{ contain(simple): False, contain(algorithms): False, contain(a): False, contain(seed-driven): False, contain(bottom-up): False, contain(machine): False, contain(learning): False, contain(complex): False, contain(extraction): False, contain(relation): False, contain(extracting): False, contain(relations): False, contain(of): False, contain(various): False, contain(complexity): False, contain(biomedical): True, contain(ie): True, contain(for): False, contain(with): False, contain(application): False, contain(to): False, contain(framework): False, contain(for): False }
```

3. Learning and testing model

In this stage, we prepared training set. The training set is a collection of extracted feature for each token in dataset then mapped with the label owned by the token. If in title dataset consists of 1000 tokens then we have 1000 feature set mapped with the label. We applied Naïve Bayes, Maximum Entropy, and Support Vector Machines using two groups of the feature set with shuffling parameter. The classification models were learned and tested by 10-fold cross-validation. We measured precision, recall, and f-measure for each category to understand the effect of shuffling parameter, the performance of feature set and algorithm.

4. Entity recognition and extraction

The best model is then used as a classifier which recognizes and classify every token in title sentences into problem, method, domain or none category. If any token in sentence classified as a problem,

method, or domain category, our program then chunked the sentence into tokens and extracted those tokens.

EXPERIMENTAL STUDY

We conducted some experiments with some different conditions. The difference is defined by feature set used, shuffling

parameter and machine learning algorithm applied.

1. The first experiment

On the first experiment, we built classification model using the first group of feature set and Naïve Bayes algorithm. We applied 10-fold cross validation both on shuffled and non-shuffled training data. Shuffling the training data cause the order of the data to be random. The results are:

```
sperimen1-NaiveBayes-FiturSetKelompokPertama
C:\Python34\python.exe C:/Python34/Eksperimen1-NaiveBayes-FiturSetKelompokPertama.py
accuracy NaiveBayes per iterasi: 0.6711509715994021
accuracy NaiveBayes per iterasi: 0.8697604790419161
accuracy NaiveBayes per iterasi: 0.8532934131736527
accuracy NaiveBayes per iterasi: 0.8877245508982036
accuracy NaiveBayes per iterasi: 0.9041916167664671
accuracy NaiveBayes per iterasi: 0.8922155688622755
accuracy NaiveBayes per iterasi: 0.8682634730538922
accuracy NaiveBayes per iterasi: 0.8712574850299402
accuracy NaiveBayes per iterasi: 0.8083832335329342
accuracy NaiveBayes per iterasi: 0.7005988023952096
Akurasi Classifier dengan 10 fold Cross Validation: 0.8326839594353892
method precision: 0.6351931330472103
method recall: 0.7184466019417476
method F-measures: 0.6742596810933941
problem precision: 0.7061855670103093
problem recall: 0.6523809523809524
problem F-measures: 0.6782178217821784
domain precision: 0.39215686274509803
domain recall: 0.7407407407407407
domain F-measures: 0.5128205128205129
none precision: 0.8578947368421053
none recall: 0.7244444444444444
none F-measures: 0.7855421686746988
Hasil klasifikasi: None
Hasil klasifikasi potongan kalimat: ['m', 'None']
Hasil klasifikasi kalimat utuh: ['m', 'm', 'None', 'p', 'p', 'p', 'None', 'None', 'd', 'd', 'None']

Process finished with exit code 0
```

Figure 2. Learning performance on first group of feature set using Naïve Bayes with shuffle

```
sperimen1-NaiveBayes-FiturSetKelompokPertama
C:\Python34\python.exe C:/Python34/Eksperimen1-NaiveBayes-FiturSetKelompokPertama.py
accuracy NaiveBayes per iterasi: 0.8690565471726414
accuracy NaiveBayes per iterasi: 0.8689975990396158
accuracy NaiveBayes per iterasi: 0.8690958164642375
accuracy NaiveBayes per iterasi: 0.8688303918635956
accuracy NaiveBayes per iterasi: 0.8696888090963495
accuracy NaiveBayes per iterasi: 0.8689385978081369
accuracy NaiveBayes per iterasi: 0.8692134831460674
accuracy NaiveBayes per iterasi: 0.8694091249065071
accuracy NaiveBayes per iterasi: 0.8692918101512203
accuracy NaiveBayes per iterasi: 0.8693309384822632
Akurasi Classifier dengan 10 fold Cross Validation: 0.8691853118130636
method precision: 0.8373071528751753
method recall: 0.8919322709163346
method F-measures: 0.8637569327224499
problem precision: 0.8971109984794728
problem recall: 0.8575581395348837
problem F-measures: 0.8768887787961357
domain precision: 0.604735883424408
domain recall: 0.8736842105263158
domain F-measures: 0.7147470398277718
none precision: 0.948019801980198
none recall: 0.8591296545536115
none F-measures: 0.9013885620145916
Hasil klasifikasi: None
Hasil klasifikasi potongan kalimat: ['m', 'None']
Hasil klasifikasi kalimat utuh: ['m', 'm', 'None', 'p', 'p', 'p', 'None', 'None', 'd', 'd', 'None']

Process finished with exit code 0
```

Figure 3. Learning performance on first group of feature set using Naïve Bayes with no shuffle

Table 1. The comparison of shuffle and no shuffle condition on first group of feature set using Naïve Bayes algorithm

| The measurements | Without shuffle | With shuffle |
|---------------------|-----------------|--------------|
| Classifier accuracy | 0.83268 | 0.86919 |
| Method precision | 0.63519 | 0.83730 |
| Method recall | 0.71845 | 0.89193 |
| Method F-Measures | 0.67426 | 0.86376 |
| Problem precision | 0.70618 | 0.89711 |
| Problem recall | 0.65238 | 0.85755 |
| Problem F-Measures | 0.67821 | 0.87688 |
| Domain precision | 0.39216 | 0.64047 |
| Domain recall | 0.74074 | 0.87368 |
| Domain F-Measures | 0.51282 | 0.71475 |
| None precision | 0.85789 | 0.94802 |
| None recall | 0.72444 | 0.85913 |
| None F-Measures | 0.78554 | 0.90139 |

The table shows that the shuffle parameter causes the difference of classifier accuracy 0.03. It is aligned with the concept of fold cross validation which at every iteration, it divides the data into ten parts with nine parts as training and one as a testing set. The repetition is done until all elements have been a test set. The shuffle can affect the sampling of those parts. Our hypothesis is shuffle will minimize the probability a label does not appear in training set. It means that with shuffle, the distribution of the existence of each label is equal. Without shuffle, the process building up the members of 10 parts is done sequentially. Therefore, the probability of skewed distribution of category is higher.

Overall, recall values for all categories are above 0.85, and the difference of recall for each class is not significant. The precision values for the method, problem, and domain are 0.83730, 0.89711, and 0.64047. The precision for domain category is lower than others because the true positive is higher and false positive. After we evaluated the training set, the number of domain examples is more inferior than method and problem examples.

2. The second experiment

On the second experiment, we built classification model using the second group of feature set and Naïve Bayes algorithm. We applied 10-fold cross validation both on shuffled and non-shuffled training data.

Table 2. The comparison of shuffle and no shuffle condition on the second group of feature set using Naïve Bayes algorithm

| The measurements | Without shuffle | With shuffle |
|---------------------|-----------------|--------------|
| Classifier accuracy | 0.81323 | 0.86500 |
| Method precision | 0.0 | 0.92843 |
| Method recall | None | 0.73472 |
| Method F-Measures | None | 0.82029 |
| Problem precision | 0.0 | 0.90636 |
| Problem recall | None | 0.77963 |
| Problem F-Measures | None | 0.83823 |
| Domain precision | 0.0 | 0.71909 |
| Domain recall | None | 0.98807 |
| Domain F-Measures | None | 0.83234 |
| None precision | 1 | 0.98609 |
| None recall | 0.83146 | 0.95795 |
| None F-Measures | 0.90798 | 0.97182 |

Table 2 tells the performance of classifier from the second group of the feature set without and with the shuffle. The result of this experiment is much different with the last experiment. Without shuffle, the classifier failed to detect a problem, method, and domain tokens. It is explained by the values of precision, recall, and f-measures for all categories. If compared with the same treatment (with shuffle), this classifier learned from the first group of feature set performs almost equal with the classifier acquired from the second group of the feature set.

Table 3. The comparison of the group of feature set using Naïve Bayes algorithm with shuffle

| The classifier | The First Group of Feature Set | The Second Group of Feature Set |
|---------------------|--------------------------------|---------------------------------|
| Classifier accuracy | 0.86919 | 0.86500 |
| Method F-Measures | 0.86376 | 0.82029 |
| Problem F-Measures | 0.87688 | 0.83823 |
| Domain F-Measures | 0.71475 | 0.83234 |
| None F-Measures | 0.90139 | 0.97182 |

Table 3 shows that the classifiers from two groups are almost similar. The first classifier is accurate for classifying method and problem tokens, while the second classifier is accurate for recognizing domain and none tokens. Our hypothesis is method and problem tokens are good explained with contextual and syntactic features. It means

that method and problem tokens may have regular tokens previous and after with regular POS tag.

3. The third experiment

On this experiment, we examined Maximum Entropy (MaxEnt) algorithm to validate the effect of different feature set on classifier. We trained the model with 10-fold cross validation and shuffle.

Table 4. The comparison of the group of feature set using maximum entropy algorithm with shuffle

| The classifier | The First Group of Feature Set | The Second Group of Feature Set |
|---------------------|--------------------------------|---------------------------------|
| Classifier accuracy | 0.83975 | 0.25216 |
| Method F-Measures | 0.86918 | None |
| Problem F-Measures | 0.84124 | None |
| Domain F-Measures | 0.01047 | 0.40059 |
| None F-Measures | 0.88192 | None |

Table 4 tells us that accuracy classifier on the first group around 83.975% is better than on the second group of the feature set. It is aligned with the f-measures for the method, problem, and none categories. The interesting one is MaxEnt fails to classify domain category using the first group off; we feature set. It is caused by precision value for domain is 1.0, but the recall is 0.00526. It means that coverage ability of MaxEnt classifier for domain category is low. MaxEnt also miscarries the second group of the feature set.

From three experiments conducted, we concluded that Naïve Bayes classifier is robust on both the first and the second group of feature sets. Naïve Bayes classifier with the first group of feature set outperforms than others. It also delivers informative features. The informative feature means that the feature is the most significant feature in determining a token belongs to a category. The shuffle improves the performance a classifier than it is not shuffled.

The first group of feature set consists of a word, tag, prevWord, prevTag, prevBigram, prevBigramTag, nextWord, nextTag, nextBigram, nextBigramTag. Using Naïve Bayes with shuffle and 10-fold

cross validation, the accuracy acquired is 0.86919. It means that 86,919% of test set will classified correctly. The following descriptions are the explanation for every informative feature.

| Most Informative Features | | |
|---------------------------|---|-------------|
| word = 'for' | = | 243.7 : 1.0 |
| prevWord = 'for' | = | 211.5 : 1.0 |
| prevBigram = '-' | = | 113.1 : 1.0 |
| word = 'using' | = | 83.2 : 1.0 |
| prevWord = '-' | = | 79.5 : 1.0 |
| word = '.' | = | 72.2 : 1.0 |
| nextWord = '-' | = | 70.0 : 1.0 |
| nextTag = '-' | = | 57.1 : 1.0 |
| word = 'a' | = | 40.9 : 1.0 |
| prevWord = 'using' | = | 40.2 : 1.0 |
| prevWord = 'from' | = | 38.8 : 1.0 |
| prevBigramTag = '-' | = | 37.3 : 1.0 |
| prevTag = '-' | = | 35.9 : 1.0 |
| prevWord = 'in' | = | 34.4 : 1.0 |
| nextBigramTag = 'IN JJ' | = | 30.5 : 1.0 |
| word = 'in' | = | 27.7 : 1.0 |
| nextTag = 'VBG' | = | 26.9 : 1.0 |
| prevBigramTag = 'NN NNS' | = | 26.5 : 1.0 |
| nextWord = 'for' | = | 25.8 : 1.0 |
| nextWord = 'using' | = | 24.8 : 1.0 |

Figure 4. The Most Informative Features from The First Group of Feature Set

- a. The word ‘for’ appears 243 times on none class than problem class. It explains the word ‘for’ has high probability to be classified as none category and not belongs to problem, domain, and domain classes.
- b. PrevWord = ‘for’ occurs 211 times on problem class than on method class. It means that a word or a phrase preceded by the word ‘for’ has high chance to be classified as problem class.
- c. The third (prevBigram = '-'), the fifth (prevWord = '-'), the twelfth (prevBigramTag = '-'), and the thirteenth information (prevTag = '-') explain that a token which does have any previous token is more frequent classified as method class than domain class. It indicates that a word or a phrase at the beginning of the title sentence has high chance to be classified as method class. It is aligned with the fact. We observed directly some title sentences which prove this information.

The first title: <m>simple algorithms</m> for <p>complex relation extraction</p> with applications to <d>biomedical ie</d>

The second title: <m>a seed-driven bottom-up machine learning</m> framework for <p>extracting relations of various complexity</p>

- d. The tenth information (`prevWord = 'using'`) appears 40 times on the method class than on the problem class. It shows that a word or a phrase preceded by the word 'using' has more chance to be classified as method class.
- e. The eleventh (`prevWord = 'from'`) and the fourteenth information (`prevWord = 'in'`) appear more than 30 times on domain class than on the method class. It describes that a word or a phrase preceded by the word 'from' or 'in' has a higher probability to be classified as domain class.
- f. The fifteenth (`nextBigramTag = 'IN JJ'`) occurs 30 times and the eighteenth (`prevBigramTag = 'NN NNS'`) appears 26 times on class method than on class domain. It indicates that a word or a phrase preceded by noun words will be classified as method class.
- g. The seventeenth (`nextTag = 'VBG'`) occurs 26 times more on domain class than problem class. It means that a word or a phrase followed by gerund (verb + 'ing') has a higher probability to be classified as domain class.
- h. The nineteenth (`nextWord = 'for'`) appears 25 times more on the method class and the twentieth information (`nextWord = 'using'`) occurs 20 times on problem class. It indicates that a word or a phrase followed by the word 'for' will be classified as method class and followed by the word 'using' has higher chance to be classified as problem class.

| Most Informative Features | | |
|---|---|-------------|
| <code>contain(for) = True</code> | = | 291.8 : 1.0 |
| <code>contain(using) = True</code> | = | 107.3 : 1.0 |
| <code>contain(in) = True</code> | = | 69.5 : 1.0 |
| <code>contain(a) = True</code> | = | 61.7 : 1.0 |
| <code>contain(extraction) = True</code> | = | 47.0 : 1.0 |
| <code>contain(and) = True</code> | = | 30.2 : 1.0 |
| <code>contain(classification) = True</code> | = | 26.3 : 1.0 |
| <code>contain(of) = True</code> | = | 24.6 : 1.0 |
| <code>contain(information) = True</code> | = | 22.2 : 1.0 |
| <code>contain(an) = True</code> | = | 21.7 : 1.0 |
| <code>contain(summarization) = True</code> | = | 20.3 : 1.0 |
| <code>contain(method) = True</code> | = | 17.7 : 1.0 |
| <code>contain(approach) = True</code> | = | 14.2 : 1.0 |
| <code>contain(knowledge) = True</code> | = | 13.0 : 1.0 |
| <code>contain(models) = True</code> | = | 12.3 : 1.0 |
| <code>contain(on) = True</code> | = | 12.1 : 1.0 |
| <code>contain(traffic) = True</code> | = | 11.0 : 1.0 |
| <code>contain(algorithm) = True</code> | = | 11.0 : 1.0 |
| <code>contain(detection) = True</code> | = | 11.0 : 1.0 |
| <code>contain(fuzzy) = True</code> | = | 9.7 : 1.0 |

Figure 5.The Most Informative Features from The Second Group of Feature Set

The picture tells about:

- a. If a word or a phrase is/contains a word 'for', 'using', 'in', 'a', or 'an', then the word or phrase has more chance to be classified as none class.
- b. If a word or a phrase is/contains a word 'extraction', 'classification', 'information', 'summarization', 'traffic', or 'detection', then the word or phrase has higher chance to be classified as problem class.
- c. If a word or a phrase is/contains a word 'method', 'approach', 'knowledge', 'models', 'algorithm', or 'fuzzy', then the word or phrase has more chance to be classified as method class.

We conducted significance test to examine two hypotheses. The first hypothesis is the performance of Naïve Bayes and MaxEnt classifier learned from the first group of feature set is same. The second hypothesis is the performance of two classifiers are different, one classifier is better than another. This is the significance test algorithm:

1. The data is partitioned into k disjoint test sets T_1, T_2, \dots, T_k with same size. The minimum size is 30.
2. For i from 1 to k , do # $k = 10$
Use T_i for the test set and the remaining data for training set S_i
 $S_i \leftarrow \{D_0 - T_i\}$ # S_i : training set
 $h_A \leftarrow L_A(S_i)$ # L_A : Naïve Bayes classifier
 $h_B \leftarrow L_B(S_i)$ # L_B : MaxEnt classifier
 $\delta_i \leftarrow errorT_i(h_A) - errorT_i(h_B)$
3. Return:

$$\bar{\delta} = \frac{1}{k} \sum_{i=1}^k \delta_i$$

The result of $\bar{\delta} = -0.029512$

Next step is measuring confidence interval. We took confidence interval 90% so that the confidence interval estimation for

$$\delta: \bar{\delta} \pm t_{N,k-1} s_{\bar{\delta}},$$

Where:

$$s_{\bar{\delta}} = \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^k (\delta_i - \bar{\delta})^2}$$

$$s_{\bar{\delta}} = \sqrt{\frac{1}{10(10-1)} \sum_{i=1}^{10} (\delta_i - (-0.029512))^2}$$

$$= 7,86554E - 05$$

The value $t_{N,k-1} = t_{90,9} = 1.833$ is acquired from t-table. The confidence interval is:

$$\begin{aligned} & -0.029512 \pm 1.833*(7,86554E-05) \\ & = -0.029512 \pm 0.000144 \end{aligned}$$

The upper limit of the interval:

$$-0.029512 + 0.000144 = -0.02936582$$

The lower limit of the interval is

$$-0.029512 - 0.000144 = -0.0296542$$

The error difference is -0.029512. It means that the error of Naïve Bayes classifier is less than MaxEnt classifier. The upper and lower limit of the interval has small range, approximately 0.000004. It shows that with 90% of confidence, we can conclude that Naïve Bayes classifier is better than MaxEnt classifier but the accuracy of both classifiers is not significant different.

After we got the best classifier, we conduct the post processing to extract the word or phrase belongs to method, problem, and domain categories on research title dataset. The post processing includes classification each token in every title sentence and token chunking. This is the example of post processing result:

Title sentence: large scale learning of relation extraction rules with distant supervision from the web

After classification: large p scale p learning p of p relation p extraction p rules p with none distant m supervision m from none the none web d

Chunking result:

Method class: distant supervision

Problem class: large scale learning of relation extraction rules

Domain class: web

To enrich analysis and answer the research problem, we examined the Naïve Bayes classifiers constructed from two groups of the feature set. We deliver the chunking results from four titles:

Table 5. The post processing results of naïve bayes classifier constructed from the first group of feature set

| Title Sentence | Predicted class | Actual class |
|---|---|---|
| simple algorithms for complex relation extraction with applications to biomedical ie | ['m', 'm', 'None', 'p', 'p', 'p', 'None', 'None', 'None', 'd', 'd'] | ['m', 'm', 'None', 'p', 'p', 'p', 'None', 'None', 'None', 'd', 'd'] |
| a seed-driven bottom-up machine learning framework for extracting relations of various complexity | ['m', 'm', 'd', 'm', 'm', 'm', 'None', 'p', 'p', 'p', 'p'] | ['m', 'm', 'm', 'm', 'm', 'm', 'None', 'p', 'p', 'p', 'p'] |
| a machine learning approach for efficient traffic classification | ['None', 'm', 'm', 'm', 'm', 'None', 'p', 'p', 'p'] | ['None', 'm', 'm', 'm', 'm', 'm', 'None', 'p', 'p', 'p'] |
| ddos attack detection at local area networks using information theoretical metrics | ['p', 'p', 'p', 'p', 'p', 'p', 'd', 'None', 'm', 'm', 'd'] | ['p', 'p', 'p', 'p', 'p', 'p', 'p', 'p', 'p', 'm', 'm'] |

Tables 5 shows that there is no wrong prediction on the 1st and the 3rd sentences. But on the 2nd and the 4th sentences, the Naïve Bayes classifier tends to misclassify the domain class.

Table 6. The post processing results of naïve bayes classifier constructed from the second group of feature set

| Title Sentence | Predicted class | Actual class |
|---|--|---|
| simple algorithms for complex relation extraction with applications to biomedical ie | ['d', 'd', 'None', 'd', 'p', 'p', 'None', 'd', 'None', 'd', 'd'] | ['m', 'm', 'None', 'p', 'p', 'p', 'None', 'None', 'None', 'd', 'd'] |
| a seed-driven bottom-up machine learning framework for extracting relations of various complexity | ['None', 'd', 'd', 'd', 'm', 'm', 'None', 'd', 'p', 'p', 'd', 'd'] | ['m', 'm', 'm', 'm', 'm', 'm', 'None', 'p', 'p', 'p', 'p', 'None'] |
| a machine learning approach for efficient traffic classification | ['None', 'd', 'm', 'm', 'm', 'm', 'None', 'd', 'd', 'p'] | ['None', 'm', 'm', 'm', 'm', 'm', 'None', 'p', 'p', 'p'] |
| ddos attack detection at local area networks using information theoretical metrics | ['d', 'd', 'p', 'd', 'd', 'd', 'None', 'p', 'd', 'd'] | ['p', 'p', 'p', 'p', 'p', 'p', 'None', 'm', 'm', 'm'] |

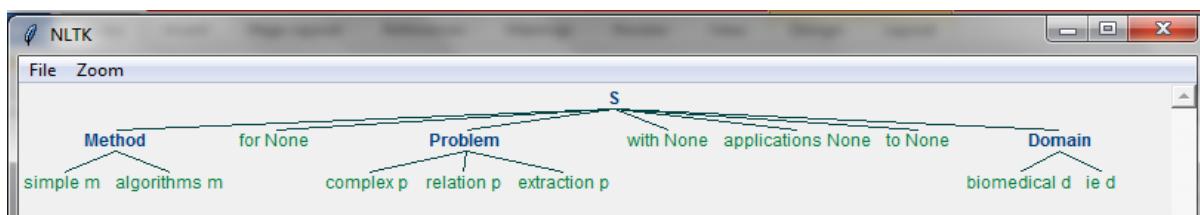


Figure 6. The chunking result of the first title using Naïve Bayes classifier learned from the first group of feature set



Figure 7. The chunking result of the second title using Naïve Bayes classifier learned from the first group of feature set



Figure 8. The chunking result of the third title using Naïve Bayes classifier learned from the first group of feature set

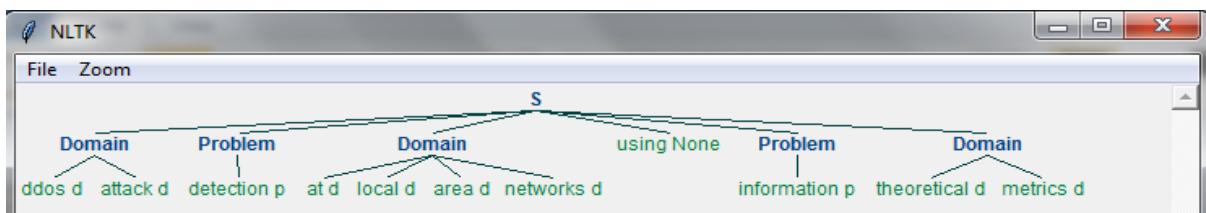


Figure 9. The chunking result of the fourth title using Naïve Bayes classifier learned from the first group of feature set

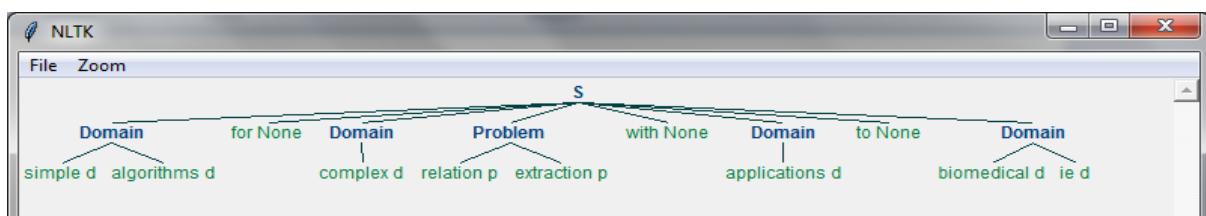


Figure 10. The chunking result of the first title using Naïve Bayes classifier learned from the second group of feature set

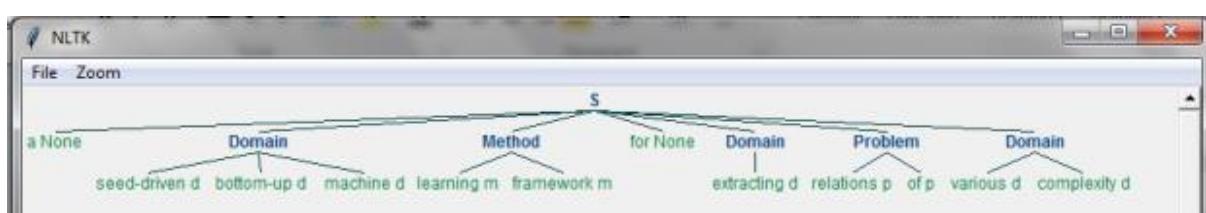


Figure 11. The chunking result of the second title using Naïve Bayes classifier learned from the second group of feature set

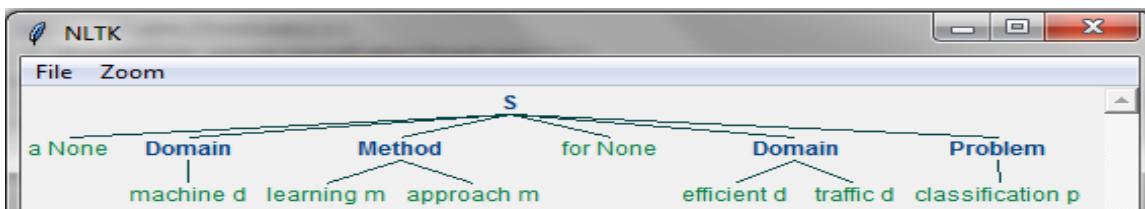


Figure 12. The chunking result of the third title using Naïve Bayes classifier learned from the second group of feature set

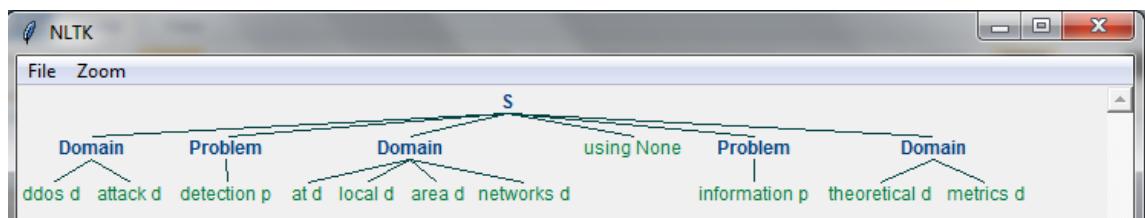


Figure 13. The chunking result of the fourth title using Naïve Bayes classifier learned from the second group of feature set

Table 6 tells us that Naïve Bayes classifier learned from the second group of feature set also tends to misclassify the domain class. The domain class is mostly classified as the method class. This classifier is not appropriate to predict class of a word instead of a phrase. If we examined to classified a phrase such as ‘biomedical ie’ or ‘complex relation extraction’ then this classifier will predict ‘biomedical ie’ as domain class and ‘complex relation extraction’ as problem class.

RESULTS AND CONCLUSIONS

There are some aspects we learn from the experimental study. The first, the labeling process should be consistent since the inconsistent label for tokens can influence the modeling process and might worse the model itself. The annotated dataset has to be validated before it is used for modeling to check the consistency of labels and the completeness of labeled tokens. Shuffle on training set produces more accurate classifier than without shuffle because shuffle lets each category/class has equal data distribution on the dataset. Therefore, each class has its representatives on both the training and testing set.

On the small size dataset, the 10-fold cross validation is an appropriate method to construct and validate/test the models instead of holdout method. The Naïve Bayes

classifier learned from the first group of the feature set is successful in predicting category of each token in title dataset. The accuracy and f1-score for each class are more than 0.80 since the first group of feature set considers the contextual and syntactic feature of a token. This classifier determines the location of a token within a sentence, considers the token and POS tag of some tokens before and after and deliberates the rules of a token. While the Naïve Bayes classifier learned from the second group of the feature set is more appropriate classifying a phrase token than a word token. This classifier just considering the tokens owned by a phrase instead determines the characteristics of word token. The definition of the token in our experimental study is a word.

We believe that it is a good idea to try the same information extraction techniques we have built on the large title dataset from various research fields. We also encourage to conduct semi-supervised learning in classifier modeling because the cost for annotation is expensive. The idea is utilizing the limited annotated titles to construct a classifier then applying the ensemble methods to improve the performance of the classifier.

REFERENCES

- Ayan, Necip Fazil, and Bonnie J. Dorr. 2006. A Maximum Entropy Approach to Combining Word Alignments.

- Proceedings of the Human Language Technology Conference of the NAACL, Main Conference* (June): 96–103.
- Bodenreider, Olivier, and Pierre Zweigenbaum. 2000. Identifying Proper Names in Parallel Medical Terminologies. *Studies in Health Technology and Informatics* 77: 443–47.
- Chodey, Krishna Prasad, and Gongzhu Hu. 2016. Clinical Text Analysis Using Machine Learning Methods. *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*.
- Dimililer, Nazife, Ekrem Varoğlu, and Hakan Altınçay. 2009. Classifier Subset Selection for Biomedical Named Entity Recognition. *Applied Intelligence* 31(3): 267–82.
- Ek, Tobias, Camilla Kirkegaard, Håkan Jonsson, and Pierre Nugues. 2011. Named Entity Recognition for Short Text Messages. *Procedia - Social and Behavioral Sciences* 27(Pacling): 178–87.
- Joachims, Thorsten. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *The 10th European Conference on Machine Learning*, , 137–42.
- Mao, Xinnian et al. 2007. Using Non-Local Features to Improve Named Entity Recognition Recall. In *Proceedings of the 21st Pasific Asia Conference on Language, Information, and Computation*, 303–10. http://dspace.wul.waseda.ac.jp/dspace/bitstream/2065/29132/1/PACLIC_21_00_031_Mao.pdf.
- McKenzie, Amber. 2013. Focused Training Sets to Reduce Noise in NER Feature Models. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, , 411–15. <http://www.aclweb.org/anthology/N13-1042>.
- Nadeau, D. 2007. A Survey of Named Entity Recognition and Classification. *Linguisticae Investigationes* (30): 3–26. <http://nlp.cs.nyu.edu/sekine/papers/li07.pdf>.
- Qin, Ying, Taozheng Zhang, and Xiaojie Wang. 2008. Chinese Named Entity Recognition with New Contextual Features. *2008 International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2008*: 1–6.
- Rafi, Muhammad, Sundus Hassan, and Mohammad Shahid Shaikh. 2012. Content-Based Text Categorization Using Wikitology. *International Journal of Computer Science Issues* 9(4): 9. <http://arxiv.org/abs/1208.3623>.
- S, Amarappa, and Sathyaranayana S.V. 2015. Kannada Named Entity Recognition and Classification (NERC) Based on Multinomial Naïve Bayes (MNB) Classifier. *International Journal on Natural Language Computing* 4(4): 39–52. <http://www.airccse.org/journal/ijnlc/paper/s/4415ijnlc04.pdf>.
- Saha, Sujan Kumar, Sudeshna Sarkar, and Pabitra Mitra. 2009. Feature Selection Techniques for Maximum Entropy Based Biomedical Named Entity Recognition. *Journal of Biomedical Informatics* 42(5): 905–11. <http://dx.doi.org/10.1016/j.jbi.2008.12.012>.
- Sebastiani, Fabrizio. 2001. Machine Learning in Automated Text Categorization. *Journal ACM Computing Surveys (CSUR)* 34(1): 1–47. <http://arxiv.org/abs/cs/0110053>.
- Suakkaphong, Nichalin, Zhu Zhang, and Hsinchun Chen. 2009. Disease Named Entity Recognition Using Semisupervised Learning and Conditional Random Fields. *Journal of The American Society for Information Science and Technology* 3(2): 80–90.
- Wu, Tianhao, William M Pottenger, and Computer Science. 2005. A Semi-Supervised Active Learning Algorithm for Information Extraction from Textual Data. *Journal of the American Society for Information Science and Technology* 56(3): 258–71. <http://doi.wiley.com/10.1002/asi.20119>.