INFORMATION EXTRACTION USING SVM UNEVEN MARGIN FOR MULTI-LANGUAGE DOCUMENT

Dwi Hendratmo Widyantoro*, Ayu Purwarianti*, Paramita*
* School of Electrical Engineering and Informatics, Institut Teknologi Bandung
Ganesha 10, Bandung, Indonesia
email: dwi@if.itb.ac.id, ayu@informatika.org

ABSTRACT

We applied SVM Uneven Margin (SVMUM) for cross lingual Information Extraction (IE) task. SVM has been proved to have good performance in IE task for English documents. So far, no IE research has been conducted for multi-language documents where the majority text are written in language with low resource tool such as Indonesian. In the experiment, we compared several training data composition to see its impact on the token classification of IE task for the multi-language document. The result shows that even though the linguistics information for Indonesian are not satisfying, the IE performance can still be improved by adding the training data. We also compared the SVMUM with kNN and NB as the learning algorithm in the token classification of IE task. The experimental results showed that SVMUM is suitable for multi-language documents. The IE accuracy achieved better performance than the other two algorithms.

Keywords: Information Extraction, SVM with Uneven Margin, multi-language documents.

1 INTRODUCTION

Information Extraction (IE) aims at extracting snippets of information automatically. The process takes unstructured text as input and produces fixed-format, unambiguous data as output. IE task mainly is supported by MUC (Message Understanding Conference) and ACE (Automatic Content Extraction, successor of MUC). Both events supply data and evaluation for the IE task. All data are provided in English language.

Extraction rules are key ingredients in IE task. These rules can be manually hand-crafted by human experts or automatically generated using machine learning approach. Hand-crafted rules may have high extraction accuracy, but are not scalable and are also very expensive to create. For this reason, researchers tend to use machine learning algorithm in constructing an IE system, where the system automatically learns extraction rules from training data (i.e., texts with annotation on piece of information to be extracted).

There are two categories in employing this machine learning algorithm: rule learning and statistical learning. Rule learning method induces a set extraction rules from training examples. IE rule based learning systems such as SRV[8], RAPIER[1], WHISK[15], BWI[10], and (LP)2 [5] belong to this category. The second category, statistical learning, aims to classify token into Named Entity (NE) category by using statistical model such as HMM[9], Maximum Entropy[4], SVM[7,11,12,13] and Perceptron[2]. These systems employed different machine learning techniques as the classifier and also utilized different features such as linguistic features or document structure features.

SVM as a general supervised machine learning algorithm has achieved a good performance on many classification tasks including Named Entity (NE) recognition. Isozaki and Kazawa[11] compared three methods for NE recognition (SVM with quadratic kernel, maximum entropy method and a rule based learning system). Their experiments showed that SVM-based system achieved better performance than other systems. Mayfield et al.[13] employed SVM with cubic kernel to compute transition probabilities in a lattice-based approach for NE recognition. Their results were comparable to others on the CoNLL2003 shared task. ELIE[7] used two-level boundary classification with SVM algorithm. The first level classifies each word into start or end tags. This first level gives high precision score but low recall score. The second classifier tries to handle the problem by detecting the end of a fragment given a start tag and the start of a fragment given an end tag. GATE-SVM[12] used SVM uneven margin to classify each word into start or end tag in one time classification. They used a post-processing to guarantee tag consistency and improve the results by exploring other information. To guarantee the consistency, the document is scanned from left to right to remove start tags without matching end tags and end tags without matching start tags. They also filtered candidates by its length, if the length is not
equal to the length of any entity of the same type in the training set then it is removed from the candidate tags list. In the last process, they put together all possible tags for a sequence of tokens and choose the best one according to the probability computed from the output of the classifier via a Sigmoid function. Based on the experimental results[7,12], ELIE achieved better performance of 78.8% average score than GATE-SVM with 77.9% average score.

All systems mentioned above employ English documents as the training and testing data. Using English documents, the linguistic features can be easily gained from the available language tools such as GATE, Collins parsers, etc. Meanwhile, there also demands of information extraction system for other languages. For major language such as Japan, the linguistic features can also be gained from the available language processing tools such as Chasen, Mecab, EDR dictionary, etc. However, for language such as Indonesian, there is no available language processing tool.

This paper explores the use of the available language processing tools to perform IE on document written in multi-languages (English and Indonesian). We found that these kind of documents are widely available on the Internet. In particular, we address the IE task on multi-language documents as the classification problem, using the SVMUM (Support Vector Machine with Uneven Margin) for learning the classification model.

2 INFORMATION EXTRACTION AS TOKEN CLASSIFICATION

There are three main components in modeling IE as token classification problem[14]: context representation, classification algorithm, and tagging strategy. Each component will be described in the following sections.

2.1 Context Representation

Before token classification is performed, each token in a document will be preprocessed into several features. In this work, we used ANNIE-GATE[6] as the natural language processing tool. Even though ANNIE was designed for English, it can still be used to obtain linguistic features in our multi-language documents because English and Indonesian have similar characteristics.

Specifically, there are four linguistic features for each token:

1. Lexical form. The English words are processed by using English morphological analyzer tool. For example, “are” is transformed into “be”. The Indonesian words are assumed as the OOV (out of vocabulary) words, the lexical form is not changed.
2. Orthographic information. Indonesian and English have the same rules on the orthographic style. Name words (such as person, location, organization) are usually headed with an uppercase letter. This feature has values of upperInitial, allCaps, and standard. For example the orthographical feature for the word “PM” is “allCaps”.
3. Semantic gazetteer information. A word has this feature if the word exists in the gazetteer of ANNIIE. For example the gazetteer feature for the word “PM” is “Time”. Indonesian words have poor information feature since the ANNIE’s gazetteer has only English words in it.
4. ANNIIE’s predicted named entity (NE). Other than gazetteer, ANNIE also has an NE predictor. The NE resulted by this predictor is used to help the token classification in our multi-language IE system.

Using only the feature vector of one current token is not sufficient to perform token classification because of the context effect from surrounding token. Therefore, to classify a particular token, the feature vector needed for the classification includes linguistics feature of the current token, three previous tokens and three successor tokens.

2.2 SVMUM as Classification Algorithm

SVM has been proved to have a good performance on classification task. To handle the imbalence training data, Li[12] has proposed SVM with Uneven Margin to handle document categorization and token classification in IE. Their experimental results showed that SVM with Uneven Margin gave better performance on the IE task compared to other modification of SVM method.

Given a training set $Z = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $x$ is the $n$-dimensional vector input and $y$ is the label (-1, +1), the SVM with uneven margin is obtained by solving the quadratic optimization problem:

$$
\min_{w, b, \xi} \langle w, x \rangle + C \sum_{i=1}^{m} \xi_i \quad (1)
$$

s.t. $\langle w, x_i \rangle + \xi_i + b \leq 1$ if $y_i = +1$

$\langle w, x_i \rangle - \xi_i + b \leq -\tau$ if $y_i = -1$

$\xi_i \geq 0$ for $i = 1, \ldots, m$

$\tau$ is the ratio of negative margin to the positive margin of the classifier and is equal to 1 in the standard SVM. In this paper, we used the Java version of LibSVM[3].
2.3 Tagging Strategy

Each token is classified into a start tag or an end tag of each class of information to be extracted. There are 14 classes in our job domain problem: industry (company domain), company_name, job_category, job_title, location education_level (minimum desired education degree), foreign_language, description (job description), salary, contact (email address), deadline, posting_date, needed_experience, and experience_duration.

After each token is classified into start tag, end tag or none of these, tag consistency checking is performed. Token with start tag but without an end tag for a particular class will not be considered as the class candidate. If for one text snippet, there are more than one class candidate, the selected class candidate is the one with the highest probability.

3 EXPERIMENTS

3.1 Experimental Data and Setup

To test the IE system on multi-language documents, we collected multi-language job advertisement from the Internet. We also collected English only job ad. There are 90 English documents and 90 Indonesian-English documents downloaded from several company web sites. There are three training data sets used in the experiments: (1) 80 English – 10 Indonesian-English documents, (2) 45 English – 45 Indonesian-English documents, and (3) 10 English – 80 Indonesian-English documents. Each training data is tested twice: by using 10 English documents, and by using 10 Indonesian-English documents.

3.2 Experimental Results

The overall IE performance on the three types of training data is shown in Figure 1. The score shown in the figure is the average of lenient and strict F-measure. The F-measure score is calculated using equation 2.

\[
F - measure = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]  

Figure 1 shows that although there is some missing information on the feature vector for the Indonesian-English documents, the IE performance can still be improved by adding the size of training data. The performance on Indonesian-English documents on Data-3 (80 Indonesian-English) improved by 14% over that of on Data-1 (10 Indonesian-English). The additional training data enhances the token-class vocabulary.

The performance of English documents is 20% higher that that of the Indonesian-English documents on Data-1 (80 English – 10 English-Indonesian). When the composition is reversed such as on Data-3 (10 English – 80 English-Indonesian), the performance improvement of the English-Indonesian documents is only 3.7% higher than that of the English documents. It points out that although the number of training data affects the IE accuracy, linguistics features also plays an important role in improving the IE performance.

<table>
<thead>
<tr>
<th>Class</th>
<th>Data-1</th>
<th>Data-2</th>
<th>Data-3</th>
<th>Data-1</th>
<th>Data-2</th>
<th>Data-3</th>
<th>Data-1</th>
<th>Data-2</th>
<th>Data-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>industry</td>
<td>1.00</td>
<td>1.00</td>
<td>0.91</td>
<td>0.59</td>
<td>0.59</td>
<td>0.67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>company_name</td>
<td>0.93</td>
<td>0.90</td>
<td>0.90</td>
<td>0.67</td>
<td>0.80</td>
<td>0.77</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>job_category</td>
<td>0.64</td>
<td>0.50</td>
<td>0.38</td>
<td>0.25</td>
<td>0.40</td>
<td>0.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>job_title</td>
<td>0.57</td>
<td>0.50</td>
<td>0.35</td>
<td>0.48</td>
<td>0.69</td>
<td>0.67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>location</td>
<td>0.90</td>
<td>0.90</td>
<td>0.63</td>
<td>0.61</td>
<td>0.58</td>
<td>0.67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>education_level</td>
<td>0.91</td>
<td>0.83</td>
<td>0.87</td>
<td>0.90</td>
<td>0.95</td>
<td>0.93</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>foreign_language</td>
<td>1.00</td>
<td>0.92</td>
<td>0.92</td>
<td>0.72</td>
<td>0.81</td>
<td>0.76</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>0.19</td>
<td>0.21</td>
<td>0.43</td>
<td>0.22</td>
<td>0.35</td>
<td>0.55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>salary</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contact</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
<td>0.94</td>
<td>0.86</td>
<td>0.81</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The detail experimental result for each class is shown in Table 1. The larger the size of the training data, the higher the performance it could achieve, except the “description” class in the English testing data set. Looking into the English testing data, we found that the “description” class contains a large number of variety of text snippet. Unlike other classes such as “posting_date”, “salary”, or “company_name”, the “description” class could not be classified by its orthographical nor token type information. The lexical forms or the semantic information of the start words and end words also vary. Further analysis is needed in order to find representative features of a text snippet to have a good classification on the “description” class.

In the experiments, we also compared the performance of SVM algorithm with kNN and Naïve Bayes algorithms. The IE accuracy is shown in Figure 2. The figure shows SVM algorithm achieved higher IE performance compared to kNN and Naïve Bayes algorithms, more over when using Indonesian-English documents as the testing data set. Using English documents as the testing data set, the performance of SVM and kNN are quite comparable. It is different from the result comparison on the Indonesian-English documents, the SVM algorithms outperforms kNN about 22.6% accuracy. It assures that SVM with Uneven Margin is suitable for imbalanced data such as the one in multi-language documents. In the experiments, SVM with Uneven Margin outperformed two other machine learning algorithms: kNN and Naïve Bayes.

It seems that token classification with the start and end tagging strategy does not work well for variant-long text snippets such as “description” class. This kind of more general information needs more global information that the one contained in the three window size that we used in our research. Our future work will further explore this problem.

4 CONCLUSION

Our work shows that although the linguistics features of multi-language documents are not as adequate as the English documents, the SVMUM still gives good IE performance on these multi-language documents. Furthermore, the performance can be improved by providing more training data.

The experimental results also assured that SVM with Uneven Margin is suitable for imbalanced data such as the one in multi-language documents. In the experiments, SVM with Uneven Margin outperformed two other machine learning algorithms: kNN and Naïve Bayes.

ACKNOWLEDGEMENT

This research is supported by Insentif Program from the Ministry of Research & Technology under contract# 53/RT/Insentif/PPK/II/2008 and by Research Grant from Institute of Technology Bandung under contract # 0045/K01.03.2/PL2.1.5/I/2008.

REFERENCE


