Towards a methodology for entity error analysis in annotated corpora

Qi Wei
National Institute of Informatics
2-1-2, Chiyoda-ku
Tokyo 101-8430, Japan
qiwei@nii.ac.jp

Yuval Krymolowski
Department of Computer Science
University of Haifa
Haifa 31905, Israel
yuval@cl.haifa.ac.il

Nigel Collier
National Institute of Informatics
2-1-2, Chiyoda-ku
Tokyo 101-8430, Japan
collier@nii.ac.jp

ABSTRACT

We present a methodology for error analysis in entity annotation. To increase the accuracy in corpora, there is a need for an analysis method for detecting human annotation and schema errors. We use easiness statistics and information gain to gain insights into possible causes of error in the GENIA corpus of MEDLINE abstracts.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligent**]: Natural language processing

General Terms

error analyse algorithms

1. INTRODUCTION

With the rapid expansion of biomedical research, an overwhelming number of research publications are being produced which require searching. In order to help with this task, text mining has been applied in areas ranging from the extraction of signal transduction pathways to the analysis of infectious disease outbreaks. Within text mining, named entity recognition (NER), which seeks to identify and classify terms into predefined target classes, is regarded as the first key stage in mapping to a computable semantic representation.

NER originated from the Message Understanding Conferences (MUC) in 1990s. The task in MUC is to identify terms such as person name, organization name, etc., in the Newswire domain. During the last few years, NER in the biological domain has improved rapidly. The task in biological named entity recognition (BioNER) is to identify and label DNA and other products. The accuracy for BioNER (about 70%) is much lower the average 90% accuracy for the MUC task.Compared with the Newswire domain, the entities in the biomedical domain tend to be more complex due to factors such as long and descriptive naming conventions and conjunctive and disjunctive structures.

In most of the current error analyses[3, 5], one selects a fixed number of errors and classifies them manually. In such cases, there is a critical need for analysis tools and methods for detecting human annotation errors and schema inconsistencies.

In this paper, we present a general method for error analysis on annotated corpora. By applying this method, we can access every error in our testing data and get more detailed information on the errors.

2. METHOD

After obtaining the test results from 400 models, we applied easiness and hardness statistics[4] to each instance. Then we constructed a confusion matrix from the hard instances. In addition, we used the information gain derived from the easiness and hardness statistic to calculate the contribution of each feature used in the NER system.

2.1 Easiness and hardness statistics

Easiness and hardness statistics were first created by Krymolowski [4]. Consider a collection of models with similar recalls and precisions; correctly classified words may be different. If a word can be classified by all models, it is treated as easy and if it is classified wrongly by all models, it is treated as hard. The definition of easiness and hardness comes from this idea. Let L denote a set of supervised learning models and T the set of test data. Each instance $t \in \mathtt{T}$ can be characterized by a bit-vector:

$$v(t) = \{v_1(t), \cdots, v_n(t)\},$$

where

$$v_i(t) = \begin{cases} 1 & \text{t was labeled correctly by model I,} \\ 0 & \text{t was labeled wrongly by model I} \end{cases}$$

Easiness is defined according to the vector v(t):

$$easiness(t) = \frac{\sum_1^n v_n(t)}{n}$$

which is the probability of correctly labeling t by one of the classification models. The value of easiness(t) is between 0 and 1. Here, we define that an instance whose easiness

is between 0 and 0.1 is called hard and an instance whose easiness is between 0.9 and 1 is called easy.

Hard and easy instances can be further divided. We focus on hard instances that most models can not recognize correctly.

2.2 Information Gain

Information gain[1] is used to calculate the contribution of each feature used in the NER system.The entropy for NE classes H(C) is defined by

$$H(C) = -\sum_{c \in C} p(c) log_2 p(c)$$

where $p(c) = \frac{n(c)}{N}$,n(c) stands for the number of words in class c and N stands for the total number of words in data pool

When a feature F is given, the conditional entropy for NE classes H(C|F) is defined by

$$H(C|F) = -\sum_{c \in C} \sum_{f \in F} p(c, f) log_2 p(c|f)$$

where $p(c, f) = \frac{n(c,f)}{N}$; $p(c|f) = \frac{n(c,f)}{n(f)}$; n(c,f) stands for the number of words in class c with the feature value f and n(f) stands for the number of words with the feature value f

The information gain for NE classes and a feature I(C;F) can be calculated as:

$$I(C; F) = H(C) - H(C|F)$$

The information gain shows how the feature F contributed to the classification. I(C;F) equals 0 if feature F is completely independent of C and equals 1 if F gives sufficient information to label named entities.

To deal with different features, the information gain has to be normalized as the information ratio:

$$GR(C; F) = \frac{I(C; F)}{H(C)}$$

GR(C; F) ratios are close to 1 and 0 and can be compared even if the class entropies are different.

3. EXPERIMENT

3.1 Data set and models

GENIA corpus version 3.02 was used in this experiment. 36 classes were used to annotate the corpus.SVM[2] was selected as the supervised model in the test and 400 different models were used. 40% of the corpus taken from the beginning was used for testing. 24% of the corpus (randomly sampled) was used to train the 400 different models. No cascaded entities existed in this experiment; only the longest entity was annotated.

3.2 Results

Using the method described above, errors were successfully classified into three types: 1. Boundary errors and no classification errors; 2. Boundary errors with classification errors; 3. Only classification errors with no boundary errors.

Most of the errors were caused by inconsistent annotations.For example,

1. .. in normal T cells in which IL-2R alpha expression has been induced.
2. .. are activated in normal T cells in response to IL-2.

In the first sentence, "T cells" without "normal" was annotated as a cell type, while in the second sentence, "normal T cells" was annotated as a cell type in the original corpus.

In the result, a kind of errors were found which we called incomplete forms,For example,

1. <proteinmolecule> **protein kinase C-alpha** , - **epsilon** , and - zeta <pro-teinmolecule>
2. < proteinmolecule > **LMP1** and 2 < proteinmolecule >

Forms like '-epsilon', '-zeta' are in-complete, and they need to be recovered to their full terms of 'C-epsilon' and 'C-zeta'.

4. CONCLUSIONS

Corpus error analysis is an important step in improving the accuracy of bioNER. The easiness and hardness statistics used here are effective in measuring the degree of hardness that a model has in recognizing one entity. We focused on the hard entities and this made it easy to get all errors in the experiment results. Also, this allowed us to select error categories for drill down analysis. The importance of a feature can be learned by using the information gain, and from the import features, evidence can be found to strengthen the results. We used these two methods together and it helped us to find inconsistent annotations in the GENIA corpus.

5. REFERENCES

[1] L. Breiman, R. Friedman, A. Olshen, and C.Stone. Classification and regression tree. In *Belmont CA: Wadsworth International Group*, 1984.

[2] N. Cristianini and J. Shawe-Taylor. An introduction to support vector machines: and other kernel based learning methods. In *Cambridge University Press, New York, NY*, 2000.

[3] S. Dingare, M. Nission, J.Finkel, C. Manning, and C. Grover. A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and evaluations comparative and func-tional genomics. 2005.

[4] Y. Krymolowski. Distinguishing easy and hard instances international. In *Conference On Computational Linguistics*, 2002.

[5] G. Zhou. Recognizing names in biomedical texts using hidden markov model and svm plus sigmoid. In *International Joint workshop on Natural language Processing in Biomedicine and its Applica-tions(JNLPBA) 2004*, 2004.