Bacteria Biotope - Event extraction of microorganisms and habitats *

Animesh Karmakar
Roll No.: 120101010
a.karmakar@iitg.ernet.in

Bharat Narayan Gupta
Roll No.: 120101017
g.bharat@iitg.ernet.in

Abhijeet Singh
Roll No.: 120101083
abhijeet.singh@iitg.ernet.in

Abhishek Sen
Roll No.: 120101087
abhishek.sen@iitg.ernet.in

ABSTRACT

Bacteria Biotope is a part of BioNLP Shared Task 2016, which aims to promote information extraction on microorganism biodiversity and assess the performance of automatic categorization and relation extraction systems. This task is useful in studying the interaction mechanisms of the bacteria with their environment from genetic, phylogenetic and ecology perspectives. The habitat information corresponding to a bacteria is greatly used in applied microbiology such as food processing and safety, health sciences and waste processing. In this report we describe our approach to the problem and the progress made so far. The problem is divided into 3 parts: entity recognition, entity categorization and event detection. We present results for the three tasks in Bacteria Biotope(BB3).

1. INTRODUCTION

Bacteria Biotope 3 is a task in BioNLP Shared Task 2016. The aim is to build a Information Extraction System that has the capability to:

- detect mentions of habitats and species;
- categorize them with large ontologies;
- extract events between bacteria and their habitats.

The task is aimed to promote information extraction on microorganism biodiversity and assess the performance of automatic categorization and relation extraction systems.

Bacteria Biotope is a critical information for studying the interaction mechanisms of the bacteria with their environment from genetic, phylogenetic and ecology perspectives. The information on habitats where bacteria live is a particularly critical in applied microbiology such as food processing and safety, health sciences and waste processing.

The task was broadly divided into 3 categories namely:

- Entity Recognition
  The Bacteria Biotope problem includes three types of entities namely bacteria, habitats and geographical places for named entity recognition.The bacteria entities are annotated as contiguous spans of text that contains a full unambiguous prokaryote taxon name. The category that the text entities have to be assigned to is the most specific and unique category of the NCBI taxonomy resource. Habitat entities are annotated as spans of text that contains a complete mention of a potential habitat for bacteria.

- Entity Categorization
  After the entities are recognized the next task is to categorize them. Habitat entities are assigned one or several concepts from the habitat subpart of the OntoBiotope ontology. Geographical entities are geographical and organization places denoted by official names.

- Event extraction
  There is only one type of event i.e. Lives_In. The Lives_In event takes bacteria and the place where it lives (which can be habitat as well as a geographic location) as the two arguments. The event does not include trigger words.

2. HABITAT AND BACTERIA ENTITY RECOGNITION

We have used linear chain Conditional Random Fields (CRF) for entity recognition. CRF is a class of statistical modeling method, used for structured prediction in machine learning, pattern recognition (like object and image recognition) etc. The CRF is different from an ordinary classifier because it takes into consideration the neighboring samples rather than just focusing on a single sample, one such type is the Linear Chain CRF which we have used.
The linear chain CRF model[2] consists of set of all possible states, set of all possible state sequences, a model which gives the probability of a sequence of states given an input sequence. Linear chain CRF consists of a feature vector which maps an entire input sequence, paired with an entire state sequence to some d-dimensional feature vector. With the help of feature vectors, input sequence and states, we define a giant log-linear model. The liner CRF further consists of decoding i.e. given an input sequence,we would like to find the most likely underlying state sequence under the model. Further, we have parameter estimation where we consider a set of n labeled examples i.e. collection of input sequence and corresponding state sequence and then proceed similarly to regular log-linear models.

*This template is adapted from http://www.acm.org/publications/article-templates/SIG%20Proceedings%20Template-May2015%20Zip.zip

We used the linear chain CRF python wrapper of CRFsuite [5] [7] for named entity recognition.

3. FEATURES SET USED

We initially tokenized the data given to us and then passed it to python wrapper CRFsuite for named entity recognition. The features used for labeling are:

- Basic Features
  These features were based only on the word in consideration, not considering any other word in the data. The features considered:
  - Suffix of size 2,3 and 4 characters.
  - Prefix of size 2,3 and 4 characters.
  - Whether the given word is a title.
  - Whether the word is alphanumeric.
  - Whether the initial letter is in upper case, one character is in upper case or complete word is in upper case, etc.
  - Whether the initial letter is in lower case, one character is in lower case or complete word is in lower case, etc.
  - Whether the word is lemmatized.
  - POS tags.
  - Punctuation character.
- Contextual Features
  These features considered the context of the word along with the word. The context ranges to multiple neighboring word. We mainly focused on the bigrams and thus considered 3 cases: two words before, one word before and one after and both words after the word in consideration. The features were based upon applying the basic features on the words along with the contextual neighboring words. There were thus lemmatized words, normal words and the POS of words.

Results and top likely and unlikely transitions are shown in the tables.

Table 1: Results using linear chain CRF

	Precision	Recall	F1-score	Support
Bacteria_B	0.92	0.73	0.81	237
Bacteria_I	0.87	0.74	0.80	334
Habitat_B	0.77	0.33	0.46	363
Habitat_I	0.62	0.36	0.46	407
Others (O)	0.93	0.98	0.96	8025

Table 2: Exact Boundary Detection Results

Tag	Precision	Recall	F1
Bacteria	76.04%	61.60%	68.07
Habitat	53.42%	23.69%	32.82
Overall	65.72%	38.67%	48.69

4. ENTITY CATEGORIZATION TASK

After identifying bacteria and habitat mentions in the extract the next task is the categorization. OntoBiotope habitat ontology file assigns each habitat a unique identification number. We have to provide each habitat in the document its closest identification number. Similarly each bacteria mention is to be categorized into predefined bacteria Taxonomy ID defined in NCBI Taxonomy file. For Example 1 gives an example of entity categorization. Bacteria name "Serratia" is given the reference number 613 and habitat name "Hospital S. Camillo De Lellis" is categorized to OBT:001835.

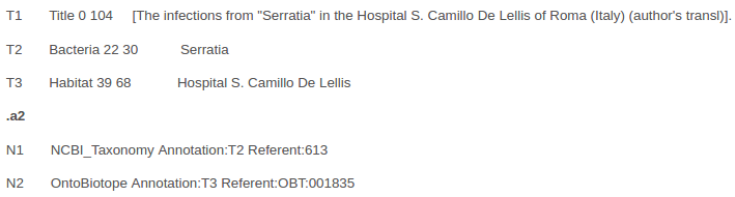

Figure 1: Entity Categorization

The number of ontology categories is approximately 4000 and the number of taxonomy categories is $2.1x10^6$ which makes this problem even more challenging. Due to this huge number of categories and very little training data it is not possible to use any supervised learning algorithm.

We started with creating four dictionaries. The first one maps each habitat name provided in the OntoBiotope file to its referent ID. The second one maps each bacteria name provided in the NCBI taxnomy file to its unique identifier. We applied following normalizations to the keys of the dictionary:

- lowercase
- removing stop-words
- stemming using SnowballStemmer

The Third and fourth dictionaries is the mapping of habitat and bacteria mentions in the training data to their respective IDs. Now we have modeled the categorization problem as a **K Nearest Neighbor** problem. We categorize the entity mentions in the test data in two steps: First we calculate word matching similarity score of the mention that is to be categorized with every mention in the training data (i.e. with keys of dictionary three for habitat mention and keys of dictionary four for bacteria mention). We save the key with the maximum similarity score. Second we calculate the word matching similarity score of the mention with every key in the first dictionary in case of habitat mention and with every key of second dictionary in case of bacteria mention. We again save the key with the maximum score.

Now we take the key with the maximum score among the above two keys and map the entity with its ID. The similarity score that we have used is **gestalt pattern matching**.

4.1 Results

We made the dictionaries using the training data, Ontology file and taxonomy. We tested on the development corpus. Here accuracy is defined as the percentage of categorizations that were correctly assigned their respective categories.

Table 3: Accuracy on Development Data-Set

Entity	Accuracy
Habitat Categorization	42.43%
Bacteria Categorization	62.04%

5. EVENT EXTRACTION TASK

In this task we are required to identify event **Lives_In** among bacteria and habitat / Geographical location. According to problem specifications this event does not include any trigger words. The two arguments, habitat and bacteria must belong to the same document. It is not necessary that the two arguments belong to the same sentence. Example: *"Long-term Helicobacter pylori infection and the development of atrophic gastritis and gastric cancer in Japan."* Here the entities are :

Bacteria	Helicobacter pylori
Habitat	gastric
Geographical	Japan

Here bacteria *"Helicobacter pylori"* is in Lives_In relation with Geographical entity *"Japan"* but it is not related to Habitat entity *"gastric"*.

Annotation task is very challenging because of domain specific and difficult rules used by the manual annotators. The event extraction requires understanding of the text and coming up with a model that is capable of inferring details that are not explicit. For example there are rules like "transitivity", vaccine identification,whether a sentence is a hypothesis or not,topological constraints, disease mention recognition as shown in figure. Since there is no exhaustive list for trigger words, vaccines, disease names, locations etc the problem of event detection in this corpus is even more aggravated.

M. leprae *directly bind to ErbB2 on* *myelinated Schwann cells*.

→ "myelinated Schwann cells" is a cell and the bacteria do not live inside, so the relation is not annotated.

Figure 2: Example of Entity Categorization

Marinobacter *belong to the class of* *Gammaproteobacteria* *and these motile, halophilic or halotolerent bacteria are widely distributed throughout the* *world's oceans*.

→ The relation between "Marinobacter" and "world's oceans" is annotated. However there is no relation annotated between "Gammaproteobacteria" and "world's oceans" because this relation would not be universal.

Figure 3: Example of Entity Categorization

5.1 Model Description

We framed the problem as a binary classification task. For every pair of bacteria and habitat in a document we try to classify whether they are in the Lives_In relationship or not. After analyzing the annotation specification we came up with the following features to generate the feature matrix.

This feature matrix is fed into the classification algorithms to train it. We used **SVM**, **Random Forest** and **KNN** as the classification algorithms [4].

- **Number of sentences between the occurrence of habitat and bacteria entities.** We have observed that if the habitat and bacteria mention belong to the same sentence then the chances of them being in the Lives_In relationship increases, although this is not always true as we have seen from the previous figures.
- Determining which entity comes before - bacteria or habitat. We have seen that mostly bacteria comes before the habitat mention. Since we are using all the bacteria habitat pairs in a document, this feature becomes important.
- Whether the location entity is a habitat or geographical location.
- **Number of words in between the bacteria and habitat entities**. As the number of words between the entities increases chances of them being in the relationship also decreases. This feature tries to enforce the importance of context.
- There are certain words which when appear in a context increases the chances of signifying the bacteria habitat relationship. For example we use the presence of following words as a feature for training:

isolate	disease	inside
habitat	lives	colonization
commensality	invasion	infection
abscess	found	symbiotic

- **Presence of Disease name** in the context increases the chances of the bacteria infecting the habitat there by implying that they are in the Lives_In relationship. In order to identify whether disease name is present or not in the context we used a Conditional Random Fields based trained model over the NCBI disease recognition corpus. The features that we used in training the CRF over the disease mention corpus were the same that we used for habitat and bacteria recognition task. The context was fed to the trained model which generated a Boolean output of whether the tag "Begin Disease" is present or not. This is used as as a Boolean feature for training.

Non-O1 Vibrio cholerae *bacteremia in* *patients with cirrhosis*: *5-yr experience from a single* *medical center*. *... The overall case-fatality rate was 23.8%, but 75% of the deaths were observed in* *patients with* *skin* *manifestation*.

→ The relation between "Non-O1 Vibrio cholerae" and "patients with cirrhosis" is annotated. However the relation between the bacteria and the skin is not established, even though it causes a symptom on the skin.

Figure 4: Presence of disease name

- **Word2Vec based similarity**: Word2Vec is a set of algorithms that tries to map words to multidimensional vectors. These vectors represent the word's relation to other words. The model has been formed

using two-layer neural networks based on skip-grams and Continuous Bag of Words. We used the pretrained model trained on the Google News data-set. Our use is restricted to finding whether the (bag of) words present in between the bacteria and habitat entities in the text signify some kind of Lives_In relationship or not. We find out the cosine similarity between the following two vectors - the vector of context words (between the bacteria entity and the habitat entity) and vectors like of phrases like:

lives inside isolated from causes disease infection	lives in habitat lives colonization

Good similarity score from these vector depict more chance of the Lives_In relationship. Similarly good similarity score with phrases such as *binds cell* and *dead* show less chances of bacteria being in the Lives_In relationship. We also saw that these features were giving more **correlation** with the Lives_In event.

5.2 Classification Algorithms Used

Following are the three learning algorithms that we used for binary classification:

- Random Forest - This[1] is a classification technique wherein we combine several decision trees, each of which is built over a sub-sample of the dataset, to predict the class of the object. The decision trees are combined by taking the mode of the classification results of the individual trees. Several techniques are used to improve accuracy and control over-fitting like that of bootstrap aggregating or bagging. Here, feature bagging is used which selects a random subset of the features at each candidate split. This way, it builds an internal estimate of the generalization error which is unbiased.

- K-Nearest Neighbor - This is a non-parametric type of classification where the object to be classified is assigned the class that is most common among its neighbors. This does not involve much pre-processing and is one of the most basic Machine Learning Algorithm. It can become inaccurate when the class distribution is skewed and most of the neighbors of an object are dominated by a class containing a number of objects [6].

- SVM - SVM[3] is a supervised learning approach which builds a hyperplane to separate the two classes. The data points are transformed into a separate space (using kernels) by which we can even perform non-linear classification. We used polynomial kernel as it is useful when the data-points are not linearly separable. The step size that we used for training was 3.5.

5.3 Results

Before the experiments, we saw the correlation of the various features with the Lives_In event. We saw that feature "Number of words in the context" and the Word2Vec features were showing more correlation (as high as 0.35). We even tested with removing low-correlated features, but this step didn't show any significant change in the evaluation scores. The evaluation metric we used is based on accuracy, precision, recall and F1 score of the classifiers. The performance of various algorithms have been shown in Table 4. The total number of training examples in the experiment were 4355, out of which we chose 80% for training and the rest for testing. We can see that the Random Forest performs the best in terms of F1 score.

Since the data is skewed towards Bacteria-Habitat pairs which are not in Lives_In relationship, we also tried to do clustering of the data first, and then classify on individual clusters. We divided the data into two clusters, but the results were not promising. For one cluster, the precision, recall, F1 score as - (0.59, 0.25, 0.35) and for the other cluster we got - (0.75, 0.1, 0.17).

Table 4: Exact Boundary Detection Results

Classifier	Accuracy	Precision	Recall	F1
SVM (poly kernel)	0.902	0.64	0.26	0.37
Random Forest	0.895	0.70	0.27	0.39
KNN	0.893	0.56	0.15	0.23

6. CONCLUSION

In this report we tried to solve the tasks of BioNLP 2016. We started with identifying mentions of bacteria and habitat in the corpus using Conditional Random Field. We achieved very impressive F1 score in Bacteria identification and decent F1 score in Habitat identification. After identifying the entity mentions the next task was to assign them categories. The number of categories were very huge. So we adopted a dictionary based approach and used the textual similarity of entities with the keys of the dictionary to assign them appropriate categories. After that we solved the problem of identifying Lives_in event extraction. We started with analyzing the rules for annotating the event. For every possible habitat bacteria pair in the every document we created a set of features. This feature matrix was input to our classification algorithms. The classification algorithm we used were Random Forest, KNN and SVM.

7. REFERENCES

[1] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] M. Collins. Log-linear models, memms, and crfs.

[3] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

[4] İ. Karadeniz and A. Özgür, "Detection and categorization of bacteria habitats using shallow linguistic analysis," *BMC bioinformatics*, vol. 16, no. Suppl 10, p. S5, 2015.

[5] N. Okazaki. CRFsuite a fast implementation of conditional random fields.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] J. N. Viewer. CRFsuite example to build a ner system.