Constructing a KB for Bio-chemical concepts for accelerated resource discovery

Shubham Singla, Manas Joshi, Maya Ramanath
IIT-Delhi

August 3, 2019

1 Introduction

Unstructured content makes up the bulk of organizational data and is growing fast. The existing process to search for information in literature, a combination of reading and existing search tools, suffers from gaps because of the complex nature of questions being asked within the literature survey. There is a need for computing capability that can evaluate text like a human mind and enhance discovery with subject matter expertise.

1.1 Goal -

We aimed at constructing a system that given a pre-defined set of documents

1. automatically identifies microbes (by genes and species) that have been described in the literature
2. extracts the following attributes of interest
   (a) Substrates on which microbe acts
   (b) Environment in which the microbe appears
   (c) reactions in which microbe participate
   (d) Properties of the microbes

1.2 Documents format

100 Bio-medical research papers were available in the form of portable document format which act as an unstructured content. These documents contain images along with texts written in haphazard manner. To extract useful stuff from these documents, these have to be preprocessed by ignoring the images. This was done manually and all the data was converted into text format. Any image present therein along with their description was ignored. This preprocessing of documents is necessary to extract the important information which would be useful for research purposes.

1.3 Approach

The approach taken to extract useful information from the unstructured research articles is described briefly below -

1. Preprocessing of the unstructured research articles
2. Deciding useful entities for annotation and relations to be extracted
3. Formation of entity database
4. Annotation of useful entities
5. Annotation on the basis of adjectives
6. Extraction of meaningful relations using co-occurrence
7. Refining relations using machine learning models such as neural networks, Randomforest etc.

2 Entity extraction

2.1 Types of interesting entities

Given the goal outlined in the Introduction, we are interested in the following types of entities:

- Microbes: Microbes or microorganisms are the small life forms that can not be seen by naked eye surrounding us. For example Bacteria like Cyanobacteria, Escherichia coli, Fungi like Penicillium, Rhizopus stolonifer(a bread mold) etc.

- Substrates: A substrate is a chemical substance on which a reaction or a process is carried on. For example Benzene, Ethanol, gasoline etc.

- Environment: Environment here refers to the surroundings or conditions in which a process takes place or a microbe lives in. For example groundwater, hydrocarbon-contaminated soil etc.

- Reaction/Process: It refers to a process in which a substance undergoes a physical or chemical change. For example - fermentation, oxidation, cracking(of oil).

- Enzyme: Enzymes are biological catalysts. They accelerate a chemical reaction. For example - monooxygenase, carboxylase etc.

- Nutrient: Nutrients are substances used by organisms to maintain life and growth. For example - iron, NaCl etc.

- Property: These are used to describe a microbe. For example - anaerobic, rod-shaped etc.

We consulted experts for examples for each kind of entity above and were able to obtain four different kinds of entities. There were 23 entities which represented ”property” (abiotic, contaminated, prokaryotic), 17 represented surrounding conditions or ”environment”(wastewater, beach, soil), 10 represented a ”chemical reaction” (degradation, denaturing, fertilization), 6 represented a ”substance” (diesel, oil, hydrocarbon) . We used these entities as given to annotate our document set. However, on closer examination of these annotations, we noticed that fine-grained entities were not being annotated (indeed, they were not in our expert-prepared entity set at all). For example, annotated entity was ”soil” whereas it should have been ”hydrocarbon-free soil”, ”methanogenic soil”, similarly for an annotated entity ”degradation”, it should have been ”aerobic degradation”, ”benzene degradation” and so on. In order to ensure that our relation extraction gave meaningful facts, we first decided to improve the entity extraction step of our system.

2.2 Automatic entity extraction

Our key observation was that, the entities that we annotated in our documents typically came with adjectives that further described the entities (such as ”anoxic” groundwater, ”alluvial” sand). Therefore, our idea was to *automatically* extract these adjectives and acquire fine-grained entities. We proceeded as follows.

We used Stanford CoreNLP to assign part of speech tag to each word in the corpus. Then certain rules were formed to extract desired entities from the corpus as described below(Penn Treebank P.O.S. Tags are used to describe the rules):

Rule 1

Merge all the adjectives(JJ) on the left side of entity(E) with it to form a new entity(E’).

$$\text{E'} = \text{JJ JJ} \ldots \text{E}$$

For example, for an entity sediments(E), a more relevant entity extracted was ”aromatic(JJ) hydrocarbon-contaminated(JJ) sediments(E)” (E’)

Going through a few entities it was observed a few words on the left along with a noun make a descriptor of an entity. A set(S) of such words was created. A few of them are ”substituted”, ”derived”, ”degrading” etc.

Rule 2

Merge a noun(NN) and a word from S on the left side of entity(E) with it to form a new entity(E’).

$$E' = NN\ S\ E$$

For example, "alkyl(NN) substituted(VBN) aromatic(JJ)" gives more information then the original entity "aromatic"(E).

It was observed that some adjectives didn't contribute to the meaning of the entity, for example "other", "different", "typical" etc. So a list of stopwords was created.

Using the two rules and stopword pruning, we were able to extract a total of 1406 from the provided 56 entities. A sample of 200 entities was chosen. This sample gave an accuracy of 74.5 percent.

Next following new rules motivated from Rule 2 were introduced:

Rule 3
Merge a noun(NN/NNS/NNP/NNPS) and an adjective(JJ) on the left side of entity(E) with it to form a new entity(E').

$$E' = NN/NNS/NNP/NNPS\ JJ\ E$$

For example, "microbial(NN) hydrocarbon(JJ) biodegradation(E)" gives more information then the original entity "biodegradation"(E).

Rule 4
Merge a noun(NN/NNS/NNP/NNPS) and an verb(VB/VBG/VBN/VBP/VBZ) on the left side of entity(E) with it to form a new entity(E').

$$E' = NN/NNS/NNP/NNPS\ VB/VBG/VBN/VBP/VBZ\ E$$

For example, "hydrocarbon(NN) contaminated(VBN) antarctic(E)" was extracted for the original entity "antarctic(E)"

Introduction of these two new rules resulted in a total of 1900 entity extractions. A sample of 200 entities from them gave an accuracy of 75 percent.

A reason for low accuracy was that although an extracted entity "nitrate-reducing aromatic" correctly compliments the original entity "aromatic", but "aromatic" is itself an adjective here. On further examination, it was observed that the correct entity should have been "nitrate-reducing aromatic hydrocarbon-degrading bacteria". So if the present seed word was itself an adjective, we looked on the right side it and find potential candidate for a seed word. Therefore we formed a new rule.

Rule 5
If the original entity(E) was an adjective(JJ), look on the right side to find a noun(NN/NNS/NNP/NNPS) as a seed word. Merge them to form a new entity(E').

$$E' = E(JJ) \ldots NN/NNS/NNP/NNPS$$

For example, as given previously, for an entity "aromatic" which itself is an adjective we look at the right side and find that the actual entity should have been "aromatic hydrocarbon-degrading bacteria".
It should be noted that "aromatic" can also be a noun phrase itself. For example, "aromatics" can refer to the set of all aromatic compounds, this problem is easily eliminated by use of POS tags. The POS tag for "aromatics" here would be a noun(NNS) rather than an adjective(JJ).

Rule 6

Merge an adjective(JJ), a conjunction(CC) and an adjective(JJ) on the left side of entity(E) with it to form a new entity(E').

$$E' = JJ\ CC\ JJ\ E$$

For example, "aliphatic(JJ) and(CC) aromatic(JJ) hydrocarbons(E)" was extracted for the original entity "hydrocarbons(E)"

Extending to these rules and addition of some more stopwords resulted in a total of 2015 entity extractions. A sample of 200 entities from them gave an accuracy of 81 percent.

Since all the documents in the corpus can't be analyzed, it is difficult to give a complete set of rules. Therefore the accuracy can be improved by adding more rules.
Also, addition of stopwords will help in improving the accuracy. It is possible that a word acts like a stopword in one sentence and a useful descriptor in another sentence, so its difficult to come up with an extensive list of stopwords. (Note, here by stopwords, we refer to the words that don't add any extra information to the original entity).

We manually went through the extracted entities and relevant entities were added to our list of entities.

2.3 Entity acquisition from web-resources

Despite finding fine-grained entities, our final entity set was still restricted by our initial entity set, that our experts provided us with. But for several of the entity types, we observed that there are a number of web-resources that can provide us with additional entities. Therefore, we scraped these websites for lists of entities:

- BRENDA (http://www.brenda-enzymes.info) - A set of 7974 enzymes
- CyanoBase (http://genome.microbedb.jp/cyanobase/) - A set of 376 microorganisms
- MicrobesOnline (http://www.microbesonline.org/) - A set of 3707 genomes
- metaMicrobesOnline (http://meta.microbesonline.org/) - A set of 1429 metagenomes
- PubMLST (https://pubmlst.org/) - A set of 99 bacteria
- ChEBI (http://www.ebi.ac.uk/chebi/) - A set of 395852 compounds
- GenomeNet(http://www.genome.jp/) - A set of 12034 compounds and 10827 reactions

The entities obtained from these resources were matched with our corpus and entities having a significant term frequency were put into our database.

3 Relation extraction

Once we acquired the entities as above, we then moved on to the extracting relationships among them. Given our goal, we were particularly interested in the following kinds of relationships:

- FoundIn - This type of relation exists between a microbe and environment. It tells us the property of a microbe, where it is found. Information about the place of existence of microbes automatically yields a lot of different properties eg, Pseudomonas is found in sea, here Pseudomonas acts as microbe which is found in sea which works as an environment.

- Of - This type of relation exists between a microbe and its property. It tells us about a particular attribute of the microbe. eg, heterotrophic bacterial population of culturable gasoline degraders, here heterotrophic bacterial population is being treated as entity type microbe which are culturatble gasoline degraders and it is their property.

- ActOn - This type of relation exists between microbes and substrates. It tells the type of substances on which a microbe acts on and results in a chemical reaction. eg, Pseudomonas acts on acetone, here Pseudomonas is a microbe acting on acetone which is a substrate

- CarriedOn - This type of relation exists between microbe and reaction/process. It predicts the type of reaction in which a microbe participates. Knowing about the reaction a microbe participates in helps in identifying about the properties of a microbe.This type of relation exists between a process/reaction and substrate also, apart from microbe and process. It identifies the type of reaction carried on a substrate to produce end product. eg, hydrocarbon degradation carried on acetone by Pseudomonas, here reaction of hydrocarbon degradation is carried on acetone which is a substrate by a microbe, Pseudomonas.

- Produce - This type of relation exists between microbe and enzyme. It identifies the type of enzyme being produced by a microbe after a chemical reaction. eg, Pseudomonas produces carboxylase, here Pseudomonas which is a microbe produces carboxylase, an enzyme.

- OccursIn - This type of relation exists between reaction/process and environment. It tells us about the possible places where a reaction can take place. Indirectly, it tells us about the environment in which a microbe involved in that reaction can be found. eg, Hydrocarbon degradation occurs in sea, here reaction of degradation of hydrocarbons takes place in environment which is found near to sea.

- Inhibit_Catalyse- This type of relation exists between enzymes and process/reaction. The role of an enzyme is to help in either inhibition or catalysis of a reaction or a process which is predicted by this relation. eg, Oxygenase helps in inhibition of photosynthesis, here oxygenase is an enzyme which inhibits the process of photosynthesis.

3.1 Methodology for relationship extraction

3.1.1 Creating Data Set

To extract the above defined relations, we created a gold standard, a training set and a test set of research articles.

1. Gold Standard - We hired people who have expertise in this area. These people manually went through 40 documents to give 3000 relations. These relations are assumed to be true since these have been manually extracted by subject matter experts.

2. Training Set - To train our machine learning models, training set was prepared using 70% of the relations from the gold standard prepared earlier. Training was done using both positive and negative set of relations. Negative set of relations include those which are between same entity types but convey different meanings.

3. Test Set - To check the validity of the machine learning models, testing was done on the remaining 30% of the articles from the gold standard and the results were compared against the gold standard documents to check for precision, recall and F-Score.

3.1.2 Different Algorithms

1. Co-occurrence Model - In this model, we simply made all the relations that could possibly exist in a document using the specified set of rules as described in section 3.1.1. For example, consider a sentence such as Pseudomonas lives in sea acting on acetone. Since, one sentence contains entity types microbe, environment and substrate, all the relations which involved any two of these three entity types are made using this model. In general, if a sentence contained any of two entity types which could result in a relation, it is made. The results obtained using this were not good and have been listed below in the table.

2. Dependency Distance - Co-occurrence model resulted in a lot of ambiguous relations. Even if two entities were quite far in a sentence separated by commas, a relation was made between the entities. To remove such types of ambiguous relations and making the relations cleaner, dependency distance model was used. Dependency distance was calculated between the interested entities in a sentence using Stanford Core NLP package. Stanford Core NLP uses neural network internally to calculate the dependency distance with the help of a tree kind of data structure. This dependency distance is used as a feature in the future machine learning models to predict the relations.

3. Weighted bag of words - We used a model similar to bag of words. Weight was calculated using the gold standard relations. For each type of relation, all the words between the positive relations are considered together and their weight is found. Weight for each word is found by dividing the frequency of the word with the total number of the words. Similarly, the words found in the negative relations are given negative weight in the same way. To predict a new relation, weights of all the words between two entities are added and compared with a threshold. We manually calculated precision for 2 relations (carriedon and occursin) for a small sample of around 200. Precision obtained is around 0.37 and 0.35, respectively which is very less. So, this model was rejected.

4. Logistic Regression - We used logistic regression with various variations in the feature vector to predict the relations.

   (a) Positive Relations - We used the model to predict the relation when the training set consisted only positive relations. Feature vector used in this case included binary 0 or 1 for all the words in the dictionary. 1 was assigned to those words which were present in between the two entities, 0 otherwise. Accuracy for this was calculated with different splits ($65\% - 35\%$ and $70\% - 30\%$) between training and testing data.

   (b) Equal positive and negative relations - We used the same model as described above with equal number of positive and negative relations for each type of relation. Accuracy for this was also calculated with different splits ($65\% - 35\%$ and $80\% - 20\%$) between training and testing data. Feature vector for this included various variations such as -

      i. All the words in between as feature

      ii. All the words in between as feature along with the type of entity

      iii. All the words in between as feature along with the entity type and number of words in between

   Accuracy obtained for the above three cases has been summarized below in the table.

5. Random Forest - We also used the random forest algorithm to predict relations. We used the random forest provided by scikit module with default set of parameters to predict the relations. Random Forest greatly improved the F-Score for the relations which had sufficient number of training examples (occurs in and carried on). Results have been summarized below in the table.

6. Neural Network - To improve upon the F-Score further, multi-layer perceptron algorithm was used that uses back propagation for training. One hidden layer with 100 hidden unit was used for training. 'Relu' was used as activation function along with 'Adam' solver. Other parameters were used as default defined in MLP Classifier of sklearn. Feature vector for this included variations such as -

   (a) Feature vector included words obtained after lemmatization and stemming done using wordnet lemmatizer and Snowball stemmer.

   (b) Feature vector included words without lemmatization or stemming along with dependency distance, number of words in between as the feature.

   F-Score obtained for the above cases has been summarized below in the table.

3.2 Results

1. Co-occurence Model -

After Web Scrapping

Relation	Precision	Recall	F-Score
FoundIn	0.346	0.945	0.507
Of	0.316	0.756	0.446
ActOn	0.168	0.91	0.283
CarriedOn	0.371	0.933	0.531
OccursIn	0.388	0.926	0.547

2. Logistic Regression -

   (a) Positive Relations -

Before Web Scrapping (Average of two Splits)

Relation	Training Accuracy	Test Accuracy
FoundIn	0.946	0.927
Of	0.983	0.946
ActOn	0.954	0.902
CarriedOn	0.964	0.949
OccursIn	0.942	0.923

   (b) Equal Positive and Negative Relations

      i. All words as feature -

Before Web Scrapping (Average of two Splits)

Relation	Training Accuracy	Test Accuracy
FoundIn	0.884	0.7
Of	0.912	0.668
ActOn	0.905	0.747
CarriedOn	0.859	0.759
OccursIn	0.857	0.726

      ii. All words along with entity type as feature -

Before Web Scrapping (Average of two Splits)

Relation	Training Accuracy	Test Accuracy
FoundIn	0.974	0.979
Of	0.989	0.992
ActOn	0.971	0.974
CarriedOn	0.98	0.966
OccursIn	0.96	0.935

      iii. All words along with entity type and number of words in between as feature -

Before Web Scrapping (Average of two Splits)

Relation	Training Accuracy	Test Accuracy
FoundIn	0.978	0.968
Of	0.993	0.969
ActOn	0.97	0.977
CarriedOn	0.978	0.971
OccursIn	0.962	0.933

3. RandomForest -

Before Web Scrapping

Relation	Precision	Recall	F-Score
FoundIn	0.168	0.246	0.2
Of	0.577	0.369	0.45
ActOn	0.214	0.932	0.348
CarriedOn	0.761	0.607	0.675
OccursIn	0.651	0.921	0.763

After Web Scrapping

Relation	Precision	Recall	F-Score
FoundIn	0.892	0.193	0.317
Of	0.629	0.351	0.451
ActOn	0.609	0.736	0.667
CarriedOn	0.813	0.622	0.704
OccursIn	0.855	0.908	0.881

4. Neural Network -

Before Web Scrapping

Relation	Precision	Recall	F-Score
FoundIn	0.254	0.193	0.219
Of	0.447	0.189	0.266
ActOn	0.226	0.939	0.364
CarriedOn	0.833	0.79	0.811
OccursIn	0.383	0.925	0.541

After Web Scrapping

Relation	Precision	Recall	F-Score
FoundIn	0.839	0.152	0.257
Of	0.612	0.712	0.658
ActOn	0.559	0.865	0.679
CarriedOn	0.765	0.827	0.795
OccursIn	0.851	0.958	0.902