Biomedical Term Classification

Vasile Rus, PhD
Assistant Professor of Computer Science
The University of Memphis
vrus@memphis.edu

1. Introduction

Biomedicine studies the relationship between the human genome and human health. It addresses issues such as diseases and aging from the genome perspective. Discoveries in the area of biomedicine could have dramatic effects on the wellbeing of humans. For instance, new drugs could be developed that would treat major diseases such as hypertension or Alzheimer. Biomedicine is a very promising field with a huge growth potential. Computers are at the forefront of biomedical research and thus it is beneficial for computer science students to be exposed to issues in biomedical research.

Due to the explosive growth of knowledge in biotech (about 1500 research abstracts are added every single day to MEDLINE, an electronic repository of biomedical papers) researchers have difficulties keeping track with the latest information in their area of interest. An acute need for knowledge management tools has arisen. Knowledge in scientific articles is encoded in natural language and thus techniques that process human language are needed. Natural Language Processing (NLP) offers the necessary technologies to organize, mine, and naturally access large collections of text or text combined with other types of media such as tables, charts, and images. BioNLP is a new field at the intersection of biomedical and NLP technologies.

A major problem in BioNLP is that same biomedical term can be frequently used with different meanings in biological texts. For instance, the biomedical term SBP2 can refer both to a protein or a gene. If you are a researcher studying the SBP2 gene, and not the protein, then if you searched in MEDLINE, a collection of biomedical research papers, for articles published recently on the SBP2 gene there is a high chance you would also get articles related to the protein. The researcher would have to manually sort out the gene-related articles from the rest. It would be extremely beneficial if an automated method were available that would classify occurrences of the token SBP2 into genes and proteins. In this project, we combine NLP with Machine Learning techniques, namely decision trees and naïve Bayes, to build software tools that classify biomedical terms. In particular, we focus on terms that belong to one of the following five categories: DNA, RNA, protein and cell_line, cell_type.

MEDLINE is the largest component of PubMed (http://pubmed.gov), the freely accessible online database of biomedical journal citations and abstracts created by the U.S. National Library of Medicine. Approximately 5,000 journals published in the United States and more than 80 other countries have been selected and are currently indexed for MEDLINE. MEDLINE contains over 16 million references to journal articles in life sciences with a concentration on biomedicine. MEDLINE is available online at the following link http://www.ncbi.nlm.nih.gov/ .

2. Project Overview

In this project, we develop software classifiers that categorize words or groups of words that denote a biomedical term into five classes: DNA, RNA, protein and cell_line, cell_type.

There are two issues related to classifying biomedical terms in scientific articles. First, the terms must be delimitated from the surrounding words in the sentence. This first step is called *biomedical term recognition*. Second, the delimited terms must be *labeled or categorized* into a biomedical class such as DNA or RNA. In this project, we assume the biomedical terms are already delimited and thus we focus on the classification step.

We will work with a dataset of 300 sentences extracted from MEDLINE that contain biomedical terms belonging to the five classes mentioned above. The 300 sentences are a subset of the data set used in the Shared Task of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm)**.** The plan is to use machine learning to develop a software tool that could automatically classify biomedical terms into biomedical term classes. We will use instances of biomedical terms in sentences, which we already know what category they belong to, to train a machine learner on how to categorize biomedical terms and then use the learner to automatically classify new instances of terms.

3. Project Description

As many projects in machine learning, this project is split into three major parts: data collection, feature extraction, and machine learning. These parts are also phases in the overall process of biomedical term classification from research articles and classification of new instances (labeling). As this process is interactive and iterative in nature, the phases may be included in a loop structure that would allow each stage to be revisited so that some feedback from later stages can be used. The parts are well defined and can be developed separately (e.g. by different teams) and then put together as components in a semi-automated system or executed manually. Hereafter, we describe the project phases in detail along with the deliverables that the students need to submit on completion of each stage.

Phase 1 consists of collecting a set of 200 sentences containing terms that belong to the five categories that we target in our study. Because expertise in biomedical area is needed, we will use a pre-existing set of annotated sentences. However, students will be asked to try to collect several sentences by themselves in order to understand the issues with collecting data. The biomedical terms in the pre-existing set of sentences will serve as our training data. Phase 2 involves feature identification, feature extraction, and data preparation. During this phase the biomedical terms instances will be represented by feature vectors, which in turn are used to form a training data set for the Machine Learning stage. Phase 3 is the machine learning phase. Machine learning algorithms are used to create classifiers of the data sets. These classifiers are used for two purposes. The accuracy of the initial data set is evaluated and secondly, new terms are classified into existing topics.

Phase 1 – Collecting data

The first phase consists of collecting a set of sentences containing biomedical terms belonging to different classes. Because of the nature of the domain, namely biomedical, expertise is needed to be able to identify such sentences. While we will not ask students to become experts in biomedicine, we do want to expose students to what it means to collect data.

One could start by querying MEDLINE for articles using keywords such as *human*, *blood cell*, or *transcription factor*. Sentences containing biomedical terms need to be identified. An example of such a sentence is given below.

IL-2 gene expression and NF-kappa activation through CD28 requires reactive oxygen production by 5-lipoxygenase.

The next step is to delimitate terms belonging to one of the five classes we are interested in: DNA, RNA, protein and cell_line, cell_type. The delimitation can be done automatically or manually by an expert. Automatic delimitation is error-prone while manual delimitation requires domain expertise or employing an expensive expert. In our case, we assume the delimitation is already done. For the above sentence, the following biomedical terms are identified.

<DNA>IL-2</DNA> gene expression and <protein>NF-kappa B</protein> activation through <protein>CD28</protein> requires reactive oxygen production by <protein>5-lipoxygenase</protein>.

3.1.1 Phase 1 Deliverables

1. A list of 5 sentences annotated with delimited biomedical terms. The sentences could be collected from MEDLINE or elsewhere. It is fine if there annotation errors because students are not experts.

Phase 2 – Feature Selection and Extraction, Data Preparation

In this phase, every biomedical term delimited and labeled with the corresponding class in the data collection step from Phase 1 will be mapped onto a feature vector representation, which in turn will be used as training data set during the machine learning phase. The mapping process is detailed in the three steps below.

3.2.1. Step 1: Feature Selection

We will characterize each biomedical term occurrence by the sentential context in which they appear. In other words, we will characterize a biomedical term occurrence by the company it keeps, i.e. surrounding words. The number of surrounding words to characterize biomedical terms does matter. A large number of surrounding words would have more discriminative power but would fail to generalize enough. Too few surrounding words would generalize too much and lead to less discriminative power. We will use a window of three words before and after the target term to characterize each instance.

There are many other possibilities to characterize occurrences of a target term. For instance, a window of four or five words could be used or only surrounding nouns can be employed. Different set of features could lead to different models of the biomedical term classification problem. A comparison could be made among different models to see which one best describes the problem. The best model could then be used to classify new instances.

3.2.1.1. Step 1 Deliverables

1. Pick five sentences from the given data set and for every biomedical term in each sentence represent the term in features vector representation.

3.2.2. Step 2: Feature Extraction

In this step, the whole data collection provided in Phase I needs to be mapped onto the features vector representation. Students will have to write a small program the scans one sentence at a time and for each biomedical term it generates the vector representation by printing the three previous words and the three following words of the term, together with the class the term belongs to.

For instance, the term CD28 below will be mapped in the feature vector *<B, activation, through, requires, reactive, oxygen, PROTEIN>*, where last entry in the vector is the class of the term. For terms that occur at the beginning or the end of the sentence, such as IL-2 below, the previous context or following words could be replaced with default words such as *beg1, beg2, beg3,* or *end1, end2, end3* for previous or following words, respectively.

<DNA>IL-2</DNA> gene expression and <protein>NF-kappa B</protein> activation through <protein>CD28</protein> requires reactive oxygen production by <protein>5-lipoxygenase</protein>.

Due to morphological variations of words, we suggest that every word is stemmed before being considered as a feature value in the features vector. The stemming process maps each word variation to its base form. For instance, *go, going,* and *went* would all be mapped onto their base form of *go*. For stemming, we propose to use Porter's stemmer which is freely available at: http://www.tartarus.org/martin/PorterStemmer/.

3.2.2.1. Step 2 Deliverables

1. The data collection mapped into features vector representation.

3.2.3. Step 3: Data Preparation

Next, you will need to create a data set in the ARFF format to be used by the Weka 3 Data Mining System which we will be using in the next phase. The input data to Weka should be in Attribute-Relation File Format (ARFF) format. An ARFF file is a text file, which defines the attribute/feature types and lists all biomedical term feature vectors along with their class value (the biomedical term class).

In the next phase and once we load the ARFF formatted files into Weka, we will be using several learning algorithms implemented in Weka to create classifiers based on our data and to test these classifiers in order to decide which is the best to use.

Weka 3 Data Mining System is a free Machine Learning software package implemented in Java. Weka is available from http://www.cs.waikato.ac.nz/~ml/weka/index.html. Install the Weka package using the information provided on the Weka software page and familiarize yourself with its functionality. Weka 3 tips and tricks are available at: http://weka.sourceforge.net/wekadoc/index.php/en:Troubleshooting

This is one of the most popular Machine Learning systems used for educational purposes. It is the companion software package of the book titled Machine Learning and Data Mining [Witten and Frank, 2000]. Chapter 8 of Witten's book describes the command-line-based version of Weka.

For the GUI version, read Weka's User Guide in the Documentation section at http://www.cs.waikato.ac.nz/ml/weka/. An introduction to Weka is also available in the Documentation section.

Once you have installed Weka and read the User Guide, run some experiments using the data sets provided with the package (e.g. the weather data).

The links below provide additional information on the ARFF format: http://www.cs.waikato.ac.nz/~ml/weka/arff.html .

The next step is to generate the ARFF file for all 300 sentences. You may write your own program to generate the ARFF file or may generate the file manually. The ARFF file will serve as input to Weka in the machine learning phase.

3.2.3.1. Deliverables

1. The ARFF data file containing the feature vectors for all instances of biomedical terms collected during Phase I.
2. A description of the ARFF data file including:

- An explanation of the correspondence between the surrounding words and the attribute declaration part of the ARFF file (the lines beginning with @attribute).
- An explanation of the data rows (the portion after @data). For example, pick a tuple and explain what the feature values mean for the biomedical term that this tuple represents.

Phase 3 – Machine Learning

At this stage, Machine Learning algorithms are used to create models of the data sets. These models are then used for two purposes. The accuracy of the biomedical term training set is evaluated and secondly, new terms, the instances in test data set, are classified into existing classes. For both purposes we use the Weka 3 Data Mining System. The steps involved are:

1. Preprocessing of the biomedical terms data: Load the ARFF files created at project stage 2, verify their consistency and get some statistics by using the preprocess panel. Screenshots from Weka are available at http://www.cs.waikato.ac.nz/~ml/weka/gui_explorer.html.
2. Using the Weka's decision tree algorithm (J48) examine the decision tree generated from the data set. Which are the most important features (the ones appearing on the top of the tree)? Check also the classification accuracy and the confusion matrix obtained with 10-fold cross validation and find out which class is best represented by the decision tree.
3. Repeat the above steps using the Naïve Bayes algorithm and compare its classification accuracy and confusion matrices obtained with 10-fold cross validation with the ones produced by the decision tree. Which ones are better? Why?
4. New terms classifications: Collect new sentences with terms belonging to the five classes, but not belonging to the original data set of terms prepared in project stage 1. Once you collected the new terms, apply feature extraction and create an ARFF test file with one data row for each biomedical term. A data set of 100 new instances, the test data set, is provided to students. Then, using the Weka test set option classify the new terms. Compare their original class with the one predicted by Weka.

Phase 3 Deliverables

1. Explain the decision tree learning algorithm (Weka's J48) in terms of state space search by answering the following questions:
    a. What is the initial state (decision tree)?
    b. How are the state transitions implemented?
    c. What is the final state?
    d. Which search algorithm (uninformed or informed, depth/breadth/best-first etc.) is used?

   e. What is the evaluation function?
   f. What does tree pruning mean with respect to the search?

2. This stage of the project requires writing a report on the experiments performed. The report should include detailed description of the experiments (input data, Weka outputs), and answers to the questions above. The report should also include such interpretation and analysis of the results with respect to the original problem stated in the project.
3. Looking back at the process, describe what changes in the process you think could improve on the classification.

4. Extra Credit Possibilities

1. Repeat Step 3 of Phase 2 using the Nearest Neighbor (IBk) algorithm. Compare their classification accuracy and confusion matrices obtained with 10-fold cross validation with the ones produced by the decision trees and naïve Bayes algorithms. Which of the three models explored here are better? Why?

REFERENCES

[Mitchell, 97] Mitchell, T.M. Machine Learning, McGraw Hill, New York, 1997.

[Witten and Frank, 2000] Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2000.