IDA2: Intelligent Discovery of Acronyms and Abbreviations

By Adam Mallen with advising from Dr. Craig Strube and Dr. Lenwood Heath

A project for the 2009 summer REU program for the MCCS dept. at Marquette University

Introduction

The purpose of the IDA2 project is the development of a database which stores abbreviations and acronyms (short forms) and the associated definitions (long forms) found in Medline abstracts. In addition to this dictionary of abbreviation and definition pairs, the database contains references to all the Medline abstracts which include a given short form/long form pair.

Medline is a collection of over 19 million biomedical publication abstracts available through PubMed [1].

Medline Parser

The first step in building the database requires parsing the XML files in which the Medline citations are stored and using the abstract text as input for the abbreviation finding algorithm. We wrote a Java program which uses LingPipe's built in tools to handle the Medline parsing. LingPipe is a collection of Java libraries and text mining tools for linguistic analysis of human language [2].

Abbreviation Finder

To find abbreviations and the associated expanded forms we implemented the Schwartz and Hearst algorithm for identifying abbreviation definitions in biomedical text [3]. There are two main steps our program performs on each block of abstract text:

1) Identifying short form and long form candidates

Abbreviation candidates are found by matching one of two patterns:

(i) long form (‘short form’) e.g. clinical attachment level (CAL)

(ii) short form (‘long form’) e.g. CAL (clinical attachment level)

Long form candidates are a collection of words in the same sentence as the short form candidates and have no more than min(|A| + 5, |A| *2) words, where |A| represents the number of characters in the short form,

2) Identifying the correct long form

The correct long form is found by starting from the end of both the short form and long form candidates, move right to left trying to match the characters in the short form to characters in the long form candidate except the first character of a short form, which must match the first character in a word.

Take the short form/long form candidate pair

<CAL, clinical attachment level>

as an example. The character matches are shown using capital letters.

CAL would match to Clinical attachment level.

System Design

The Medline Dataset contains 593 baseline files, each containing thousands of citations stored as XML files.

The IDA2 Database contains a record of each instance where the abbreviation finder discovered a short form/long form pair.

ID A2 ENTRY

IDA2 INSTANCE

The IDA2 ENTRY table acts as a dictionary which keeps a record of all the short form/long form pairs found in the Medline abstracts. The IDA2 INSTANCE table keeps a record of each instance where the abbreviation finder discovered a short form/long form pair.

Web Interface

To interact with the IDA2 database we developed a front end user web interface using the Django Python web framework [4]. This web interface allows users to search for all related long forms for a given a short form, all related short forms for a given long form, and links to all Medline abstracts which contain a given short form/long form pair.

Future Work

There are two major tasks that are useful next steps for this project.

1) Clustering different expanded forms which refer to the same concept into one long form. For example, the following short form/long form pairs are currently treated as distinct pairs in the IDA2 database, but since they refer to the same concept, they should be treated as the same entry in the dictionary:

<AD, Alzheimer disease>, <AD, Alzheimer’s disease>,
<AD, Alzheimer type dementia>, and <AD, Alzheimer dementia>

References

The IDA2 database consists of 1,497,702 unique short form/long form pairs, found in 4,126,655 abstracts.

1,116,530 short form/long form pairs are found in only a single abstract.

The most frequently found pair is 'NO' abbreviating 'Nitric Oxide.' It appears in 30,990 abstracts.