Information Extraction of Seminars

- Ontology construction –

1. Data preprocessing

Data preprocessing was required in order to split the header and the body of the e-mail. Keeping the code cleaner and well organized, a separate Python program was created named “parser.py” which implements the Parser class. This will return a tuple of lists, first one consisting of all main headers in the mail(e.g.: ”Abstract”, “Speaker” etc.) and the second list consisting of content of the mail(e.g.: text after Abstract header etc.). Knowing that python dictionaries change the inserting order, the tuple method was preferred instead of using dictionaries. This will help when the new tagged e-mail will be written in order to maintain the same order as in untagged version of the e-mail.

2. Ontology creation

Firstly, all topics from training tagged data were retrieved in order to have an idea of the ontology’s root. Analyzing the resulted list, it comes up that “science” is the best choice for this ontology. This was also split in main areas of it: physics(mechanics, electricity, thermodynamics, optics, cosmology), chemistry(organic, inorganic), biology(botany, anatomy, microbiology), mathematics(algebra, geometry) and computer science(artificial intelligence, networking, graphics, databases, computer architecture, programming languages). Also, a list of key words were inserted manually in the ontology for each topic.

3. Classifying data

Topic tab was retrieved from every mail and check for each ontology entry the similarity using **word2vec.** After this, the greatest mean percentage was chosen and the file name will be inserted in its correspondent ontology tree topic.

4. Conclusion

For improving the program’s output, the same classification can be done for the abstract text but only on key words. A way of finding these words may be creating a counter and retrieve the words between quotes or least used words(e.g.: avoiding words as “the” etc.).