INTRODUCTION

Text mining refers to the application of text-mining techniques to the unstructured text to extract useful information. The volume of published text is expanding at an increasing rate and there is an exponential growth in the scientific publications recently. Researchers have a hard time in figuring out the main concepts of the publications, find material for their reference and the relatedness between papers. In order to gain domain specific knowledge, they have to read huge amount of papers in a short period which is quite impossible. Extracting information manually from the literature is extremely time-consuming and the explosion of information in recent years has made this task almost impractical. As more and more text becomes available a great interest can be observed in extracting useful information hidden in them. Emerging trends of NLP technologies can address the above problem as we are in need of intelligent processing of scientific papers such that they can do information extraction, identifying concepts and recognizing the semantic relation between them.

SemEval 2018 Task 7

Task 7 of the SemEval competition addresses the above mentioned problem: *Semantic relation extraction and classification is to improve the access to scientific literature through NLP technologies.* They have proposed mainly two types of subtasks which are further divided into subtasks.

1) Identifying pairs of entities that are instances of any of the six semantic relations (extraction task)

2) Classifying instances into one of the six semantic relation types (classification task)

Subtask 1: Relationship classification.

This is further divided into two tasks: classification on clean data and classification on noisy data.

Relation classification on clean data is performed on data where entities are manually annotated where entities represent domain concepts specific to NLP, while high-level scientific terms such as “experiment”, “hypothesis”. Relation classification on noisy data is identical to the previous task, but the entities that are annotated contain noise.

Subtask 2: Relationship extraction and classification.

This task combines the extraction task and the classification task. Relation instances needs to be classified into one of the following six relation categories: USAGE, RESULT, MODEL, PART_WHOLE, TOPIC and COMPARISON.

Relationship is considered between two entities X and Y. All relations except COMPARISON are asymmetrical. Directionality of the relation is considered as well. I.e. whether x is before Y or after Y in the context has to be taken into account when doing the classification.

For the project code drop 1, we have implemented subtask 1.1 of the given task, i.e. classification on clean data.

Table 1 Summary on the relation categories of the SemEval task 7.

USAGE	RESULT	MODEL_FEATURE	PART_WHOLE	TOPIC	COMPARE
X is used for Y X is a method used to perform a task Y X is a tool used to process data Y	X gives as a result Y (where Y is typically a measure of evaluation) X yields Y (where Y is an improvement or decrease)	X is a feature/an observed characteristic of Y X is a model of Y X is a tag(set) used to represent Y	X is a part, a component of Y X is found in Y Y is built from/composed of X	X deals with topic Y X (author, paper) puts forward Y (an idea, an approach)	X is compared to Y (e.g. two systems, two feature sets or two results)

METHODOLOGY

In this section, we describe the proposed NLP and machine learning based on relation extraction and classification method which classifies the relations into one the given six relations. The main steps of the proposed approach can be summarized as shown in Fig. 1. When an abstract with relations is given as the input into the model, first all the sentences are segmented. Then, the model classifies each relation into one of the six relations through feature engineering and a learned model.

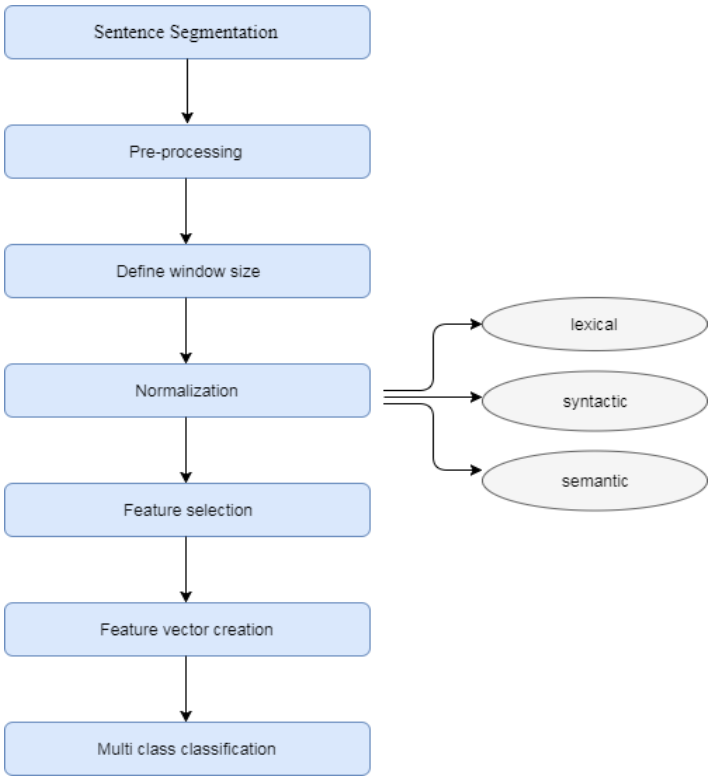

Fig. 1 General pipeline of the proposed approach

A. Sentence Segmentation

Abstracts are identified from the dataset and the sentences with the relations are extracted separately.

B. Pre-processing Steps

Some of the existing NLP techniques and tools are used for preprocessing. Preprocessing is performed as follows.

- All letters are changed to lowercase.
- Special characters and punctuations are removed.
- Stopwords are removed
- Lemmatization / stemming is performed

Stopwords are common words of the language that do not contribute to the semantics of the documents and do not contain any significance but has a high frequency. They are usually filtered out during search queries to prevent returning vast amount of unnecessary information. Lemmatization is the process of grouping together the different forms of a word and return the base or dictionary form of a word so that they can be analyzed as a single word.

C. Define window size

Prior to features selection, for each relationship a window size is defined. It determines the number of words to choose on either side of the entities

D. Normalization

All sentences in the text corpus are preprocessed in order to normalize the corpus as well as to simplify the feature selection. Lexical normalization includes n-grams and TF-IDF (term frequency-inverse document frequency) calculations. N-grams are used to develop bigram or trigram model in the relation. And TF-IDF is used to evaluate how important a word is to the abstract in the dataset. POS tagging and parsing are used in relevant places.

E. Feature selection

The crucial part of the pipeline is feature selection and feature vector generation. During the feature selection, a representative set of features is computed for each relation. Features used in building the proposed model is listed below.

1. Word distance between entities
2. Part-of-Speech tags of both entities
3. Rule based feature that allows to emphasize minority class
4. Part-of-Speech tags of words before/after entities
5. Count of words before/after entities
6. Unique integer ID of the word before first entity and after second entity
7. Unique integer ID of the word after first entity and before second entity in between the entities

F. Feature vector generation

In the final step of the proposed model, a feature vector is generated using the selected feature for each relation as shown in fig 2.

5	1	0	2	3.2	0	0	1

Fig. 2 Generated feature vector

G. Multi class classification

The generated feature vector is used to train a classifier which classifies the relation into the given six categories. Evaluation metrics are used to evaluate the performance of the classifier.

DATASET

Our system uses two datasets provided by the SemEval task 7. Dataset include abstracts of papers from the ACL Anthology Corpus with pre-annotated entities that represent concepts. The dataset provided initially is divided into two subsets: training set and test set. Dataset is in XML format containing abstracts from the articles with annotated entities and a text file containing type of relations for some entities. Training set includes 30 abstracts containing 479 entities and 93 annotated types of relations between entities. And test set includes 10 abstracts containing 131 entities and 25 annotated types of relations between entities.

Due to the changes of our SemEval task a new dataset was provided which has the same format but only training data set was given. And it includes 350 abstracts, 5256 entities and 1228 annotated relations.

FRAMEWORK - CURRENT METHODOLOGY

In this section, we describe the proposed solution that makes use of some basic Natural Language Processing as well as Machine Learning techniques. With the SemEval 2018 Task 7 in mind, framework was written from scratch in Python 3 programming language with use of NLTK and Scikit-Learn libraries. The diagram of the framework is shown on the *Figure 3*.

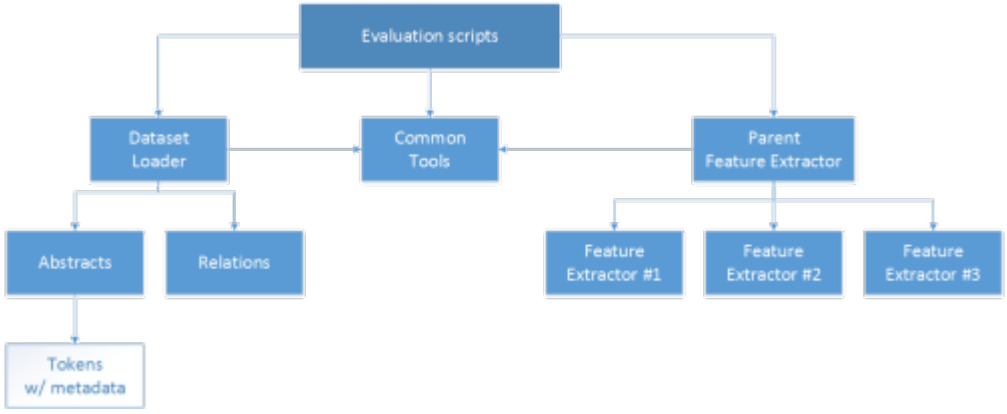

Fig 3. Diagram of the framework.

The framework consists of following main parts:

- Dataset loader allows reading the dataset in both XML file for abstracts and text-file with relations between entities. Each abstract is represented as an instance of a class Abstract, then the content is tokenized and stored as an array of instances of class Object which allows to store metadata like for example entity ID in the particular token. Relations are represented in form of class Relation.
- Common tools, which allows to map class labels to numerical values and vice versa, wrap available classifiers or make use of available metrics (Accuracy, F-Measure).
- Parent feature extractor, which is used for the feature extraction but not directly as it depends on the three Feature Extractors implemented by each team member which can be enabled or disabled in the evaluation scripts.
- Evaluation scripts, used for assessing performance of the proposed solution with both the old dataset as well as the new one.

As for now, four evaluation scripts are implemented which allows to assess performance of:

- Development set with the split 60/40 from the old dataset
- Test set from the old dataset
- Split 60/40 done on the new dataset
- k-Fold Cross Validation done on the new dataset with k = 5

Following features are used from the Feature Extractor 1 in order to achieve reasonable as of time of writing performance:

- Word distance between entities, calculated as a distance between entities (integer)
- Part-of-Speech tags of both entities, consisting of two features which are mapped PoS tags to the numerical values from the last tokens in particular entities
- Rule based feature that allows to emphasize minority class, which retrieves sentence and checks for a pattern which is most common for two imbalanced classes in our dataset

CONTRIBUTION

Our team consists of three members and contribution of each individual is listed below.

1. Chathuri Wickramasinghe
    a. Study of the research problem, related papers
    b. Literature survey
    c. Implementation of some features
    d. Code drop1 presentation preparation
    e. Assistance in writing the files needed for the code drop 1 submission (README FILE - Introduction)

2. Przemyslaw Lukasz Skryjomski
    a. Study and research of the problem
    b. Framework design
    c. Framework implementation including dataset parser, classes used, feature extractors, common tools
    d. Feature extractor with three features used in the proposed solution as they gave the best results
    e. Implementation of evaluation scripts as well as the presentation of the results
    f. Code drop 1 presentation preparation
    g. Wrote BASELINE, INSTALL, ORIGIN file
    h. Wrote the framework part of the README file
    i. Maintaining the repository, provided the scripts

3. Samantha Mahendran
    a. Study of the research problem
    b. Review on the related works
    c. Design of the NLP pipeline
    d. Selection of the set of features
    e. Implementation of some features
    f. Code drop1 presentation preparation
    g. Assistance in writing the files needed for the code drop 1 submission (README FILE - Methodology, Data set)