Evolutionary Optimization and Unsupervised Lexical Acquisition for Bio-Text Mining

A Thesis

Presented to

The Academic Faculty

by

AMBARISH MADHUKAR JADHAV

In Partial Fulfilment

of the Requirements for the M.Tech Degree of

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY PATNA.

April 2015

THESIS CERTIFICATE

This is to certify that the thesis titled **Evolutionary Optimization and Unsupervised Lexical Acquisition for Bio-Text Mining**, submitted by **Ambarish Madhukar Jadhav**, to the Indian Institute of Technology, Patna, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Asif Ekbal
Assistant Professor
Dept. of Computer Science and Engineering
IITP, Bihar, India

Dr. Chris Biemann
Assistant Professor
Dept. of Computer Science and Engineering
TU, Darmstadt, Germany

Place: Patna

Date: 17th April 2015

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisors Dr. Asif Ekbal and Dr. Chris Biemann for their excellent guidance and advice. I would also like to thank Eugen Ruppert and Seid Yimam of the LangTech group at TU, Darmstadt, Germany for their continuous support throughout this project. Their guidance helped me in all the time of research and writing of this thesis.

I would like to thank German Academic Exchange Service (DAAD) for providing me scholarship to conduct my thesis work at TU Darmstadt, Germany.

My sincere thanks also go to the director of IIT Patna and all faculty members of Department of Computer Science and Engineering. I thank all the members of language technology (LT) group at TU Darmstadt for their valuable feedback for my project.

ABSTRACT

KEYWORDS: Feature Selection, Multi-Objective Optimization, Relation Extraction

The growing use of internet and social media all over the world has resulted in huge mine of information over the internet in different forms. There is a wealth of valuable knowledge hidden in this information which, if used wisely, can be helpful towards various research fields. Therefore, to make sense of this information on the internet, all over the world, the industrialists and researchers are striving. Similar efforts are being carried out in bio-medical domain. MedLine, the primary research database serving the bio-medical community, currently contains over 19 million abstracts, with 60,000 new abstracts appearing each month. All of these resources are largely annotated manually, and the costs involved are huge. So, researchers now need an automated system to process this wealth of information to gain useful insights which will be helpful for the health care. In this thesis, we put forth our attempt to create such an automated system. We tried to effectively combine the key concepts viz. "unsupervised lexical expansion", "feature selection", and "multi-objective optimization" for the effective and efficient solution to particular problems in bio-medical domain. To be specific, we discuss the approaches we have used to solve the issues of relation extraction from the bio-medical textual data. We extracted various lexical, semantic and dependency features from the bio-medical texts and tried to find an optimal feature subset using feature selection and muli-objective optimization. We have developed Conditional Random Field (CRF) based event detection and classification system with basic feature sets and also investigated the impact of adding unsupervised lexical acquisition features to the performance of our system which is mentioned in this thesis. The system performed considerably well for Simple and Binding types of events compared to the best performing systems at BioNLP'09, but suffered on relatively Complex events.

Contents

List of Tables

List of Figures

ABBREVIATIONS

IITP	Indian Institute of Technology, Patna
TU	Technische Universitat

Chapter 1

Introduction

The growing number of internet users and increasing use of social media websites, Internet of Things (IOT) in day-to-day life has necessitated the researchers and industries all over the world to ponder upon how to use this huge amount of internet data generated everyday to their advantage. On one side, industrialists are competing to gauze the behavioral pattern of their customers to increase the revenue of their companies. On the other side, researchers are striving to find meaning out of this big digital data to solve few problems for the welfare of humanity and to make a smarter planet. One such effort is being carried out in the field of bio-medicine.

There is a huge amount of bio-medical literature available on the internet and is increasing on day to day basis. There is a wealth of information available hidden inside this magnanimous ocean of information. For example, this information can help capture important clues about various bio-medical phenomenon, bio-medical entities like protein, cell types, genes etc and relations between them. But, to make this information available to researchers in bio-medical community, this information needs to be represented in a specific, easy to understand and easy to use format. So, one way to achieve this is to annotate these resources manually. But, this process is very time-consuming, costly and sometimes error prone due to its large size. Also, for the same reasons, it is very hard to maintain and update previously annotated material to conform to changing annotation guidelines because of the high rate at which this information is increasing. This situation has naturally created an interest in automated systems for problems such as topic classification, entity extraction, event extraction, word sense disambiguation, and tokenization in the bio-medical domain [ARAW00]. In this report, we discuss one such automated system we have developed to address one of the problems of bio-text mining, i.e., event extraction.

Text to Knowledge: Before beginning to understand our system in detail, let us first look at what are the general processes involved in any text mining problem. Figure 1.1

depicts the typical workflow involved in a text mining system. As shown in Figure 1.1,

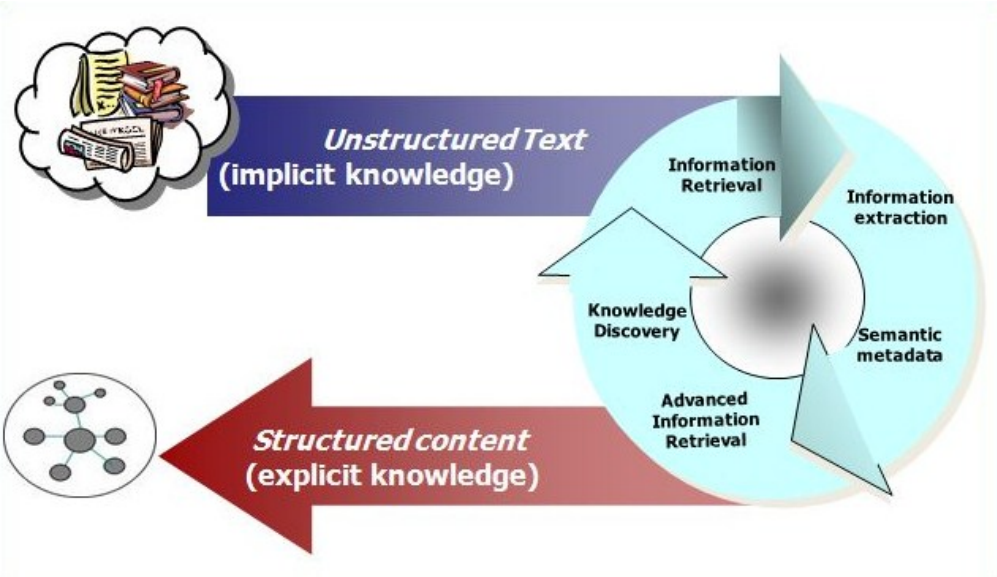

Figure 1.1: Text to Knowledge flow. Source: [bdt]

there exists wealth of unstructured information stored in the form of text books, news paper articles, journals and online libraries. The relevant information required by an application will be extracted using some of the information retrieval mechanisms. Now, to make it comprehensive and easy to use, this information needs to be annotated either manually or automatically using semantic metadata. Then, some advanced information retrieval methods on this annotated data are applied to derive the structured content from it which is actually the knowledge that is useful. So, once this structured knowledge is available to researchers, they are able to make new discoveries from it which can be useful for future.

Challenges of Bio-Text Mining To mine the wealth of information from medical texts various global initiatives have been undertaken like the BioCreative[1], which aims to support curation of PPI (Protein-Protein Interaction) databases such as MINT [ACaC07]. One important finding of such evaluation campaigns is that bio-text mining is comparatively a challenging task for the following reasons:

1. It makes use of **un-structured or un-grammatical language** than a general language usage. So, while processing bio-texts it becomes difficult to understand the structure of sentence.

2. It uses many **tele-graphical phrases**, i.e. the phrases which convey a very straightforward action and which do not contain any exaggerated meaning or adjectives. So, it

[1] http://biocreative.sourceforge.net/bc2ws/index.html

becomes hard to comprehend such phrases.

3. It uses **too many acronyms** for proteins, genes, cell types etc. So, while processing such text, one should either have a mapping from these acronyms to their extensions or one should create a dictionary for these acronyms.

4. **Named entity recognition**, i.e. how can we make our system or computer program understand that certain tokens in bio-texts are protein names, gene names, cell types etc. This is in itself a separate research problem but for the scope of our project, we need not have to address this problem because we are working on datasets already annotated with the proteins in the bio-texts. The details about these annotations will be explained later in this thesis.

1.1 Bio-Molecular Event Extraction

Earlier in this section we explained the need for bio-text mining. In this section, we explain one problem of bio-text mining which is bio-molecular event extraction. So, to begin with let us first understand what one means by an **'event'**. The definition of an event as given by the organizers of BioNLP'09 [KJDJ09] is, **"change on the state of a bio-molecule or bio-molecules due to other bio-molecule or bio-molecules"**. For example consider a sample text from one of the given datasets,

"Leukotriene B4 **stimulates** *c-fos* and *c-jun* gene **transcription** and AP-1 binding activity in human monocytes ."

In this sentence, the tokens in bold are called triggers, i.e. the tokens which indicate events or activities performed and the tokens in italics are called proteins or participants in the events (or Themes in the context of BioNLP'09 shared task) as given in the protein annotations file. Tokens 'AP-1' and 'B4' are also proteins but in this context, they are not considered as proteins because, we only consider proteins given in the protein annotation file. So, in this example, there is a change in the state of bio-molecule 'human monocytes' by another bio-molecule 'Leukotriene B4' and the transformation is '*c-fos* and *c-jun* gene **transcription** and AP-1 binding activity'. Therefore, the goal of the BioNLP'09 is to identity the transformations on bio-molecules and the factors causing these transformations.

In the BioNLP'09 shared task, the events are broadly classified as **Simple** events and **Complex** events. Simple events consist of binary relations between proteins and their textual triggers. So, they are reported by simply detecting the trigger and theme from the text. Complex events consist of multiple relations among proteins, events, and their textual triggers. So, for some complex events, they are reported by identifying multiple related themes and corresponding trigger, some are reported by identifying trigger, associated theme and the cause for this event, some are reported by identifying trigger and the dependent event which itself is reported by its own theme and trigger. The examples of these types of events will be explained later in this thesis.

The initial part of the thesis focuses on building a framework for optimal feature subset selection for event extraction from bio-medical texts. For this, we are referring the BioNLP'09 (Bio-Medical Natural Language Processing) shared task [KJDJ09]. It concerns the recognition of bio-molecular events (bio-events) that appear in bio-medical literature. The shared task focuses on extraction of bio-events, particularly on proteins or genes. For these tasks, proteins and genes are not considered to be different.

To streamline the efforts of researchers to gain valuable insights from data, in this shared task it is assumed that the bio-medical entity recognition is already done. The researchers are provided with the gold standard protein annotations in the data set. The shared task is divided into three sub-tasks to allow separate evaluation of the performance for different aspects of the problem.

Task 1. Core event extraction

The aim of this task is to identify events concerning the given proteins. It involves event trigger detection, event typing, and primary argument recognition.

e.g. CD2 signalling induces **phosphorylation** of *CREB* in primary lymphocytes.
(Type: Phosphorylation (phosphorylation), Theme: CREB)
In this example, the trigger is **phosphorylation** of type Phosphorylation and its theme is the protein *CREB*.

Task 2. Event enrichment (optional)

The aim of this task is to find secondary arguments of events that give more details about the event extracted by Task 1. It involves the recognition of entities (other than

proteins) and the assignment of these entities as event arguments.

e.g. We demonstrate that a fusion protein composed of a C-terminal Hsp70 peptide and the *p50* subunit of NF-kappaB was **directed** into the nucleus of cells, could bind DNA specifically, and activated Igkappa expression and TNFalpha production.
(Type: Localization (directed), Theme: p50, ToLoc: nucleus)
In this example, the trigger is **directed** of type Localization and its theme is *p50*. The token 'nucleus' gives information about the location of the event.

Task 3. Negation and speculation recognition (optional)
The aim of this task is to find negations and speculations (tokens indicating evidence about events) regarding events extracted by Task 1.

e.g. The repetitive activation of T cells (priming) **enhances** the **expression** of many cytokines, such as IL-4, but not others, such as *IL-2*.
(Negation (Type: Positive_regulation (enhances), Theme: Gene_expression (expression), Theme: IL-2)) In this example, there are two nested events indicated by triggers **enhances** and **expression** with theme as *IL-2*

For our specific goal of unsupervised lexical acquisition we only consider Task 1. In all the examples discussed here on wards, the format of the protein and event annotation files is as follows.
For proteins and triggers,

⟨ *entity id, entity type, start index, end index, entity* ⟩

where, *entity id* - unique id for the protein/trigger/event (indicated later as *trigger id, protein id, event id*),
entity type - the type of entity Protein/one of the nine types of triggers (*Gene_expression, Transcription, Protein_catabolism, Phosphorylation, Localization, Binding, Regulation, Positive_regulation, Negative_regulation*),
start index - the start index of the entity in the raw text file,
end index - the end index of the entity in the raw text file,

entity - the actual entity token, i.e. either protein/trigger and it can be a multi-word token also.
and for events,

⟨ *event id, event type, trigger id, protein id/event id* ⟩

Example 1.1.1: Consider the following sample text from a PUBMED abstract.

Leukotriene B4 **stimulates** *c-fos* and *c-jun* gene **transcription** and AP-1 binding activity in human monocytes.

Protein annotation file of the above text:

T1 Protein 26 31 c-fos
T2 Protein 36 41 c-jun
T3 Protein 244 249 c-jun

From the text and protein annotation files given in Example 1.1.1 above, we have to generate the following event annotation file corresponding to Task-1 as given below.

T14 Positive_regulation 15 25 stimulates
T15 Transcription 47 60 transcription
T16 Regulation 129 135 effect
E1 Positive_regulation:T14 Theme:T1

1.2 Feature Selection and Ensemble Learning Using Multi-Objective Optimization

As given on [Deb01], the formal definition of Multi-objective optimization is that it is an area of multiple criteria decision making, that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. In [ES12], the problem of feature selection for only one classifier, namely Maximum Entropy (ME) [KM03] is formulated as a Multi-Objective Optimization (MOO)

problem. A MOO algorithm attempts to optimize more than one objective functions at the same time, which may or may not be related to each other. For example, in case of event extraction task, recall and precision, which are biased towards specific classes and hence not a correct measure, are our objective functions. The increase of one's value may degrade the other's value.

In [ES13], some Simulated Annealing [BT93] based classifier ensemble techniques were developed and those were applied for solving the Part-of-Speech (PoS) tagging problem for different Indian languages. Since, in PoS tagging, each token is assigned a label from a set of predefined PoS tags/classes, and in entity extraction task the problem is to distinguish proper names from others and to classify them into some predefined set of entities, the methodologies are likely to be transferable.

It was shown in [Bie09] that different NE classes as well as different classes of proteins, diseases and genes are distinguished in general and medical sense respectively. But, for our task these entities viz. proteins, diseases and genes are not considered to be different. Therefore, we had to determine a set of features which can cover all of these entities together.

1.3 Unsupervised Lexical Acquisition

In any natural language processing (NLP) problem, knowing the PoS tags of tokens being processed is a very important task. Many times, these tags are achieved by using readily available PoS taggers or a database such as WordNet [Fel98]. But, these systems require huge amount of annotated corpus which is hard to be available especially for bio-medical domain. Also, these systems do not perform well on any text from new domain or any resource poor language. So, there is a need for an automated system which does not require large labeled corpus for processing and instead it learns the patterns in the text on its own and classifies the tokens into different categories based on their semantic similarity similar to PoS tags. Such a system is called unsupervised lexical acquisition system.

Unsupervised PoS induction [Bie09, CC10] is a technique that induces lexical-syntactic categories through the statistical analysis of large, raw text corpora. The derived classes have semantic information like days of the week, professions, mass

nouns etc. along with general PoS tags. In [CBG07], the authors tried to investigate whether using unsupervised PoS tags along with supervised PoS tags helps in named entity recognition and word sense disambiguation. And they observed that performance of unsupervised PoS tagging was very similar to that of supervised PoS tagging for these two tasks. Along with the graph-based method of [Bie09] using word clustering approach described in [Cla03a] may help for PoS induction, as it has been shown in [Bie09].

There is another unsupervised technique for lexical acquisition called *'lexical expansion'* [TMG12]. It requires a large corpus for the induction and is based on the computation of a Distributional thesaurus (DT) [Lin98]. On a broader sense, we can use lexical expansion to replace or to suggest similar words to a given word based on the similar contexts in which these words appear in the whole corpus. So, words are given their similarity scores based on the extent of their contextual similarity and are stored in DT.

Since the induction of PoS is entirely language independent as shown in [Bie09], and DT in [Lin98] uses the contextual similarity concept, we use these features along with our baseline features as they are proved to be effective in different NLP tasks.

1.4 Research Questions

With the above mentioned theoretical background in mind, we started the project with following research questions which we seek to answer:

1. Will the extraction of relevant features for event extraction create a system which is comparable to systems at BioNLP'09?
2. Will the addition of unsupervised features to the baseline features improve the performance of event extraction system?
3. Will the use of MOO algorithms help in finding best feature subset and in turn the best performing system?
4. Will MOO select different set of optimal features if evaluation mode is changed?

How we progressed to answer these questions and did we actually get the answers, will be explained in the further chapters.

1.5 Organization of the Thesis

The rest of the chapters are organized as follows: In Chapter 2, we provide a literature survey of prior works in this domain. In Chapter 3, we discuss our research methods and approaches. In Chapter 4, we present the results we obtained through various experiments. In the end, we conclude with few takeaways for future work.

Chapter 2

Literature Review

Most of the research in bio-medical domain, few years back was mainly focused on bio-medical entity extraction. It is only recently that the research focus in bio-medical domain has shifted to event extraction from bio-medical texts. The reason for this shift is mainly the improved performance of named entity recognition systems in practical applications.

Since then, a lot of initiatives have been taken internationally through competitions and shared tasks to encourage researchers all over the world to address the challenges of event extraction with various approaches. Examples include the TREC (Text REtrieval Conference) Genomics track [WHR07], JNLPBA (Joint Workshop on Natural Language Processing in Biomedicine and its Applications) [JDKC04], LLL (Learning Language in Logic Workshop) [Nt'e05], and BioCreative [LHV07]. While the first two addressed bio-IR (information retrieval) and bio-NER (named entity recognition), respectively, the last two focused on bio-IE (information extraction), finding relations between bio-molecules. BioNLP'09 shared task [KJDJ09], on the other hand, unlike LLL and BioCreative, tried to address the problem of information extraction with complex details.

2.1 Related Work

The problem of entity recognition (viz. proteins, cell types, genes, disease names etc.) was of major focus in bio-medical text mining research community [MKV08], the interest has now grown about the problem of complex event extraction. One of the reasons for this is the availability of annotated corpora from Message Understanding Conferences (MUCs), the Automatic Content Extraction (ACE) evaluations, and the BioNLP shared tasks on event extraction.

Early work on event extraction focused on finding local sentence-level features such as token and syntactic features [RGM05]. However, these features were not enough for

complex events. So, many new approaches were proposed. For example, in [JG08] the authors used global information (considering all documents together rather than considering them individually) from related documents. The idea behind their approach is that an event can appear many times and in different forms, in the same document or in many other similar documents and this fact can be used for event extraction from original document. In another approach [LG10], document-level cross-event information and topic-based features were leveraged while Huang and Riloff explored discourse properties [HR12b]. They argue that, to accurately identify events and their arguments, one has to consider a larger context, i.e. effect of an event across sentence boundaries. They use this information and CRF classifier for event extraction. However, in our system we only consider the events within a sentence boundary, so the cases in which events/themes are present across boundaries our system fails. But, our system can be easily extended to multiple sentence or document level approach to detect such events.

In BioCreative-II challenge [LHV07], one of the task was to extract protein-protein interactions (PPI) from full text articles. One of the reason to include this task was the lack of Gold Standard training data sets due to which it was hard to compare existing automated extraction methods, as most results are reported using author-specific evaluation data sets. Also, some systems have only been evaluated using article abstracts. Another reason was, even though PPI databases were present, it was becoming difficult for the PPI database administrators to keep up with the literature by manually detecting and curating protein interaction information due to discovery of new proteins and growing bio-medical literature. So, the goal of this task was to determine state of the art in PPI. The best performing system got F-measure value of 41% on test set [JH07]. The system uses the concept of inexact pattern matching in which the patterns are actually the structure of the PPIs in the text. This structure is described by the PoS tags, word lemmas, words etc. Due to this strategy, this system does not require pre-annotated corpora or pre-defined patterns. Similarly Learning Language in Logic challenge (LLL) [Nt'e05] was aimed at learning rules to extract protein/gene interactions in the form of relations from biology abstracts. Six teams participated in the challenge and two teams achieved highest F-measure value of around 50%. The system was based on the representation of the examples as sequences.

It is a well known fact that for supervised systems to perform well one needs a large amount of annotated corpora. However, it is often very costly and time consuming to

produce such corpus. So, many semi-supervised and unsupervised approaches were proposed. As proposed in [HR12a], a bootstrapping method is used to extract event arguments using only a small amount of annotated data. Also, in [LR12] a novel unsupervised sequence labeling model was developed. In our system also, we use two unsupervised features viz. unsupervised PoS tag and distributional thesaurus words (explained in further sections). However, unlike the approach used in [HR12a], we use the complete tokenized texts from train, test and development dataset of BioNLP'09 to prepare our unsupervised models.

The BioNLP'09 shared task contains simple events and complex events. The simple events consist of binary relations between proteins and their textual triggers, while the complex events consist of multiple relations among proteins, events, and their textual triggers viz. binding and regulation events. As for our specific goal of unsupervised lexical acquisition, we mainly focused on simple events, i.e. task 1. Among all teams who submitted their predictions in BioNLP'09 our system ranked 6th. Also, for binding events, our system surpassed the performance of the best system. The detailed results of our system and all the teams in BioNLP'09 are given in Chapter 4.

2.2 Feature Selection

As in any other text mining task, feature selection is an important step in the analysis of high-dimensional bio-medical data. Feature selection is the technique of selecting a subset of features for building a model for classification. The reason for selecting a subset is because there can be too many features obtained by analyzing dataset but only a few of them are actually useful for better classification. Apart from identifying relevant features, there are four more reasons for feature selection as explained in [Nav06]:

1. The **computational complexity** of machine learning algorithms is reduced. As the number of features are reduced, the time to build the prediction model is reduced due to the reduction in processing time.

2. It makes the machine learning system **economical** because as we get a subset of best features, we need not check the system performance on the rest not selected features.

3. In many cases it increases the **accuracy** of the machine learning system. Because, presence of non-relevant or weakly relevant features affect the performance of machine

learning system. So, once we know the good features, the accuracy of the system is increased.

4. It provides **valuable insights** into the problem to be solved and these insights are much more important than the prediction problem. For example, to identify whether a patient is sick we are building a model based on the symptoms of sick patients. So, in this case, the sickness of the patient can be determined by many other ways but the features that we would extract related to symptoms will be very helpful in determining the health status of patients.

Instead of trying to tune model parameters for the full feature subset, we now try to derive required model parameters for the optimal feature subset, as there is no guarantee that the optimal parameters for the full feature set are equally optimal for the optimal feature subset as explained in [WD03].

If the feature selection is conducted independent of the classifier as in the t-test [Le03], it is normally referred to as the filter method. If feature selection uses the classifier to evaluate the performance of each subset as in sequential floating forward selection (SFFS) [PP94], it is normally referred to as the wrapper method. Feature selection can also be a combination of both like the generalized wrapper method. In some cases, feature selection is strongly coupled with the classifier design, as in boosting or recursive ridge regression [FL05], which is referred to as the embedded method. For our project, we will use the MOO approach for optimal feature selection which will be explained in later sections.

2.3 Ensemble Learning

A single machine learning classifier may not always be sufficient for a problem. One classifier may not perform well on a dataset with less features while another classifier may perform well on a same dataset but with more features. One classifier may give a better precision while another classifier may give better recall on the same dataset and features. So, we need to find best possible combination of classifiers with optimal parameters to achieve greater accuracy. This is where ensemble learning is used. Various Ensemble Learning approaches have been studied in the past such as Bagging [Bre96], Boosting [Sch99], ensemble of features and so on.

Bagging proposed in [Bre96] is based on Bootstrap sampling [ET93]. In a Bagging ensemble, each base learner is trained on a set of n training samples, which are drawn uniformly at random with replacement from the original training set of size n. And the final prediction is made by simple averaging.

AdaBoost is one of the best known variations of Boosting [Sch99]. Its main idea is to introduce weights on the training set D and pay more attention to those training samples that are mis-classified by former classifier in the training of next classifier.

In the approach used in [LM99], all input variables are first grouped based on their mutual information. Statistically similar variables are assigned to the same group. Each base learner's input set is then formed by input variables extracted from different groups. The designed ensembles have been successfully applied to drug design [Mam03] and medical diagnosis [ATP03].

2.4 Evolutionary Optimization for Feature Selection and Ensemble Learning

As explained in previous sections, we incorporated evolutionary MOO algorithm and ensemble learning for optimal feature subset selection. Evolutionary MOO algorithm is often used to search the optimal trade-off between different objectives using a population (a group of candidate solutions) of chromosomes (a candidate solution in case of Genetic Algorithms) and biasing toward the Pareto front (a set of non-dominating candidate solutions explained later) in parallel and at the same time maintaining population diversity to obtain as many diverse optimal solutions as possible [CY06]. These properties are very useful in ensemble design. From [Opt99], [Ho98], it can be observed that varying the feature subsets used by each member of the ensemble can help achieve this necessary diversity.

Traditional feature selection algorithms aim at finding the best trade-off between features and generalization. In addition to this, ensemble feature selection tries to find a set of feature sets that will make the component members of the ensemble diverse [LSOS06]. The Random Subspace Method (RMS) proposed by Ho in [Ho98] was one early algorithm that constructs an ensemble by varying the subset of features.

More recently, some strategies based on Genetic Algorithms (GAs- a type of evolutionary algorithm which uses techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover to generate solutions to optimization problems) have been proposed [Opt99]. All these strategies claim better results than those produced by traditional methods for creating ensembles such as Bagging and Boosting.

It has been demonstrated that feature selection through Multi-Objective Genetic Algorithm (MOGA) is a very powerful tool to find a set of good classifiers, since GA is quite effective in rapid global search of large, non-linear and poorly understood spaces [LSOS03]. Besides, it can overcome problems such as scaling and sensitivity towards the weights. Also in [KS00], it has been concluded that GAs are suitable when dealing with large-scale feature selection (number of features is over 50) after comparing it with other feature selection algorithms.

2.5 Conclusion

This chapter explained the previous work carried out in bio-text mining and the various initiatives undertaken. It also provided the details of the some of the existing techniques and tools for event extraction in bio-medical texts. We also discussed the rationale behind feature selection and evolutionary optimization for feature selection. In next chapter, we discuss our approach and tools to address this issue.

Chapter 3

Research Methods and Approach

In this section, we describe in detail the methods used to address issues in event extraction from bio-medical texts and the various approaches we followed to answer our research questions.

3.1 Unsupervised PoS Tagging

As we know that for any NLP problem, knowing PoS tags of words to be processed is an important task. An obvious solution to automate this task is, using a supervised approach as proposed in [Cha93], in which the authors use statistical methods. But, again, this approach isn't feasible because it needs a huge amount of labeled data which is costly and time-consuming to produce. Also, it does not work well on unstructured texts (like Tweets) and texts from different domains. This paved the way for a domain agnostic system which can categorize texts as accurately as a supervised system would do. And one such system is the unsupervised PoS tagging.

Unsupervised PoS tagging is a technique that requires no pre-existing manually tagged corpus to build a tagging model. Thus, the problems associated with supervised system as mentioned above are largely alleviated. Various techniques have been proposed for unsupervised PoS tagging in the past. For example, in [PFBM92], a class-based n-gram model is proposed, which is the oldest and simplest unsupervised PoS tagging system. It uses a bi-gram model in which each word is assigned a probability of corpus and the system tries to optimize the probability of the corpus under this model. Quite similar approach was followed in [Cla03b], in which class based n-gram model and clusters of word types were formed. The only difference is that in this case the system augments the probabilistic model with a prior that prefers clustering where morphologically similar words are clustered together. A comparative evaluation of seven unsupervised PoS systems was performed in [CC10] using the Wall Street

Journal (WSJ) corpus. The authors concluded that incorporating morphological features is generally helpful for PoS induction because two of the seven systems that used morphological information for PoS induction performed better than the other five.

3.1.1 A Use Case

The system implemented in [Bie09], takes as input, a considerable amount of unlabeled and tokenized text without any POS information mentioned in it and then induces the number of word clusters. A graph clustering algorithm called *Chinese Whispers* which is based on contextual similarity is used in two stages. The principle parameters of the algorithm are the number of target words, feature words, and window size. The most frequent target words, based on their context statistics, within a context of the most frequent feature words which appear either to the left or right of a target word, are clustered in first stage. In the second stage, pairwise similarity scores calculated by the number of shared neighbors between two words in a four-word context window is used. The clusters obtained in these two stages are then combined to form the final clustering. The pseudocode for this algorithm is shown below:

Algorithm 1 Graph clustering based Chinese Whispers

```
for all v_i ∈ V do
    class(v_i) = i
end for
for it = 1 to number-of-iterations do
    for all v ∈ V, randomized order do
        class(v) = predominant class in neigh(v)
    end for
end for
return partition P induced by class labels
```

In simple words, using the contextual similarity of words the system tries to categorize words in different iterations. For e.g. words like *activates, stimulates, prevents* will fall in same category while words like *transcription, prescription, description* will fall in different category. Eventually, after certain number of iterations each word is assigned to a cluster, i.e. the PoS tag of the word. Using unsupervised PoS approach, the author in [Bie06] claimed language-independence by creating and evaluating a unsupervised PoS tagging system and evaluating it against three languages. Even though unsupervised PoS system yields results which are little bit different from a linguisti-

cally motivated POS-tagger, they are found to be comparable to the widely used PoS taggers [Bie09]. Therefore, in our experiments, we use this system to provide one of the features for event extraction.

An Example: To generate unsupervised PoS tags, we have to first convert a raw text into tokenized text and then apply a script[1] to get the required output. For example, consider the following raw sentence from one of the abstracts.
"Functional interaction between the two zinc finger domains of the v-erb A oncoprotein." After applying tokenization and unsupervised PoS script, we get,
" Functional|316 interaction|2 between|491 the|463 two|378 zinc|2 finger|14 domains|2 of|391 the|463 v-erb|3* A|489 oncoprotein|3* .|485"
So, the number besides each token is the category to which it belongs,
e.g. all proteins in this sentence belong to cluster 3* (v-erb|3*, oncoprotein|3*).

3.2 Distributional Thesaurus (DT)

Similar to a normal thesaurus, which lists words grouped together according to similarity of meaning (like synonyms and sometimes antonyms), DT groups the words together according to the distribution similarity index. For certain high-frequency words, DT finds the most similar words to the given word from the whole text which appear in similar contexts as explained in [Har09]. The problem in which the goal is to assign a sense or meaning to a word in a given sentence by comparing its context with its dictionary definitions, is called word sense disambiguation (WSD) problem. In [TMG12], the authors tried to test the assistance of DT words for WSD problem. *Lexical gap problem* is one hurdle that is faced by techniques which try to match sense definitions from dictionaries with contexts of ambiguous terms to assign the correct sense. The *'lexical gap problem'* arises when we do not have sufficient words to express a certain concept or meaning. To counter this problem, the concept of lexical expansion was used in [TMG12] with the help of DT words. So, each word was then lexically expanded by using similar words obtained from a DT which was constructed from extracting the contextual information from the unlabeled corpora. In this way, the authors in [TMG12] demonstrated that the approach of using DT words for WSD task performs better than

[1]Chinese Whispers http://wortschatz.uni-leipzig.de/%7Ecbiemann/software/jUnsupos1.0.zip

Tokens from bio-medical abstracts	Similar DT words		
elevation	elevations	increase	rise
interact	interfere	associated	associate
synergistically	cooperatively	directly	additively
found	observed	detected	seen

Table 3.1: An example of DT features

other traditional approaches. An example of lexically expanded words can be found in Table 3.1.

3.2.1 An Example

For our project, we constructed a DT [BR13] from the unlabeled, tokenized training, development and test datasets. And in our experiments, we use the DT as a feature for a token. Specifically, we use the three most similar words obtained from DT (ranked by their distributional similarity score) as the features of a given token. So, if a token is a trigger, then its DT feature will indicate the similar words and in turn similar contexts in which this event can occur and will help in event classification.

As shown in Table 3.1, the first column contains the words from bio-texts, the second column contains the first most similar DT word, the third column contains the second most similar DT word and so on. In our project, one of these three DT words' feature became a part of the best performing feature combination for event extraction system. The details are discussed in the next section.

3.3 Sequence Labeling Machine Learning Model

We use Conditional random field (CRF) [JDLP01] which is a statistical modeling method for building probabilistic models to segment and label sequence data. CRF models have the ability to relax strong independence assumptions made in Hidden Markov Models (HMM) and stochastic grammar models which gives CRF models an edge over them. Maximum Entropy Markov models (MEMMs) and other discriminative Markov models can sometimes be biased towards states with fewer successor states is one shortcoming when it comes to sequence labeling problem. But, this is not the case with CRFs.

Model	PoS tagging	chunking	Named entity recognition
CRF	94.00	92.96	92.88
HMM	91.70	89.23	85.48
MEMM	56.03	48.21	49.09

Table 3.2: CRF vs HMM vs MEMM (accuracy in %). Source: [MA11]

For PoS tagging, chunking and named entity recognition tasks, a comparative study was performed by authors in [MA11]. They created and tested the CRF, HMM and MEMM models on a morphologically rich Hindi language. They observed that in all three tasks mentioned earlier, CRFs performed consistently better than HMMs and MEMMs. The observed results are shown in Table 3.2.

Now, we explain the technical difference between CRFs and HMMs. The graphical model of simple hidden Markov models, maximum entropy Markov models and conditional random fields is shown in Figure 3.1. The circles identified by *Y* are called the *states* and the circles identified by *X* are called *observations*. An indication that the observation is not generated by the model is given in this Figure by a hollow circle.

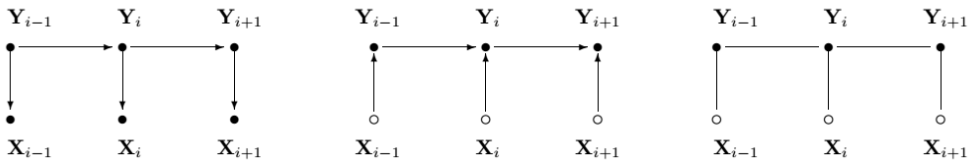

Figure 3.1: Representation of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences in graphical manner. Source: [JDLP01]

As we can see in Figure 3.1, the model for CRF is an undirected acyclic graph and given the observation sequence CRF has a single exponential model for the joint probability of the entire sequence of labels. In case of HMMs, we need to enumerate all possible observation sequences but it is not practical to represent multiple interacting features or long-range dependencies of the observations especially in case of bio-texts. MEMM uses per-state exponential models for the conditional probabilities of next states given the current state.

Definition 1 describes the formal definition of CRF in which X is a random variable over data sequences (i.e. word sequences) to be labeled and Y is another random variable over corresponding label (i.e. tag) sequences. In other cases, X might range over language sentences and Y range over named-entity taggings of those sentences with γ

be the set of all possible named-entity tags. This is the scenario in a NLP sequence tagging problem.

Definition 1 *Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Yw, w \sim v)$, where $w \sim v$ means that w and v are neighbours in G.*

We have earlier explained the better performance of CRFs over other techniques for performing various sequence labeling tasks. CRF is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. Given an observation sequence $o = (o_1, o_2, ..., o_T)$, the conditional probability of a state sequence $= (s_1, s_2, ..., s_T)$ is calculated as:

$$P_v(s|o) = \frac{1}{Z_0} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k * f_k(s_{t-1}, s_t, o, t)$$

where, $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight λ_k is to be learned via training. Typically the values of the feature functions are binary but in some cases, they may range between $-\infty, ... + \infty$. We must calculate the normalization factor,

$$Z_o = \sum_{s} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k * f_k(s_{t-1}, s_t, o, t)$$

, which as in HMMs, can be obtained efficiently by dynamic programming, to make all conditional probabilities sum up to 1. The objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequences to train a CRF model:

$$L_v = \sum_{i=1}^{N} \log(P_v(s^{(i)}|o^{(i)})) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

where $(o^{(i)}, s^{(i)})$ is the labeled training data. The optimization if facilitated by making the likelihood surface strictly convex through the second sum which corresponds to a zero-mean,σ^2-variance Gaussian prior over parameters. To maximize the penalized log-likelihood using limited-memory BFGS [SP03], a quasi-Newton method that is significantly more efficient, which results in only minor changes in accuracy due to

changes in λ we set parameters λ. A feature function $f_k(s_{t-1}, s_t, o, t)$ is set to be 1, when s_{t-1}, s_t are certain states and the observation has certain properties, otherwise, for most cases, it has a value of 0.

We have used an open source C++ based CRF++[2] [JDLP01] tool kit for building probabilistic models. CRF++ requires the training and test data in a specific format for its commands to work correctly. The training and test data has to be in CoNLL[3] format in which columns represent different features and rows indicate the instances of tokens (from which these features are extracted) and their different feature values. The Figure 3.2 shows an example of part of a training file in this format. It can be seen that the last column in each row viz. NULL, B-NP, I-NP are the labels assigned to the tokens which occupy the first column. All other columns contain the features extracted from these tokens. In our training files, the first column is the token itself, last column is the event type (one of the ten classes including the Other class) while rest are the feature columns.

```
Reactive 5 amod 1 0 1 0 NULL NULL oxygen intermediate-dependent NF-kappaB NULL
oxygen 5 nn 0 0 0 1 Reactive NULL intermediate-dependent NF-kappaB activation B-NP
intermediate-dependent 5 nn 0 0 0 0 oxygen NULL NF-kappaB activation by I-NP
```

Figure 3.2: A part of training file in CoNLL format

Each line contains information about single token in a sentence while sentences are separated by a blank line between them. The feature columns are separated by either space or tabular characters. It is obvious that the training and test files should have same number of features for each token, i.e. same number of feature columns except that test file does not have a class label column.

CRF Feature Template: While creating the CRF model, we have to specify the template file along with training file. A template corresponds to each line in the template file. A special macro **%x[row number, column number]** is to be used to specify a feature in the input data for each template. Here, row number - the relative position of a token from the current token (current token is assumed at position 0), column number - the absolute position of the column in a row. Table 3.3 shows a sample template which can be used to build a CRF model. From the Table 3.3, it can be seen that there are three features specified for each token viz. current token itself, previous token and third feature of current token. So, even if training file has more than four columns, CRF++

[2]CRF++: http://taku910.github.io/crfpp/
[3]CoNLL-X Shared Task http://ilk.uvt.nl/conll/

#Unigram	'U' to specify unigram features
U1:%x[0,0]	current token itself
U2:%x[-1,0]	previous token as feature of current token
U4:%x[0,3]	third feature of current token

Table 3.3: Sample feature template for CRF++

will only consider the features specified at the positions mentioned in feature template file to create a model.

3.4 Multi-Objective Optimization (MOO)

Before starting to understand MOO algorithms let us first understand the difference between single objective optimization problems (SOOP) and MOO problems. When the task of optimization problem involves only one objective function, it is called SOOP. For example, buying a car with the best mileage. In this case, we can easily list out the mileage values of all cars available and select the one with maximum value. However, many a time, in day to day life, we come across situations where we have to deal with more than one objective function, each having its own constraint. At such a time, we have to find a way out of this situation such that all the objectives are achieved and also all the constraints are satisfied. In a technical jargon, this scenario is called as Multi-Objective Optimization problem (MOOP). As an example, consider the same problem of buying a car. But this time, we can set various criteria for the car to meet like affordable price, large interior space, large luggage cabin, minimum maintenance cost, maximum mileage etc. Also, we can add few constraints like the car should have 6 seats for adults and a good stereo system. In this case, our decision variables are all available cars. If we look at these criteria, some are aimed at maximizing certain value, some are within some range of values while others are aimed at minimizing certain values. So, the car has to satisfy all of these criteria equally and at the same time satisfying the constraints. For example, we can not buy a car which has large interior space for 6 adults but very small cabin space or a car which has affordable price but very low on mileage and poor stereo system. So, as per the definition mentioned earlier, the problem of buying a car now becomes a MOOP.

A general MOOP is shown in Equation (3.1).

$$\begin{aligned} &\text{Minimize/Maximize} && f_m(x), m = 1, 2 \ldots, M; \\ &\text{subject to} && g_j(x) \geq 0,\ j = 1, \ldots, J; (\text{ inequality constraints}) \\ & && h_k(x) = 0,\ k = 1, \ldots, K; (\text{ equality constraints}) \\ & && x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, \ldots, n. \end{aligned} \tag{3.1}$$

A vector $(x_1, x_2, \ldots, x_n)^T$ of size n is the solution x, where $x_i; i = 1, \ldots, n$ indicate decision variables. Each decision variable x_i mentioned in the last set of constraints is restricted to take a value within a lower $x_i^{(L)}$ and upper $x_i^{(U)}$ bound and is called a bounded variable. As shown in the equation (3.1) above, there are J number of inequality constraints and K number of equality constraints for this optimization problem. Constraint functions are defined by the terms $g_i(x)$ and $h_k(x)$. A solution x that does not satisfy all of the $(J + K)$ number of constraints and all of the $2N$ (total population size including the parent and offsprings) variable bounds is called an *Infeasible solution* while on the contrary, a solution y which satisfies all variable bounds and constraints is called a *feasible solution*.

The variable m in the expression $f_m(x)$ as shown in equation (3.1), can take any value from 1 to M. This means one can define more than one objective function as minimization or maximization corresponding to each value of m. One more important point to note here is that the objective functions form a multi-dimensional space in MOO problems unlike single-objective optimization problems, in addition to the usual decision variable space that is common in both types of optimizations. Also, in MOO problems mapping takes place between an n-dimensional solution vector and M-dimensional objective vector while on the other side, in single-objective optimization problems mapping takes place between an n-dimensional solution vector and one-dimensional objective vector.

3.4.1 The MOO Algorithms

Equation (3.1) depicts the general form of MOOP. Following illustration is another way to represent the same problem:
For the given M objective functions $f_1(x), f_2(x), ..., f_M(x)$, find the vectors x of deci-

sion variables that simultaneously optimize them while satisfying the necessary objective constraints, if any.

'Domination' plays an important role in the context of MOOPs. If $\forall k \in 1, 2, \ldots, M, f_k(x_i) \leq f_k(x_j)$ and $\exists k \in 1, 2, \ldots, M$, such that $f_k(x_i) < f_k(x_j)$, if we are considering minimization problem, then the solution x_i is said to dominate x_j. This candidate solution x_i is called pareto optimal solution. In other words, a pareto optimal candidate solution is the one which is,

1. At least as good as all other candidates for all objectives, and

2. better than all other candidates for at least one objective.

The set of solutions in which no solution is dominated by any other solution in the same set is called **non-dominated set** and the globally pareto optimal set is the non-dominated set of the entire search space S. Figure 3.3 shows the pareto optimal front of solutions for two maximization objective functions f1 and f2. It can be seen that solutions A and B are non-dominating to each other as f1(A) > f1(B) while f2(A) < f2(B).

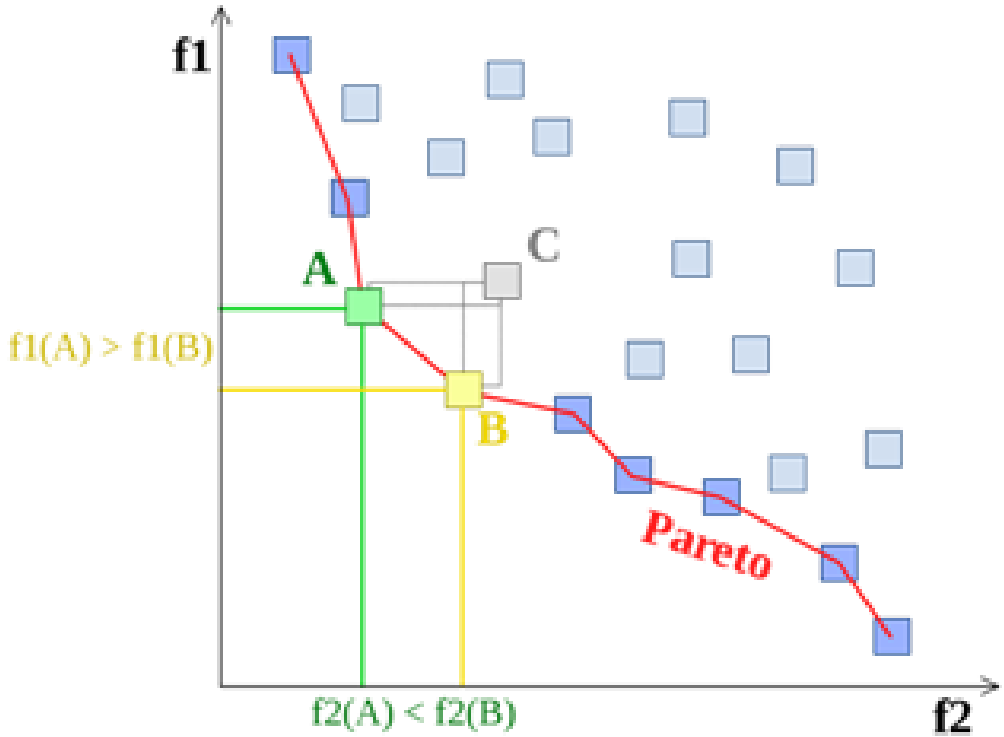

Figure 3.3: Dominance and Pareto-Optimality. Source: [wik]

Similarly, a curve is fit (pareto optimal front) passing through all such solution points which are non-dominating to each other. All points which are above this curve are feasible solutions for this MOOP.

A large number of approaches exist in the literature to solve MOOPs [Deb01]. These are aggregating, population based non-Pareto and Pareto based techniques. We now discuss one such approach we have used in our project to find the best possible set of solutions for event extraction.

3.4.2 Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

As explained in [KDM02], many disadvantages of MOO algorithms using non-dominated sorting were revealed like 1) high computational complexity of $O(MN^3)$(where is M the number of objectives and N is the population size), 2) lack of elitism, i.e. the approach in which best performing solutions are kept and carried forwards for further generations, 3) need to specify sharing parameter to ensure diversity in population. These problems were overcome through a new algorithm in [KDM02] and is proved to be effective for MOO problems. The advantages of the new NSGA–II algorithm are:
1. Low computational complexity of $O(MN^2)$(where M is the number of objectives and N is the population size) for non-dominated sorting,
2. High elitism,
3. No need to specify sharing parameter to ensure diversity in population,
4. Easy to use - NSGA is written in C-Programming language and it provides an interface where we have to specify the objective functions to be maximized or minimized.
Due to these reasons we are using NSGA–II [KDM02] for the purpose of MOO in our experiments. We now discuss the actual algorithm in detail.

NSGA-II Algorithm: The algorithm begins by creating initial random population P_0. This population is then sorted on the basis of non-domination (as explained earlier) and for each candidate, the fitness (objective function value) is assigned a level, the lower the level, the higher the fitness value. This is because this algorithm, by default, tries to minimize objective functions. Thus, while using the NSGA-II software we have to make necessary changes if we are maximizing a objective function. After this, binary tournament selection (a method of randomly selecting a solution with best fitness value from a population in genetic algorithm) is performed on P_0 and a copy of winner is kept in mating pool. Child population Q_0 is then formed through crossover and mutation of parents from the mating pool. Once, the parent and child population are created series of steps are performed in a loop until specified number of generations are reached as

shown in NSGA-II algorithm 2.

Algorithm 2 NSGA-II algorithm

$R_t = P_t \cup Q_t$
$F = fast - non - dominated - sort(R_t)$
$P_{t+1} = \phi$ and $i = 1$
while $|P_{t+1}| + |F_i| \leq N$ **do**
    $crowding - distance - assignment(F_i)$
    $P_{t+1} = P_{t+1} \cup F_i$
    $i = i + 1$
end while
$Sort(F_i, <_n)$
$P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$
$Q_{t+1} = make - new - pop(P_{t+1})$
$t = t + 1$

Step 1: At any generation t, P_t is the parent population selected, Q_t is the child/offspring population generated from P_t and R_t is a combination of parent and child populations. As P_t and Q_t are of same size N, R_t has size 2N. Non-dominated sorting is then applied to R_t population. All the non-dominated fronts of $P_t + Q_t$ are copied to the parent population one by one as shown in Figure 3.4 to prepare parent population $P_t + 1$ for next generation.

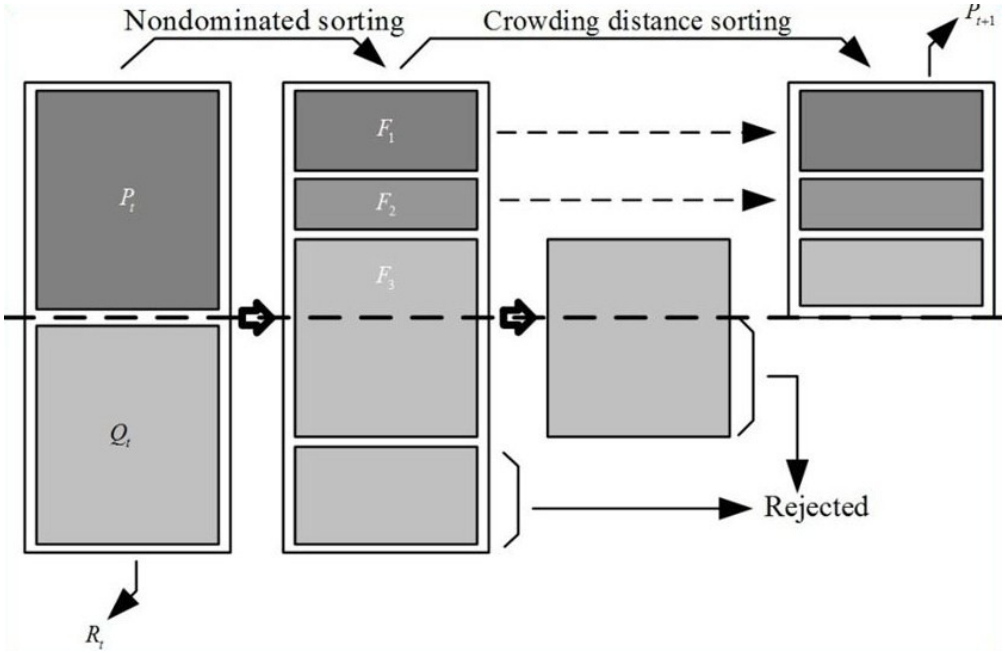

Figure 3.4: Working of NSGA-II. Source: [CV12]

Step 2: When the size of parent population $P_t + 1$ gets larger than the population size N, the algorithm stops adding individuals to it. The individuals that make the parent population size larger than N in the last accepted rank (it is shown in Figure 3.4 by F3),

are sorted by crowded distance sorting [KDM02]. Crowding distance is a method to remove unwanted individuals from a population. To fit the required number of individuals for $P_t + 1$, the algorithm first sorts them in each objective domain. A crowding distance of infinity is assigned to the first and last individual in the sorted ranks while for others, the difference of objective values of neighbors is set as crowding distance. If there is tie of non-dominated rank between two solutions, then the solution with lower crowding distance is selected. In all other cases, the solutions with better non-dominated ranks are selected. After this, again selection, crossover (a genetic algorithm operator which is used to generate a new individual by combining two parent individuals from existing population and this extent of re-combination is governed by probability of crossover), mutation (a genetic algorithm operator which is used to alter some parts of newly generated individual and this extent of alteration is governed by probability of mutation) is performed on the new population $P_t + 1$ to create child population $Q_t + 1$ of size N. This process then repeats until the specified number of generations are reached.

3.5 Performance Measures

The evaluations scripts provided by BioNLP'09 shared task for evaluation of our event extraction system reports the results in the standard measures recall, precision and F-measure. To understand this, let us look at an example. Consider a search engine which returns 20 web pages for a search query for which there are actually 30 relevant pages. Out of 20 web pages only 10 are found to be relevant. So, the recall of this search engine is 10/30 and precision is 10/20. In terms of mathematical equations these definitions can be given as follows:

Recall

$$Recall = \frac{\text{relevant elements} \bigcap \text{retrieved elements}}{\text{relevant elements}} \tag{3.2}$$

Recall is defined as the fraction of the relevant elements/items that are retrieved. Thus, recall is actually a measure of completeness of a retrieval system.

Precision

$$Precision = \frac{\text{relevant elements} \bigcap \text{retrieved elements}}{\text{retrieved elements}} \tag{3.3}$$

Precision is defined as the fraction of the retrieved elements/items that are relevant. Thus, precision is actually a measure of correctness of a retrieval system.

F-measure

$$\text{F-measure } = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{3.4}$$

F-measure is the harmonic mean of recall and precision.

3.6 Result and Analysis

The classifiers are trained mostly with the domain independent features that can be generated automatically from the available training data. For event extraction, the classifiers are trained with the set of lexical, syntactic and semantic features as reported in [AES11b]. Then explored the unsupervised lexical acquisition possibilities.

The system for event extraction is evaluated in terms of the standard metrics, namely recall, precision and F-measure.

3.7 Event Extraction

We have developed a supervised event extraction system that performs event detection and classification together (Joint approach). However, as explained in [AES11a], it is observed that if event detection and event classification is performed in separate phases with different combination of features, then there is an improvement in the performance of the system.

So, in the stepwise approach, we separated these phases and carried out various experiments with different combination of features.

3.7.1 Features for Event Extraction

Below are the baseline features we used for event extraction task:

1) Surface form of the word is used as the feature.

2) **Lemma**: Root form of the word is used as feature.

3) **Part-of-Speech (PoS) information**: It is a syntactic feature which tells whether the token is noun, pronoun, verb etc. We use the PoS information of the current and/or the surrounding tokens as the feature. We obtain these values from the Genia Dependency files given by the organizers.

4) **Fixed length word prefix and suffix**: We have considered words of length 4 to find prefixes and suffixes. These play important role in event detection.

5) **Surrounding tokens**: In bio-medical domain, we often come across a pattern of a bunch of tokens which many times occur together in bio-texts. So, it offers a good hint to extracting any bio-medical phenomenon. So, we use four of the preceding and succeeding tokens of the current token as feature.

6) **Lexical Features**: We derive various lexical features for the token such as Capitalization, Hyphenization, Alphanumeric etc. This is an important feature because many bio-medical entities like proteins, cell types etc. possess these characteristics.

7) **Is Protein**: This is a binary feature which is set to one if the token is a protein else it is set to 0. We use the protein annotation files to derive this feature.

8) **Distance from nearest protein**: We analyzed the gold annotation for events and found that for most of the events, the theme is the nearest protein which is given in protein annotation file. So, we determined three protein related features viz. distance of current token from nearest protein, nearest protein and its PoS tag. The distance is defined in terms of the number of words between the current token and the nearest protein and used it as a feature. Lesser the value of this feature, greater is the probability of this or nearby token to be an event.

9) **Nearest Protein/POS**: We find the nearest protein and use it and its POS tag as a feature.

10) **Number of named entities in surrounding context**: Presence of many named entities around a token increases the probability of event in the given sentence. So, a context window of five is considered. So, within this, a word is considered as named entity if it's NE tag is anything except 'O' (other). A non-zero value of this feature increases the chances of occurrence of an event nearby current token. (Named entity tag information is obtained from the Genia Dependency files provided by BioNLP'09 shared task).

11) **2nd nearest protein, distance to current token, its PoS**: It has been observed that for certain events like Binding more than one protein act as a theme. So, we use this

feature along with the nearest protein.

12) **Character Category Pattern Function**[4] : A string representing the type of characters present in the given token, e.g. for string Traf-2 which is a protein this feature will have a value like UuLlLlLlCcNn where U-uppercase, L-lowercase, C-character, N-number.

Apart from these, we incorporated three more advanced features into the system which are as follows:

1. **Dependency relation feature**: For each token, we find the dependency relation of the token with the nearest protein in the given sentence. For this, we use the McClosky Charniak parser [MC08] outputs which are converted to Stanford dependency format.
2. **Unsupervised PoS tag**: As explained earlier, we use the unsupervised tag of the word obtained from the taggermodel built on unlabeled BioNLP'09 dataset.
3. **Distributional Thesaurus**: We use the top three most similar words to the current token obtained from the DT created using the BioNLP'09 dataset.

The impact of separately adding these new features to the baseline is explained in the next chapter.

To see how these features look like in actual. Consider a sentence from an abstract in the development dataset

"Activation of protein *kinase C* and **elevation** of cAMP **interact** synergistically to raise *c-Fos* and AP-1 activity in Jurkat cells."

For token 'raise' 26 of the total 49 feature values are as shown in Table 3.4

3.7.2 Observed Issues During Feature Extraction

While investigating the reason for low scores during initial evaluation, we came across many issues. Following are few of those,

1. Many to one mapping: In many abstracts, the proteins given in .a1 file had separate entries with separate spans however, in the tokenized text (which was used for

[4]CharacterCategoryPatternFunction class from Cleartk https://code.google.com/p/cleartk/

Value	Feature
raise	lemma
I-VP	PoS Tag
VB	chunking Tag
O	NE Tag
to	previous token
c-Fos	next token
0	isProtein
0	distance to nearest protein
c-Fos	nearest protein
B-NP	nearest protein PoS
100	distance to second nearest protein
null	second nearest protein
null	second nearest protein PoS
aise, ise, se, e	suffixes
rais, rai, ra, r	prefixes
amod.dobj	dependency relation
8	unsupervised PoS tag
raise, have, raises	Distributional thesaurus words

Table 3.4: An example of features extracted

feature extraction), all these proteins belonged to a single token.
For e.g., consider one of the protein annotation (.a1) file entry,

T6 Protein 553 557 DCoH
T7 Protein 558 569 HNF-1 alpha

This entry indicates 3 proteins but in the corresponding tokenized file these tokens appear together as *'DCoH-HNF-1-alpha'*.

2. One to many mapping: In many abstracts, the proteins given in .a1 file had single entry however, in the tokenized text, these proteins appeared as separate tokens delimited by some character like '(' and ')'.
For e.g., consider one of the protein annotation (.a1) file entry,

T1 Protein 37 51 p70(S6)-kinase

This entry indicates a single protein but in the corresponding tokenized file these tokens appear separately as *'p70 -LRB- S6 -RRB- -kinase'*.

3. Protein/Trigger as a substring of a token: In many abstracts, the proteins/triggers had their own separate entry however, in the corresponding tokenized file these pro-

teins/triggers were found to be a substring of a non-protein/non-trigger token.
For e.g., consider one of the protein annotation (.a1) file entry,

T1 Protein 24 29 gp-41

and corresponding event annotation (.a2) file,

T1 Protein_catabolism 31 38 induced

These two tokens seem to have separate entries in the corresponding tokenized file however, they appeared as a single token as *'gp41-induced'*. Also, in this case the same token *'gp41-induced'* is protein as well as trigger.

4. Wrong span for proteins: In many abstracts, it was observed that the spans given for proteins were not correct,
For e.g., consider one of the protein annotation (.a1) file entry,

T3 Protein 598 602 CD44
T4 Protein 603 607 CD25

From the above entries, it appears that both these proteins occur one after another however, in the corresponding tokenized file, the token appeared as *'CD44-CD25+'* which clearly indicates that the span for *'CD25'* was wrong.

So, these tokenization issues were addressed during feature extraction as and when they appeared. Due to this, we saw an increase in the evaluation results' score which were very low initially. We were unaware of these tokenization issues prior to beginning our task. We hope that this information will be helpful for anyone who would like to work on BioNLP'09 datasets in future.

3.8 Conclusion

In this chapter, we described the technical approach we followed in this project and the various tools and algorithms that were used. We explained the reason behind using unsupervised features and how we have used these in our thesis. We explained the

setup required for machine learning classifier and multi-objective optimization. We also described the features used for event extraction and issues observed during this process. In the next chapter, we report the results obtained for the various experiments we performed using different parameter sets.

Chapter 4

Experiments and Results

As explained in the previous chapter, we address the event extraction problem using two approaches. This chapter depicts the experiments carried out in that context and results obtained. For all result tables, the values reported for recall, precision and F-measure are in (%) percentages.

4.1 Datasets

For our experiments, we use the benchmark setups of BioNLP'09 shared task datasets. Following Table 4.1 illustrates the summary of the BioNLP'09 shared task datasets:

There are various types of resources provided for training, development and test data sets. They are as follows:

1. protein annotation file - These files contain the annotations for proteins in the abstract.

2. event annotation file - These are the annotations for the triggers into one of the nine event types.

3. raw text file - This is the file in which first line indicates title of the abstract and second line indicates the actual abstract.

Along with these files various analyses were provided. These include the original

Item	Train	Development	Test
Abstracts	800	150	260
Sentences	7,449	1,450	2,447
Words	1,76,146	33,937	57,367
Events	8,597	1,809	3,182

Table 4.1: Summary of BioNLP'09 datasets

outputs of various parsers like Bikel[1], McClosky-Charniak[2], GDep[3] and CCG[4] and also their conversion to Stanford Dependency format.

We are using McClosky-Charniak and GDep parser outputs to extract few features. Because using self-training of McClosky-Charniak parser on unlabeled Bio-Medical abstracts, the authors in [MC08] achieved good performance and GDep parser is the most widely used parser in bio-medical domain. The events were selected from the GENIA ontology based on their significance and the amount of annotated instances in the GENIA corpus. The event types are related to protein biology which means that they take protein as their theme.

As the gold data for test set was not available, we divided our training data into two parts. 440 abstracts were randomly selected for training purpose, 360 for development and 150 abstracts which were previously used for development purpose were later used as a test set. There is a very little difference of approximately +/- 3% between performance measures of best performing systems at BioNLP'09 shared task for test and development sets [RS11]. Therefore, even though the final teams' scores provided in Figure 4.1 are on test sets, we can compare them with our system's scores on development set.

For unsupervised lexical acquisition, we used the benchmark datasets, namely JNLPBA 2004 shared task[5] and AIMed. The JNLPBA datasets were extracted from the GENIA Version 3.02 corpus of the GENIA project. This was constructed by a controlled search on Medline using MeSH terms such as *human, blood cells and transcription factors*. Like GENIA, AIMed also focuses on the human domain, and exhaustively collects sentences from the abstracts of PubMed. But, it selects the different text spans for protein annotation. Unlike GENIA, protein families are not annotated in AIMed. Unlike GENIA and AIMed, GENETAG covers a more general domain of PubMed. We will first of all, use the techniques for induction of PoS and DT with various parameters. Then we will also generate models from the possible space of unsupervised lexical acquisition models.

[1] http://people.csail.mit.edu/mcollins/code.html
[2] http://nlp.stanford.edu/~mcclosky/selftraining.html
[3] http://people.ict.usc.edu/~sagae/parser/gdep/index.html
[4] http://svn.ask.it.usyd.edu.au/trac/candc/wiki
[5] http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html

4.2 Evaluation

The event annotations are provided for training and development data while for test data, the event annotations had to be created for evaluation. We started with the development dataset for evaluating and tuning our system.

We first create a model on training data (800 abstracts) using CRF classifier and we test this model on development data (150 abstracts). We store this output in separate 150 files and convert it to event annotations file format (explained in previous chapter) for evaluation. The evaluation scripts are taken from the BioNLP'09 Shared Task website.

There are 3 modes of evaluation:

1. Strict Span match mode (SMM): In this mode, the spans for triggers must match exactly, the event types must be correct and the event themes must be correct too.
For example, consider one trigger entry in gold data,

T9 Binding 34 39 binds
E1 Binding:T9 Theme:T1

and in the predicted file it appears as,

T9 Binding 34 39 binds
E1 Binding:T9 Theme:T1

Then, as per SMM evaluation, 34 = 34 and 39 = 39, so this trigger entry will be considered as a match to the gold entry.

2. Approximate Span match mode (ASM): In this mode, there is a relaxation for span match. For e.g., if (beg1, end1) is the given span and (ebeg2, eend2) is the gold span, then the spans are considered to match if beg1 >= ebeg2 and end1 >= eend2 (Here- beg, end are the beginning index and ending index of the entity respectively). The other two constraints remain same.
For example, consider one trigger entry in gold data,

T9 Binding 34 39 binds
E1 Binding:T9 Theme:T1

and in the predicted file it appears as,

T9 Binding 36 41 binds
E1 Binding:T9 Theme:T1

Then, as per ASM evaluation, 34 <= 36 and 39 <= 41, so this trigger entry, even though not exactly correct, will be considered as a match to the gold entry.

3. Approximate and Recursive Span match (ASRM): In this mode, there is a relaxation on the requirement for recursive event matching (in case of Regulation events where an event act as a theme for another event), so that an event can match even if the events it refers to are only partially correct.
For example, consider one trigger entry in gold data,

T9 Positive_regulation 34 42 increases
T10 Gene_expression 50 60 expression
E1 Positive_regulation:T9 Theme:E2
E2 Gene_expression:T10 Theme:T2

and in the predicted file it appears as,

T9 Regulation 36 44 increases
T10 Gene_expression 52 62 expression
E1 Regulation:T9 Theme:E2
E2 Gene_expression:T10 Theme:T2

In this case, as per ASRM evaluation, event though the spans are not exactly correct (34 <= 36, 42 <= 44, 50 <= 52 and 60 <= 62) and the referred event of E2 is Regulation and not the actual Positive_regulation, this is considered as a match to the gold entry.

Once event annotation files are produced from CRF output, we evaluate them using the evaluation scripts provided. The output of the scripts specify the standard measures recall, precision and F-measure for individual events, groups of event together viz. Simple, Binding and Regulation events and overall measures.

Label	Recall	Precision	F-measure
Event	27.87	53.27	36.59

Table 4.2: Stepwise approach-without protein related features

Label	Recall	Precision	F-measure
Event	39.15	62.82	48.24

Table 4.3: Stepwise approach-with protein related features

4.3 Results of Event Extraction

4.3.1 Stepwise Approach

In this approach, we perform event detection and classification in separate steps to see if there is any change in performance. In this case, the problem becomes a binary classification task contrary to the nine-class classification task as in the previous approach. So, we just classify the tokens as event or not-event.

Table 4.2 illustrates the preliminary results using the same set of features mentioned above except without protein related features:

When we added three new features viz. distance from nearest protein, nearest protein and nearest protein PoS, we saw significant change in precision, recall and F-measure values. These are as shown in following Table 4.3:

For event classification and argument extraction, we also need non-trigger words along with trigger words, as some of them act as the Theme for events. So, we concluded that only using triggers detected in first phase for event classification in second phase is not a promising approach. So, we did not investigate this approach further.

4.3.2 Joint Approach

In this approach, we perform both event detection and classification together.

Table 4.4 shows the results obtained using the baseline features mentioned in the previous chapter along with some new features.

Error Analysis - When we analyzed the gold-data of event annotations to understand the reason for low recall, we came across two observations which may have af-

Features	Recall	Precision	F-measure
Baseline	20.46	39.40	26.93
Baseline + unsupervised PoS feature	20.46	39.31	26.91
Baseline + unsupervised POS feature + Dependency relation feature	20.57	39.32	27.01
Baseline + unsupervised POS feature + Dependency relation feature + Distributional thesaurus feature	19.90	39.51	26.47

Table 4.4: Strict span match-with complete dataset

fected the recall:

1. **Multi-theme events:** For many events, even though they are correctly predicted but are unreported due the way event annotations file is being created. For example,

The active nuclear form of the NF-kappa B transcription factor complex is composed of two DNA **binding** subunits, NF-kappa B *p65* and NF-kappa B *p50*, both of which share extensive N-terminal sequence homology with the v-rel oncogene product.

Events in gold data,

*E5 Binding:T32 (**binding**) Theme:T4 (p65)*
*E6 Binding:T32 (**binding**) Theme:T5 (p50)*

and events in the generated event annotations file,

*E5 Binding:T32 (**binding**) Theme:T4 (p65)*

Currently we are only considering the nearest protein as the Theme for a predicted event. Even though it is correct in some cases, but in other cases the nearest protein is not the Theme for an event. And in other cases, as in the example above, along with nearest protein, other proteins are also the Themes for the same event.

2. **Nested (Regulation) events:** Similarly, for many nested events (Regulation events), they were partially predicted correctly but went unreported due to the same reason of event annotations file creation. For example,

The binding of I kappa B/MAD-3 to NF-kappa B *p65* is **sufficient** to **retarget** NF-kappa B *p65* from the nucleus to the cytoplasm.

Features	Recall	Precision	F-measure
Baseline	33.49	39.40	36.20
Baseline + unsupervised POS feature	33.49	39.31	36.17
Baseline + unsupervised POS feature + Dependency relation feature	33.67	39.32	36.27
Baseline + unsupervised POS feature + Dependency relation feature + Distributional thesaurus feature	32.57	39.51	35.71

Table 4.5: Strict span match-without complex events in the dataset

Events in gold data,

*E10 Positive_Regulation:T36 (**sufficient**) Theme:E11*

*E11 Localization:T37 (**retarget**) Theme:T16 (p65)*

and events in the generated event annotations file,

*E10 Positive_Regulation:T36 (**sufficient**) Theme:T16 (p65)*

So here, Theme of E1 which is indirectly T2, is reported, but not through nested event E2.

So, to check how the system performs in case of simple events in the evaluation dataset, we removed the multi-theme events and nested/dependent events from the gold-data for evaluation and again evaluated our system using the similar feature combinations. Results are shown in Table 4.5

As we can see from this Table, there is a significant increase in the recall and F-measure values. This means that the system works well on simple and binding events and suffers on complex regulation events.

Another important observation is that in the results above there was restriction on the perfect match of triggers (strict match mode). So, when we relaxed this constraint and evaluated our system again in approximate span match mode, we observed an increase in recall and F-measure values on both datasets as shown in Table 4.6 and 4.7 (Here we show detailed results for individual events also). The reason behind this increase in performance measure values is that the spans given in the gold-data for proteins and triggers are based on the raw, non-tokenized texts, while the various parser outputs provided and which are used for feature extraction are based on the tokenized texts.

Event Class	Gold (match)	Answer (match)	Recall	Precision	F-measure
Gene_expression	356 (164)	219 (164)	46.07	74.89	57.04
Transcription	82 (28)	48 (28)	34.15	58.33	43.08
Protein_catabolism	21 (14)	17 (14)	66.67	82.35	73.68
Phosphorylation	47 (30)	35 (30)	63.83	85.71	73.17
Localization	53 (21)	25 (21)	39.62	84.00	53.85
=[SVT-TOTAL]=	559 (257)	344 (257)	45.97	74.71	56.92
Binding	312 (73)	102 (73)	23.40	71.57	35.27
==[EVT-TOTAL]==	871 (330)	446 (330)	37.89	73.99	50.11
Regulation	199 (16)	69 (16)	8.04	23.19	11.94
Positive_regulation	717 (64)	294 (64)	8.93	21.77	12.66
Negative_regulation	227 (10)	92 (10)	4.41	10.87	6.27
==[REG-TOTAL]==	1143 (90)	455 (90)	7.87	19.78	11.26
==[ALL-TOTAL]==	2014 (420)	901 (420)	20.85	46.61	28.82

Table 4.6: Approx. span match-with complete dataset

Event Class	Gold (match)	Answer (match)	Recall	Precision	F-measure
Gene_expression	356 (164)	219 (164)	46.07	74.89	57.04
Transcription	82 (28)	48 (28)	34.15	58.33	43.08
Protein_catabolism	21 (14)	17 (14)	66.67	82.35	73.68
Phosphorylation	47 (30)	35 (30)	63.83	85.71	73.17
Localization	53 (21)	25 (21)	39.62	84.00	53.85
=[SVT-TOTAL]=	559 (257)	344 (257)	45.97	74.71	56.92
Binding	312 (73)	102 (73)	23.40	71.57	35.27
==[EVT-TOTAL]==	871 (330)	446 (330)	37.89	73.99	50.11
Regulation	66 (16)	69 (16)	24.24	23.19	23.70
Positive_regulation	191 (64)	294 (64)	33.51	21.77	26.39
Negative_regulation	69 (10)	92 (10)	14.49	10.87	12.42
==[REG-TOTAL]==	326 (90)	455 (90)	27.61	19.78	23.05
==[ALL-TOTAL]==	1197 (420)	901 (420)	35.09	46.61	40.04

Table 4.7: Approx. span match-without complex events in the dataset

And as we are using the tokenized texts for feature extraction, we have to manually maintain the alignment of tokenized text with the raw text which may not be 100% accurate due to few of the tokenization issues mentioned in section 3.6.2. Therefore, we see an increase in recall, precision and F-measure values in ASM evaluation.

MOO experiments

As shown in the tables 4.5 and 4.4, we were getting low recall compared to F-measure values. So, to determine the best combination of features which will maximize F-measure and recall, we used NSGA-II software [KDM02]. It is written in C-Programming

language and it provides an interface where we have to specify the objective functions to be maximized or minimized.

NSGA-II requires various parameters to be specified before the start of MOO viz. initial population, number of generations, probability of crossover and mutation (it is recommended that crossover probability should be very much greater than mutation probability and mutation probability should be very low like reciprocal of number of features as used in [KDM02]). We ran **NSGA-II** with different parameter sets to get the optimal feature subset for **two objective functions viz. maximize F-measure and maximize recall**. The different parameter values and obtained optimal fitness values for objective functions are shown in the Table 4.8. The execution times seem very high for a few runs, however, in general, there are a lot of factors on which it depends like available CPU usage, the classifier algorithm used, i.e. the time to build and test a machine learning model, the time to obtain objective function values etc.

Note: For tables 4.8 and 4.10, the expansions of headers are as follows, Gen - number of generations, pop - initial population, pcross - probability of Crossover, pmut - probability of Mutation, rseed - initial random seed, feats - number of optimal features selected, F - optimal F-measure, R - optimal recall, P - optimal precision, eTime - execution time.

Run	Gen.	pop	pcross	pmut	rseed	feats	F	R	P	eTime (days)
1	1	24	0.89	0.02	0.5	28	26.82	20.29	39.55	2
2	4	48	0.95	0.002	0.8	25	27.95	20.91	42.13	4
3	6	36	0.96	0.001	0.6	23	27.18	20.29	41.16	7
4	8	164	0.92	0.019	0.65	27	**28.42**	**21.35**	**42.49**	18
5	15	52	0.9	0.002	0.7	27	28.13	19.7	49.17	10

Table 4.8: MOO with SMM evaluation

The F-measure and recall values without MOO were 26.47 and 19.90 respectively. So, from the Table 4.8, we can observe that the F-measure and recall values are increased by approximately 2% in Run 4 (F-measure: 28.42, recall: 21.35). Also, the number of features providing best performance are decreased to 27 from 49 in this run. These are as follows:

Lexical features (8 binary features plus the Token itself as a feature), PoS of surrounding tokens (5), Genia dependency features (3), distance from nearest and second nearest protein, suffixes/prefixes, distributional thesaurus word

These features are mainly lexical features, surrounding tokens' features, genia de-

Event Class	Gold (match)	Answer (match)	Recall	Precision	F-measure
Gene_expression	356 (169)	224 (169)	47.47	75.45	58.28
Transcription	82 (25)	41 (25)	30.49	60.98	40.65
Protein_catabolism	21 (16)	19 (16)	76.19	84.21	80.00
Phosphorylation	47 (27)	34 (27)	57.45	79.41	66.67
Localization	53 (23)	28 (23)	43.40	82.14	56.79
=[SVT-TOTAL]=	559 (260)	346 (260)	46.51	75.14	57.46
Binding	312 (74)	95 (74)	23.72	77.89	36.36
==[EVT-TOTAL]==	871 (334)	441 (334)	38.35	75.74	50.91
Regulation	199 (19)	75 (19)	9.55	25.33	13.87
Positive_regulation	717 (57)	253 (57)	7.95	22.53	11.75
Negative_regulation	227 (9)	84 (9)	3.96	10.71	5.79
==[REG-TOTAL]==	1143 (85)	412 (85)	7.44	20.63	10.93
==[ALL-TOTAL]==	2014 (419)	853 (419)	21.35	42.49	28.42

Table 4.9: Strict span match-with complete dataset and optimal feature set

pendency features and one unsupervised feature, i.e. DT word. So, we can conclude that these features are the dominant ones in the performance of bio-medical event extraction. With these best performing feature combination we again ran the event extraction system and observed the results for individual events as shown in Figure 4.9.

After comparing our systems optimal scores 4.9 with the scores of the teams in BioNLP'09 shared tasks as shown in Figure 4.1, we observed that our system ranks 16th compared to all teams, 6th in simple events, 1st in binding events and 13th in Regulation events and 3rd among the teams who made use of CRF.

Also, to check whether MOO selects same set of features if evaluation mode is changed, we ran the experiments 2,3 again with same parameter set but with ASM evaluation. The results are shown in Table 4.10.

Run	Gen.	pop	pcross	pmut	rseed	feats	F	R	P	eTime (days)
2	4	48	0.95	0.002	0.8	**26(25)**	29.46	21.2	48.27	4
3	6	36	0.96	0.001	0.6	**27(23)**	29.88	21.4	49.49	7

Table 4.10: MOO with ASM evaluation

By comparing the number of optimal features selected for Run 2 and 3 from Table 4.8 and 4.10, it can be observed that MOO selects different optimal feature subsets when the evaluation mode is changed.

Team	Simple Event	Binding	Regulation	All
UTurku	**64.21 / 77.45 / 70.21**	**40.06 / 49.82 / 44.41**	**35.63 / 45.87 / 40.11**	46.73 / 58.48 / 51.95
JULIELab	**59.81 / 79.80 / 68.38**	**49.57 / 35.25 / 41.20**	**35.03 / 34.18 / 34.60**	45.82 / 47.52 / 46.66
ConcordU	**49.75 / 81.44 / 61.76**	20.46 / 40.57 / 27.20	**27.47 / 49.89 / 35.43**	34.98 / 61.59 / 44.62
UT+DBCLS	**55.75 / 72.74 / 63.12**	23.05 / 48.19 / 31.19	**26.32 / 41.81 / 32.30**	36.90 / 55.59 / 44.35
VIBGhent	**54.48 / 79.31 / 64.59**	**38.04 / 38.60 / 38.32**	17.36 / 31.61 / 22.41	33.41 / 51.55 / 40.54
UTokyo	45.69 / 72.19 / 55.96	**34.58 / 50.63 / 41.10**	14.22 / 34.26 / 20.09	28.13 / 53.56 / 36.88
UNSW	45.85 / 69.94 / 55.39	23.63 / 37.27 / 28.92	16.58 / 28.27 / 20.90	28.22 / 45.78 / 34.92
UZurich	44.92 / 66.62 / 53.66	30.84 / 37.28 / 33.75	14.82 / 30.21 / 19.89	27.75 / 46.60 / 34.78
ASU+HU+BU	45.09 / 76.80 / 56.82	19.88 / 44.52 / 27.49	05.20 / 33.46 / 09.01	21.62 / 62.21 / 32.09
Cam	39.17 / 76.40 / 51.79	12.68 / 31.88 / 18.14	09.98 / 37.76 / 15.79	21.12 / 56.90 / 30.80
UAntwerp	41.29 / 65.68 / 50.70	12.97 / 31.03 / 18.29	11.07 / 29.85 / 16.15	22.50 / 47.70 / 30.58
UNIMAN	50.00 / 63.21 / 55.83	12.68 / 40.37 / 19.30	04.05 / 16.75 / 06.53	22.06 / 48.61 / 30.35
SCAI	43.74 / 70.73 / 54.05	28.82 / 35.21 / 31.70	12.64 / 16.55 / 14.33	25.96 / 36.26 / 30.26
UAveiro	43.57 / 71.63 / 54.18	13.54 / 34.06 / 19.38	06.29 / 21.05 / 09.69	20.93 / 49.30 / 29.38
Team 24	41.29 / 64.72 / 50.41	22.77 / 35.43 / 27.72	09.38 / 19.23 / 12.61	22.69 / 40.55 / 29.10
USzeged	47.63 / 44.44 / 45.98	15.27 / 25.73 / 19.17	04.17 / 18.21 / 06.79	21.53 / 36.99 / 27.21
NICTA	31.13 / 77.31 / 44.39	16.71 / 29.00 / 21.21	07.80 / 18.12 / 10.91	17.44 / 39.99 / 24.29
CNBMadrid	50.25 / 46.59 / 48.35	33.14 / 20.54 / 25.36	12.22 / 07.99 / 09.67	28.63 / 20.88 / 24.15
CCP-BTMG	28.17 / 87.63 / 42.64	12.68 / 40.00 / 19.26	03.09 / 48.11 / 05.80	13.45 / 71.81 / 22.66
CIPS-ASU	39.68 / 38.60 / 39.13	17.29 / 31.58 / 22.35	11.86 / 08.15 / 09.66	22.78 / 19.03 / 20.74
UMich	52.71 / 25.89 / 34.73	31.70 / 12.61 / 18.05	14.22 / 06.56 / 08.98	30.42 / 14.11 / 19.28
PIKB	26.65 / 75.72 / 39.42	07.20 / 39.68 / 12.20	01.09 / 30.51 / 02.10	11.25 / 66.54 / 19.25
Team 09	27.16 / 43.61 / 33.47	03.17 / 09.82 / 04.79	02.42 / 11.90 / 04.02	11.69 / 31.42 / 17.04
KoreaU	20.56 / 66.39 / 31.40	12.97 / 50.00 / 20.59	00.67 / 37.93 / 01.31	09.40 / 61.65 / 16.31

Figure 4.1: BioNLP'09 team scores for Task 1. Source: [KJDJ09]

4.4 Comparison to Other Approaches

As explained in previous section, our system outperforms the best system at BioNLP'09 in detecting and classifying Binding events and moderately performs well for Simple events but suffers a lot for Complex events. Especially the recall for our system is low compared to the higher precision, the reasons of which (multi-theme and nested events) are also discussed in previous section. The main reason for this was that we were unable to extract multiple candidate themes from CRF output for an event. Thus, we only used the nearest protein as the theme for any event. But, even with this hypothesis, our system did considerably well for Simple and Binding events than other CRF based teams at BioNLP'09.

The best performing system for all events as described in [JBS], makes extensive use of the graph based features and works in three stages viz. trigger recognition, argument detection and semantic post-processing. Event/trigger detection is approached as a named entity recognition task while event classification is approached as trigger-trigger or trigger-protein binary classification task. The best performing system using CRF, was ranked 7th in BioNLP'09 shared task (F-measure 34.92). It used external sources

like WordNet[6] and MetaMap[7] and a rule based approach for argument extraction. The second ranked system in CRF based approach (F-measure 30.35) used MeSH[8] and GO[9] as the external sources. However, this system's overall F-measure value was just 2% (approx.) greater than our system (F-measure 28.42) which kept us at rank three in CRF based systems.

[6] http://www.nltk.org/howto/wordnet.html
[7] http://metamap.nlm.nih.gov/
[8] http://www.ncbi.nlm.nih.gov/mesh
[9] http://geneontology.org/

Chapter 5

Conclusion and Future Work

We tried to address the issues in event extraction and unsupervised lexical acquisition by developing a CRF based event detection and classification system which has an overall F-measure value of 28.42% on complete dataset and 36.79% on simple event dataset using baseline and unsupervised features. The results of our system indicate that we were able to answer the research questions we had proposed at the beginning of this project. For example, we conclude that a few of the orthogonal and Genia dependency features form one of the best lot for event extraction system. Unsupervised feature, DT word, also helped improve the performance of our system as it was included in the best feature combination. Thus, we can conclude that even though to a smaller extent, but unsupervised features do help in bio-text mining. In the end, using MOO we were able to optimize our feature set containing 49 features to the best performing 27 features. This once again proved that NSGA-II plays a dominant role in MOO.

Our hypothesis of using nearest protein as the theme for all events did let our system down for complex events. However, given the limitation of CRF classifier and taking into account the performances of other CRF based systems, we can observe that our system did considerably well for Simple and Complex events of BioNLP'09 shared task.

In terms of F-measure, our system ranks 3rd compared to the teams using CRF based approaches, 6th in Simple events, 1st in Binding events and 13th in Regulation events and 16th compared to all teams that participated in BioNLP'09 shared task. An interesting part of future research would be to identify the relation between event and theme for complex events as these are the events which are difficult to detect and hence make the recall less. Also, testing our system with different classifier algorithms and using other relevant external resources for feature extraction like the other teams may help improve the performance of our system.

Bibliography

[ACaC07] L. Montecchi-Palazzi G. Nardelli M. V. Schneider L. Castagnoli A. Chatr-aryamontri, A. Ceol and G. Cesareni. Mint: the molecular interaction database. *Nucleic Acids Research, Database issue:D572-4*, 2007.

[AES11a] Md. Hasanuzzaman A. Ekbal, A. Majumder and S. Saha. Bio-molecular event extraction using support vector machine. *IEEE-ICoAC*, pages 298–303, 2011.

[AES11b] Md. Hasanuzzaman A. Ekbal, A. Majumder and S. Saha. Supervised machine learning approach for bio-molecular event extraction. *SEMCCO(2) 2011: Lecture Notes in Computer Science, Volume Part II*, pages 231–238, 2011.

[ARAW00] H. F. Chang S. M. Humphrey J. G. Mork S. J. Nelson T. C. Rindflesch A. R. Aronson, O. Bodenreider and W. J. Wilbur. The nlm indexing initiative. pages 17–21, 2000.

[ATP03] M. Pechenizkiy A. Tsymbal, P. Cunningham and S. Puuronen. Search strategies for ensemble feature selection in medical diagnostics. *Proceedings of 16th IEEE Symposium on Computer-Based Medical Systems, New York, USA, pages 124Ű-129*, 2003.

[bdt] *Big Data and Text Mining*. http://www.slideshare.net/MichelBruley/1-text-mining-v0a.

[Bie06] C. Biemann. Unsupervised part-of-speech tagging employing efficient graph clustering. *Proceedings of the COLING/ACL 2006, Student Research Workshop, Sydney, Australia, pages 7Ű-12*, 2006.

[Bie09] C. Biemann. Unsupervised part-of-speech tagging in the large. *Research on Language and Computation*, pages 101–135, 2009.

[BR13] C. Biemann and M. Riedl. Text: now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modeling, Vol 1, No 1, pages 55Ű-95*, 2013.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning, 24(2): pages 123-Ű140*, 1996.

[BT93] D. Bertisimas and J. Tsitsiklis. Simulated annealing. *Statistical Science*, pages 10–15, 1993.

[CBG07] C. Giuliano C. Biemann and A. Gliozzo. Unsupervised part of speech tagging supporting supervised methods. *Proceedings of RANLP-07, Borovets, Bulgaria*, 2007.

[CC10] M. Steedman C. Christodoulopoulos, S. Goldwater. Two decades of unsupervised pos induction: How far have we come? *EMNLP, Massachusetts, USA, pages 575Ű-584*, 2010.

[Cha93] Hendrickson C. Jacobson N. Perkowitz M. Charniak, E. Equations for part-of-speech tagging. *In National Conference on Artificial Intelligence, Washington, D.C., pages 784-Ű789*, 1993.

[Cla03a] A. Clark. Combining distributional and morphological information for part of speech induction. *Proceedings of European chapter of the Association for Computational Linguistics, Budapest, Hungary*, pages 59–66, 2003.

[Cla03b] A. Clark. Combining distributional and morphological information for part of speech induction. *In Proceedings of EACL 2003, Budapest, Hungary, pages 59Ű-66*, 2003.

[CV12] J. Lobry C. Versèle, O. Deblecker. Electric vehicles - modelling and simulations. *intechopen book*, 2012.

[CY06] H. Chen and X. Yao. Evolutionary multiobjective ensemble learning based on bayesian feature selection. *IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada*, pages 267–274, 2006.

[Deb01] K. Deb. Multi-objective optimization using evolutionary algorithms. *John Wiley and Sons, Ltd, England*, 2001.

[ES12] A. Ekbal and S. Saha. Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 143–166, 2012.

[ES13] A. Ekbal and S. Saha. Simulated annealing based classifier ensemble techniques: Application to part of speech tagging. *Information Fusion*, pages 288–300, 2013.

[ET93] B. Efron and R. J. Tibshirani. An introduction to the bootstrap. *Chapman & Hall, London, U.K., ISBN 9780412042317*, 1993.

[Fel98] C. Fellbaum. Wordnet: an electronic lexical database. *Computational Linguistics, Volume 25 Issue 2, pages 292–296*, 1998.

[FL05] Y. Yang F. Li. Analysis of recursive gene selection approaches from microarray data. *Bioinformatics, volume 21, issue 19*, pages 3741–3747, 2005.

[Har09] Z. S. Harris. Methods in structural linguistics. *Wiley Online Library, Volume 54, Issue 3*, 2009.

[Ho98] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 832–844, 1998.

[HR12a] R. Huang and E. Riloff. Bootstrapped training of event extraction classifiers. *In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France*, pages 286–295, 2012.

[HR12b] Ruihong Huang and Ellen Riloff. Modeling textual cohesion for event extraction. *In Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, Ontario, Canada*, 2012.

[JBS] F. Ginter A. Airola T. Pahikkala J. Bjorne, J. Heimonen and T. Salakoski. Extracting complex biological events with rich graph-based feature sets. *Proceedings of BioNLP shared task 2009, Boulder, Colorado*, pages 10–18.

[JDKC04] Y. Tsuruoka Y. Tateisi J. D. Kim, T. Ohta and N. Collier. Introduction to the bio-entity recognition task at jnlpba. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, pages 70–75, 2004.

[JDLP01] A. McCallum J. D. Lafferty and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, pages 282–289, 2001.

[JG08] H. Ji and R. Grishman. Refining event extraction through cross-document inference. *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Columbus, OH, pages 254Ű-262*, 2008.

[JH07] U. Leser J. Hakenberg, M. Schroeder. Consensus pattern alignment to find protein-protein interactions in text. *proceedings of BioCreative-II, pages 213–215*, 2007.

[KDM02] S. Agarwal K. Deb, A. Pratap and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, pages 182–197, 2002.

[KJDJ09] Pyysalo S Kano Y Kim J-D, Ohta T and Tsujii J. Overview of bionlpŰ09 shared task on event extraction. *In Proceedings of the Workshop on BioNLP, Boulder, Colorado*, 2009.

[KM03] Dan Klein and Chris Manning. Optimization, maxent models, and conditional estimation without magic. *HLT-NAACL, Edmonton, Canada*, 2003.

[KS00] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition, volume 33, No 1: pages 25–41*, 2000.

[Le03] C. T. Le. Introductory biostatistics. *Wiley, Hoboken, NJ, DOI: 10.1002/0471308889*, 2003.

[LG10] S. Liao and R. Grishman. Using document level cross-event inference to improve event extraction. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, pages 789–797, 2010.

[LHV07] M. Krallinger L. Hirschman and A. Valencia. Cnio centro nacional de investigaciones oncolt'ogicas. *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 2007.

[Lin98] D. Lin. Automatic retrieval and clustering of similar words. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, pages 768–774, 1998.

[LM99] Y. Liao and J. Moody. Constructing heterogeneous committees using input feature grouping: Application to economic forecasting. *Advances in Neural Information Processing Systems, pages 921-Ű927*, 1999.

[LR12] W. Lu and D. Roth. Automatic event extraction with structured preference modeling. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, pages 835–844*, 2012.

[LSOS03] F. Bortolozzi L. S. Oliveira, R. Sabourin and C. Y. Suen. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17:903–929, 2003.

[LSOS06] M. Morita L. S. Oliveira and R. Sabourin. Feature selection for ensembles using the multi-objective optimization approach. *Studies in Computational Intelligence (SCI), volume 16*, pages 49–74, 2006.

[MA11] A. Jain K S. Reddy P. Kosaraju S. Muktyar B. Ambati R. Sangal M. Agarwal, R. Goutam. Comparative analysis of the performance of crf, hmm and maxent for part-of-speech tagging, chunking and named entity recognition

for a morphologically rich language. *Pacific Association For Computational Linguistics (PACLING2011 2011), Kuala Lumpur, Malaysia, report no:IIIT/TR/2011/92*, 2011.

[Mam03] H. Mamitsuka. Empirical evaluation of ensemble feature subset selection methods for learning from a high-dimensional database in drug design. *Proceedings of Third IEEE Symposium on BioInformatics and BioEngineering, Bethesda, MD, USA, pages 253-Ű257*, 2003.

[MC08] D. McClosky and E. Charniak. Self-training for biomedical parsing. 2008.

[MKV08] C. Rodriguez-Penagos M. Krallinger, F. Leitner and A. Valencia. Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome Biol., 9 (Suppl. 2), S4*, 2008.

[Nav06] A. Navot. On the role of feature selection in machine learning thesis. *Thesis submitted to the Senate of the Hebrew University, pp 145*, 2006.

[Nt'e05] C. Nt'edellec. Learning language in logic-genic interaction extraction challenge. *Proceedings of the 4th Learning Language in Logic Workshop (LLL05), Bonn, Germany, pages 31–37*, 2005.

[Opt99] D. W. Optiz. Feature selection for ensembles. *Proc. of 16th International Conference on Artificial Intelligence, Stockholm, Sweden*, pages 379–384, 1999.

[PFBM92] P. Desouza J. C. Lai P. F. Brown, V. J. Della Pietra and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics, 18(4), pages 467Ű-479*, 1992.

[PP94] J. Kittler P. Pudil, J. Novovicova. Floating search methods in feature selection. *Pattern Recognition Letters*, pages 1119Ű–1125, 1994.

[RGM05] D. Westbrook R. Grishman and A. Meyers. Nyu's english ace 2005 system description. *In Proceedings of the ACE 2005 Evaluation Workshop, Washington*, 2005.

[RS11] M. Andrew R. Sebastian. Fast and robust joint models for biomedical event extraction. *Proceedings of the 2011 Conference on Empirical Methods*

in Natural Language Processing, Edinburgh, Scotland, UK, pages 1-Ű12, 2011.

[Sch99] R. E. Schapire. A brief introduction to boosting. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Stockholm, Sweden*, pages 1401–1406, 1999.

[SP03] F. Sha and F. Pereira. Shallow parsing with conditional random fields. *Proceedings of NAACL*, 2003.

[TMG12] T. Zesch T. Miller, C. Biemann and I. Gurevych. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. *Proceedings of COLINGŰ12, Mumbai, India*, pages 1781Ű–1796, 2012.

[WD03] F. D. Meulder B. Naudts W. Daelemans, V. Hoste. Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. *Proceedings of the 14th European Conference on Machine Learning, Cavtat/Dubrovnik, Croatia*, pages 84–95, 2003.

[WHR07] R. Lynn W. Hersh, A. Cohen and P. Roberts. Trec 2007 genomics track overview. *Proceeding of the Sixteenth Text REtrieval Conference, Gaithersburg, Maryland*, 2007.

[wik] *Dominance and pareto optimality*. http://en.wikipedia.org/wiki/Pareto_efficiency.