Generation of Domain Ontologies from Text

By

Bushra Qadir

NUST201260804MSEECS61312F

Supervisor

Dr. Sharifullah Khan

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science (MSCS)

In

School of Electrical Engineering and Computer Science,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(June 2015)

Approval

It is certified that the contents and form of the thesis entitled "**Generation of Domain Ontologies from Text**" submitted by **Bushra Qadir** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Sharifullah Khan**
Signature: _____

Date: _____

Committee Member 1: **Dr. Jamil Ahmad**

Signature: _____
Date: _____

Committee Member 2: **Dr. Asad Anwar Butt**

Signature: _____
Date: _____

Committee Member 3: **Dr. Muhammad Muneeb Ullah**

Signature: _____
Date: _____

I would like to dedicate this thesis to my loving parents and teachers for their unconditional support and encouragement …

Abstract

Development of high-throughput experimental techniques and computational models have accelerated the pace of research and development in the field of bio-medicine. A large number of genes and proteins are analyzed at a given time, with an aim to obtain new findings about diseases in order to improve human health. This has resulted in an exponential growth in the field of molecular biology. The knowledge of molecular interactions between genes and transcription factors is of huge interest for a biologist. However, most gene interactions are scattered throughout scientific literature, which is written in natural language and difficult to be directly processed with computers. Traditional search engines provide modest help as they return thousands of relevant documents. The user still has to read all those returned documents to get the information they need. It is becoming more and more difficult to discover required knowledge without utilizing information extraction techniques.

The existing approaches that extract gene interactions from bibliographical resources have some limitations that need to be addressed. They are limited to single interaction relations, where a single keyword is used to express relationship between the entities involved. The current relation extraction systems ignore the sentences with multiple interaction keywords. Moreover, they also ignore sentences which contain regulatory information but there is no explicit relationship keyword used in the sentence. This results in extraction errors. In this research work we propose a rule based extraction system that can automatically extract relations between entities such as genes and transcription factors, from biomedical text and present the distilled information in a structured and concise form to users. Our approach uses rules based on regular expressions over annotations to cater the limitations of existing approaches. To validate the proposed methodology, a prototype system has been implemented. The system has been evaluated against a gold standard annotation set and also compared with existing systems. The experimental results show improvement in accuracy, with an average precision of 82.3% and average recall of 89.9%. In future, we intend to incorporate the coreference resolution technique into our system to further improve its accuracy.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Bushra Qadir**

Signature: _____

Table of Contents

List of figures

List of tables

Chapter 1

Introduction

This chapter introduces the research work that has been carried out in this thesis. It includes motivation and problem definition, followed by a discussion of objectives.

1.1 Motivation

Advancements in biological technology, like DNA microarrays have made it possible to analyze a large number of genes and proteins at a given time [1]. The end results of this biological research are reported in textual publications. A large number of new publications and research articles in the field of life science is added to bibliographic databases every year [2]. The volume of data available on the web is growing rapidly. The ever-expanding literature can make it overwhelming for a biologist to find and assemble needed information from this flood of data [3].

For biomedical scientists, the knowledge of molecular interactions between genes and transcription factors (TF) is of huge interest. Transcription factors play significant role in a wide range of human disorders by controlling aberrant gene expression [4]. So a biologist is interested to know which transcriptions are involved in aberrant gene expression and in what manner do they regulate these genes. A transcription factor either activates gene expression or inhibits it. The direction of regulation is also of primary importance for construction of gene regulatory networks [5].

However, this type of information is underutilized by biomedical researchers as the published information is highly unstructured and is written in free-format and also because of overwhelming volume of sources [6]. Several databases have been constructed to store information about molecular interactions and reaction networks, for example KEGG [7]. However most data in these databases were collected manually and in order to synchronise these databases with latest research discoveries, a great deal of resource and labor intensive

maintenance is required [8]. Biomedical researchers continue to publish their research findings, without submitting their results to specific knowledge bases. Consequently most gene interactions are still scattered throughout scientific literature, which is written in natural language and difficult to be directly processed with computers [5].

For biologists interested in specific molecular interactions, the unstructured information in scientific literature poses a big challenge as they have to spend considerable amount of time reviewing papers in order to extract the facts they need [8]. Automation of this task can greatly facilitate and speed up their daily research work. This motivated us to develop a system that provides a more systematic access to gene interactions information hidden in corpora using text mining and information extraction technologies.

1.2 Problem Definition

Several methods have been proposed in the recent years to extract gene interactions from bibliographical resources. However they have some limitations discussed below:

1.2.1 Limited to single interaction keyword

The number of interaction keywords used in the text have an impact on the implied relationship between the entities, i.e. gene and transcription factor. Sometimes a relationship is expressed without explicitly using an interaction keyword. Use of more than one interaction keywords to show the relationship is also common. The current relation extraction systems are limited to single interaction keyword relations [9, 10]. In other words, they only consider cases where a single keyword is used to express relationship between the entities involved. Ignoring the sentences with multiple interaction keywords may result in reduced precision as an additional keyword may cause the extracted relation to be entirely opposite of what was actually intended. Moreover, the systems [1, 11] which assume that potential sentences must have a keyword cannot extract a number of correct relationships, thus resulting in extraction errors.

1.2.2 Trade off between precision and recall

The existing systems face a trade off between precision and recall. Most rule based systems [9, 10] have high precision as this approach is more transparent and thus the required criteria can be easily enforced. However, the existing rule based approaches consider only the cases where the relationship between TF and genes is expressed through a single keyword. As a result, a limited number of relationships are extracted, resulting in low recall. On the other

hand, machine learning based approach [1, 11] tends to achieve high recall but suffers from low precision. Moreover most machine learning systems only recognize the transcription factor contexts in the literature [12] or retrieve the set of relevant documents with regulatory content [13]. The type of relationship remains unidentified in the ML based methods. Using a rule based approach with added multi-keyword and no-keyword support in addition to named entity disambiguation will significantly increase extraction precision as well as recall.

1.3 Research Goal

The overall goal of the work described in this thesis is '*to build a system that can automatically extract relationship between TFs and their target genes*'. Given a set of known transcription factors and genes, the system should answer the following two questions:

1. Which transcription factors modulate the expression of which genes?
2. What is the type of interaction i.e., inhibition, activation or underspecified regulation?

The objectives of the methodology are:

- **Multi-keyword support**: As discussed before, if the system can deduce correct relationship when the number of relationship keywords is more than one, then accuracy of the system can be improved.
- **No-keyword support**: Similar to multi-keyword support, the system should also be able to infer relationship if there is no explicit relationship keyword used in the sentence.
- **Improved precision and recall**: The current systems face a trade-off between precision and recall. They either have high precision at the cost of recall or have high recall at the cost of precision. To improve both precision as well as recall while achieving a balance between the two measures is one of the objectives of our research.

1.4 Proposed System

Our proposed system has three main modules; (i) Pre-processing, (ii) Entity Recognition (iii) Triple Extraction. In the first module, the documents are preprocessed to convert raw text into a well organized sequence of linguistically-meaningful and machine understandable units. Then in entity recognition module, we use dictionary based approach to identify the entities like genes, transcription factors and relationship keywords. Finally in the triple extraction

module, we use rules based on regular expressions to extract regulatory triples from the given documents.

We have evaluated our system using three datasets. Two different evaluation approaches have been used. In the first approach, the system is evaluated in terms of target gene detection. In the second approach, the system is evaluated in terms of extraction of gene regulatory triples. Experimental results demonstrate an improvement in accuracy, with an average precision of 82.3%, recall of 89.9% and f-measure of 85.9%.

1.5 Thesis Outline

The rest of the document is organized as follows: Chapter 2 presents a background to some of the biological terms used in this thesis and defines terminology from the domains of information extraction and natural language processing. Chapter 3 discusses various related approaches, along with their critical analysis. In chapter 4, we give a detailed description of the proposed system methodology, explaining in detail the process of extraction of triples from unstructured text. Chapter 5 gives a complete overview of implementation details and describes the experimental results and a comparison with the existing systems. Concluding remarks and future work are presented in chapter 6.

Chapter 2

Background

This chapter briefly explains some terms that are used throughout this thesis. Being interdisciplinary in nature, our study uses terminology from different domains such as genetics, information extraction, linguistics and natural language processing. So here we will briefly touch upon these domains, in particular the following fundamental areas:

1. Biological background
2. Natural Language Processing (NLP)
3. Information Extraction (IE)
4. Triples (i.e. subject–predicate–object) expressions

The organization of above areas is in accordance with the way they lay the foundation of the research study. Firstly, as the problem is from a biological domain, it is important to have a biological background, with emphasis on basic concepts like transcription and translation, genes and proteins. Next, how information extraction can be applied to biological literature and how techniques from natural language processing help in information extraction are discussed. These areas are explained below:

2.1 Biological Background

Every organism, including human beings is composed of one or more cells, that form the basic building blocks of life. The instructional manual of a cell lies in its Deoxyribonucleic acid (DNA), which is a blueprint that "encodes the genetic instructions used in the development and functioning of all known living organisms" [14]. DNA is made up of chain of nitrogenous bases. From its inception till death, every phase of life of a living organism is controlled

and determined by its DNA. Its is so important for two reasons. Firstly, it is a medium for transferring hereditary information from parents to offspring. Secondly, it dictates the production of proteins. In this section we discuss how genetic information flows in a biological system- well known in the literature as *"central dogma of molecular biology*" and described as "DNA makes RNA and RNA makes protein"[15].

2.1.1 Genes

The number of base pairs (or bits of genetic information) in human genome approximates to 3 billion and that encodes roughly 22,000 genes. Genes are segments of DNA that code for an operative gene product, which are often proteins, however there also some non-protein coding genes that code for functional RNA.

The mechanism by which information within a gene is read and utilized for the synthesis of a functional gene product is called gene expression [16]. There are two main steps involved in gene expression: 1) Transcription and 2) Translation.

1. Transcription

DNA is located inside the nucleus of a cell, whereas proteins are synthesized outside the nucleus, in the cytoplasm. As DNA cannot leave the nucleus and move outside into the cytoplasm for protein synthesis, a medium is needed that is similar to DNA and carries genetic information from DNA, away from the nucleus inside the cytoplasm. This medium is served by a special type of RNA, known as messenger RNA (mRNA). This process by which information from DNA is copied into mRNA by the enzyme RNA polymerase, is termed as "transcription".

2. Translation

In the second step of gene expression, a ribosome decodes the information in mRNA to direct protein synthesis. In this step, the sequence of bases in mRNA are read by ribosome. Proteins are made up of polypeptide chains, the building blocks of these chains are amino acids. In a mRNA, a series of three nitrogenous bases code for one amino acid. A sequence of amino acids is assembled by transfer RNA (tRNA), one at a time, to generate a polypeptide chain which folds into a protein later on. This process also includes steps to process the protein post-translationally.

This central dogma of molecular biology is summarized in the Figure 2.1

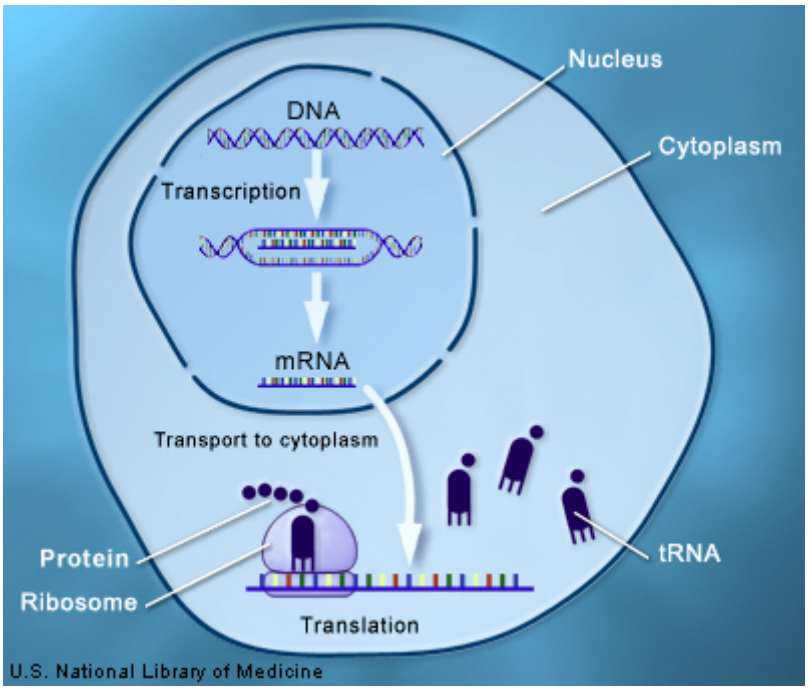

Fig. 2.1 Information flow from gene to protein
[17]

2.1.2 Gene Regulation

Not all of the genes in a cell are expressed at a given time. Only a fraction of genes are expressed while others are turned off. The mechanism which controls the rate of expression of genes is termed as 'gene regulation'. Gene regulation is a complex procedure and depends on a number of factors such as interactions between genes and special proteins, cellular signals and RNA molecules [18].

One of the most striking results of human genome project was that a considerably large fraction of the genome actually does not code for any product. About 97% of genes are non-coding and only 3% is coding. So it led to explore what is the actual purpose of non-coding DNA. It turned out that these stretches of DNA that do not encode a protein, in fact have a very significant role in controlling whether a gene is turned on or off [19] . Such sequences are called "regulatory sequences".

RNA Polymerase II

The process of transcription is catalyzed by a complex enzyme termed as "RNA polymerase II". It binds to DNA, decodes it, makes the mRNA and ultimately protein is synthesized. Although this enzyme is fairly complex in structure with multiple subunits, however it cannot

attach on its own to the DNA. It cannot distinguish between coding and non-coding regions of DNA. An external help is required by RNA Polymerase II to find out where to start reading the appropriate gene and begin transcription. This leads to the conclusion that there are other factors that assist RNA Polymerase II land into the appropriate place at the appropriate time in the genome of each cell, so that the cell can make right proteins and functions normally. These specialized factors are called "transcription factors" and are described below.

2.1.3 Transcription factors

Transcription is a highly regulated process.Transcription factors are key molecules that regulate the use of genetic information that has been encoded in the genome.They recognize the vast majority of the non-coding genome and finally interact with these little bits of genetic information to turn genes on or off [19]. They are synthesized within the cytoplasm, like other proteins. When the cell receives an external stimulus that signals the need for certain proteins encoded by its DNA, these factors migrate into the nucleus where they interact with the specific gene and regulate transcription. Transcription factors are gene specific, that is, they only recognize the particular gene whose expression they regulate.

These control regions are discussed below [18]:

1. Start site. A start site where transcription begins. RNA polymerase binds to this region and starts making RNA transcript.

2. A promoter. General transcription factors bind to promoter region of gene and facilitate in binding of RNA polymerase to the start site. This region is a few hundred nucleotides upstream of the gene. It is not read or decoded, but instead has a role in controlling gene transcription.

3. Enhancers. Once RNA polymerase has bound to the promoter region, transcription begins, when it is activated by a transcription factor. These type of transcription factors are called "activators". Enhancer is a regulatory sequence, at a distance of about thousands of nucleotides from promoter region. Activators bind to these enhancers and coordinate with mediator molecules to increase the rate of transcription.

4. Silencers. Another category of transcription factor is called repressors. They bind to regions known as "silencers" that slow down the rate of transcription. Repressors can also inhibit transcription by binding to the place where activators bind to DNA. Thus they block activators from DNA attachment and depress RNA activation. Some repressors interfere between molecular interactions between activators and RNA polymerase.

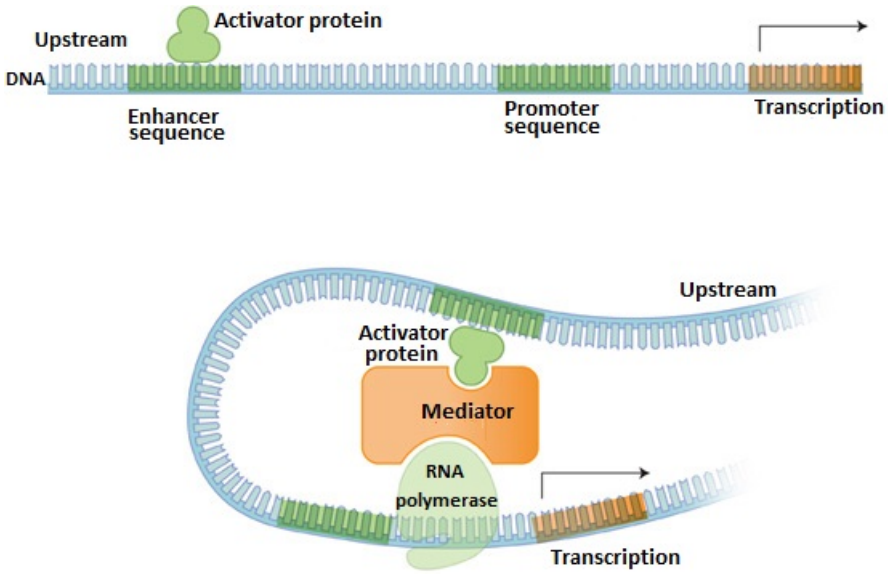

Fig. 2.2 Transcriptional activation and control sequences [20]

Activators and repressors can also play their part by alternating the chromatin alignment. Repressors coil up the chromatin tightly, making the DNA unavailable for transcription. On the other hand, activators can uncoil an already coiled DNA segment to release it for transcription. All these factors collectively play role in the temporal and spatial regulation of gene transcription.

Figure 2.2 shows an activator protein that binds to enhancer region and produces a shift in chromatin alignment. Here the changed alignment promotes RNA polymerase and transcription factor binding, resulting in an transcriptional activation of the gene.

This was a brief introduction to some of the basic biological processes. Next we discuss why transcription regulation is a hot topic in biomedical research and how it impacts human health and drug development.

2.1.4 Why study transcription regulation?

A large number of basic biological phenomena such as cell differentiation, embryonic development and cell fate make use of transcription factors. Every phase of life, including cellular functions, functioning of tissues, organism survival and reproduction almost all major aspects are dependent on the process of gene expression, and transcription is the first step in this process.

Another reason is that understanding the fundamental molecular interactions that regulate transcription, in humans or in any other organism, can help scientists discover the cause

of diseases. For example a few diseases that could be studied as a result of understanding the function and structure of transcription factor are cancer, diabetes, infectious agents, inflammation, Huntington's disease, Parkinson's disease and so forth. It is hoped that understanding the molecular and functional underpinnings of these complex diseases will enable us to develop more specific, targeted therapeutic drugs and also to design more effective and rapid diagnostic tools. Several diseases are associated with mutations in transcription factors. So these are a couple of the reasons why study of transcriptional regulation has received so much attention and many of biologists have spent years studying this process of transcriptional regulation [19].

2.2 Natural language processing

Natural language processing (NLP) [21] is a sub field of artificial intelligence, computational linguistics and computer science. Natural language processing aims at developing techniques that enable computers derive meaning from natural language. The main challenge of natural language processing is to transform the language that is human understandable into a format that is machine understandable. Natural language is the key means of communication that humans use. A set of symbols that convey a meaningful thought when arranged in a structured way forms a basic element of language. There are two concepts associated with a language statement, syntax and semantics. The structural rules that a statement should follow in order to make sense, are collectively known as syntax. Semantics deals with what is meant by a sentence and how it is interpreted. In the rest of this section, we describe how syntactic and semantic analysis is exploited to achieve the goals of NLP.

Words and tokens

Words can be characterized in many ways. The field of linguistics defines words as *"symbols for concepts"*, where symbol is the string used to denote a concept- a real world object. Just as word is the unit of spoken language, the notion of *token* represents the unit of written text, and the process of segmenting a stream of text into tokens is called *tokenization*.

The concept of token is different from that of a word. Tokens include punctuation characters, while words do not. Tokens are bordered by white spaces or line break [22]. For example, the sentence

> *"Additionally, both SLUG and SNAIL repressed endogenous E-cadherin expression."*

has twelve tokens but nine words.

Lexical relations

A number of associations exist between lexical units of a language, that characterize how semantics of two words relate to each other. Two or more words may symbolize the same concept, conversely one symbol may represent distinct concepts. Words may have opposite semantics. The most common lexical relations are defined as under [23]:

Synonymy is the relation that exist between words that share the concept they symbolize. Interchanging one for the other does not alter the meaning of the context e.g. the terms "inhibit" and "repress" are related by synonymy.

Homonymy exists where one word can represent distinct concepts. This is very common in biomedical text where the symbol is used both for gene and protein representation e.g "RUNX1" is both a protein name as well as a name of a gene that codes for the *protein* RUNX1.

Meronymy relation exist between two objects when one of them is a constituent part of the other. As an example, the ATF-2/JunD heterodimer is made of two proteins, ATF-2 and JunD. So we can say that ATF-2 is a meronym of ATF-2/JunD dimer.

Hyponymy: Hyponymy and hypernymy relations exist between a word and its generalization or specialization. The meaning of hyponym is completely contained within its hypernym. For example, the word "regulate" is a more generalized term for word "inhibit". So we say that semantically, the word *inhibit* is a hyponym of the word *regulate.*

Antonymy: Two words with opposite meaning are said to be antonyms of each other, e.g inhibit vs activate.

WordNet [24] is a highly structured lexical database that maintains the synonymous, meronymous and hyponymous relations between nouns, verbs and adjectives.

Part-of-speech

A part-of-speech (POS) of a word is a description of what role does the word play in the syntactic structure of a sentence. The most common part-of-speech categories are noun, verb and adjective. A word may have different part-of-speech based on the context in which it is used. The process of assigning POS to a word is called part-of-speech tagging. The algorithms used for POS tagging can be categorized into two types 1) Rule based that use a set of rules for POS tagging. 2) Based on stochastic methods that use probabilistic models [25].

POS tagging has a strong tie with corpus linguistics. Initially Brown Corpus was used in studies for POS tagging algorithms, but since 2005, it was replaced with a much larger corpus- the British National Corpus that consists of around 100 million words [25]. Some of the techniques used for part-of-speech tagging are as follows:

1. Hidden Markov Model: In Hidden Markov Model (HMM) based POS tagging, a corpus such as Brown Corpus is used to count the occurrence of certain sequences of POS classes, and then assigning probabilities to each. For example, counting the corpus for cases of an article followed by a noun gives 40% probability, probability of an article followed by adjective is 40% whereas that of article with a number next to it is 20%. Given this probability distribution, the algorithm can guess that the word *can* in 'the can' has more probability of being classified as noun, rather than a verb [25]. A more sophisticated HMM would be that considers triple of sequences, instead of pairs. For example a noun with a verb next to it will have very limited chance of having another verb next in the sequence. The process continues by enumerating every possibility and then multiplying probability of each to determine the probability of entire sequence. An implementation of this HMM model is CLAWS and has accuracy in the range of 93-95%.

2. Unsupervised taggers: These taggers do not use a pre-existing corpus to learn probabilities from. Using bootstrapping, they train with an untagged corpus and learn tags by induction.

3. Rule based tagger: Rule based tagger use a set of rules to tag part of speech to the words in the input text. They are different from stochastic based approaches in that they do not need to store extra information in the form of probabilistic values. A commonly used rule based tagger is the Hepple Tagger [26], which is a variation of the Brill tagger [27]. Brill tagger is a supervised tagger that initially assigns the most frequent tag to a word, without considering its context. The tag assigned in the initial step is provisional and may be incorrect as it does not consider the context of the word. The taggers then improves its performance in an incremental fashion in the next steps of the algorithm. The next steps then use contextual rules in an iterative manner to correct the initial tags. These iterations continue to be applied until no further rules are applicable or a threshold value has been achieved.

2.3 Information Extraction (IE)

Information extraction is the task of extracting structured information from the content of large text collections. In IE, the *facts* are analyzed. This is different from information retrieval (IR) which uses specific keywords or queries to pull documents from large text collections, such as, the Web. In IR, the *documents* are analyzed. Traditional query engines, are designed for retrieving whole documents. So getting the facts can be hard with traditional query engines. IR provides documents with the relevant information somewhere, whereas IE returns structured information at a much deeper level than traditional IR. If a database is constructed through IE and linked back to the documents, then it can provide a valuable alternative search tool. Results may not be always accurate, but they can be valuable if linked back to the original text.

2.3.1 Applications using IE

IE is an enabling technology for many other applications:

1. Text Mining
2. Opinion Mining
3. Decision Support
4. Question Answering
5. Rich information retrieval and exploration
6. Semantic Annotation

2.3.2 Types of IE systems

There are two main types of IE systems: Knowledge engineering and learning systems [28]. Table 2.1 differentiates between the two.

2.3.3 Named Entity (NE) recognition

Traditionally, named entity recognition is the identification of proper names in texts, and their classification into a set of predefined categories of interest, for example, person, location, organisation, gene, protein, transcription factor, disease etc.

Table 2.1 Types of IE systems

Knowledge Engineering	Learning Systems
rule based	use statistics or other machine learning techniques
developed by experienced language engineers make use of human intuition	do not need language engineering expertise
require only small amount of training data	require large amounts of annotated training data
development can be time consuming	
changes are easier to accommodate	some changes may require re-annotation of the entire training corpus

NE provides a foundation from which more complex information extraction systems can be built. Relations between named entities can provide tracking, ontological information and scenario building.

2.4 Triples

A triple is a "data entity composed of subject-predicate-object" [29]. In biomedical domain, interactions are mostly represented in the form triples where the subject and object are the biological entities, such as, genes or proteins and the predicate denotes the relationship between the subject and the object. For example, the statement "Activation of IL-4 transcription by NFAT1" can be expressed in a concise and structured form as a triple "NFAT1 activates IL-4". Here NFAT1 is the subject, IL-4 is the object and the predicate is the activation relationship between these two entities.

Chapter 3

Literature Review

Over the past decade, many approaches have been proposed that exploit text mining techniques to automatically extract gene regulatory relationships from biomedical literature. Biomedical text is inherently complex and therefore most of the relationship extraction systems work at the sentence level. The focus of this chapter is to present an overview of the existing techniques that have been used for the extraction of transcriptional regulation. Based on the techniques involved in the extraction method from text, the existing approaches can be broadly categorized into four groups:

1. Co-occurrence approach
2. Pattern based approach
3. Machine learning based approach
4. Hybrid approach

These techniques are discussed in detail in the following sections.

3.1 Co-occurrence Approach

This is the simplest approach which is based on the hypothesis that if two entities co-occur in the same sentence, abstract or document, they are somehow related to each other [30, 31]. In this approach it is quite possible that two entities co-occur in the same literature without having any relationship. Therefore most systems use frequency based ranking to omit relations that occurred by chance [32].

3.2 Rule based Approach

Rule based (also known as pattern based) approach is one of the most popular and traditional ways of identifying relationship between genes and transcription factors in text. The rules are developed heuristically through linguistic analysis and are specified using regular expressions. These rules are then implemented as finite state automata. Here, we are going to discuss some of the systems that use patterns for describing gene regulation relationships.

The earliest gene regulation relationship extraction systems date back as far as 2004 [9]. The authors in [9] have used syntacto-semantic rules to extract relational information from biological abstracts. Using baker's yeast as the model organism, they extract regulatory network for yeast. They have used syntacto-semantic chunking to recognize named entities and relation chunks from the corpus. The rules were implemented as CASS grammar. They have specified the following three conditions that must be fulfilled to extract a regulation information from a sentence:

1. The sentence must mention gene expression.
2. The identity of regulator must be known.
3. The identity of target must be known.

Moreover their rules do not consider the relationships that are reported in literature as a result of genetic modifications. Their rules ignore the relationships that occur after the genes/proteins have been artificially introduced through genetic engineering.

The rule based approach we discussed previously was extended and improved in another system, STRING-IE [10] in which they made subsequent changes to their proposed system discussed earlier. The changes are listed below:

1. Slight improvement in recall by capturing linguistic structures missed in the previous studies.
2. Extending the rule set to cover interactions other than gene expressions such as, phosphorylation and de-phosphorylation.
3. Application of the system to four model organisms.

Although they have expanded their rules and applied the system to more model organisms, their conditions for relationship extraction in this system were the same as in their previous study.

The rule based system in STRING-IE was further extended by RodrÃguez-Penagos et al. [3]. They modified the core grammars for the original parser to incorporate 'verbal

phrase coordination' and anaphoric relationships that deal with cases where a pronoun is used instead of actual gene/protein chunk. Chunked-parsed sentences were transformed into an XML file with a format they name as 'Regulatory Network Mining Markup Language (or RNM^{2}L)'. Their system is customized for E. coli.

To reduce the manual effort required in construction of hand crafted rules, machine learning based techniques have been proposed that extract sentences with gene regulation information from the literature.

3.3 Machine Learning (ML) based Approach

Machine learning systems approach the problem of extracting transcriptional regulation contexts as a binary classification task. Given a sentence, they classify it as positive if it contains regulation information or negative otherwise. They are trained automatically on manually annotated corpora and can be easily adapted to changes in the identification tasks.

The system proposed by Yang et al. [12] identifies transcription factor contexts in literature using machine learning techniques. The major components of their system are:

1. Selection of relevant features to support classification: They have analyzed two types of features.

   (a) A standard bag-of-words approach for the generic model. They used Pearson's chi-square test to rank the words in the descending order of their likelihood of distinguishing the class. A sentence vector is built by using the features above the threshold for all words that are present in it.

   (b) For the biological model the following features are identified in candidate sentences: gene/protein names, interaction words, TF-related MeSH and GO terms, and other biological words. They used the publicly available taggers: ABNER and LingPipe to recognize gene/protein names.

2. Selection of ML approaches to be employed for context recognition: The feature vectors are used in three different machine learning algorithm: Naive Bayes, Support Vector Machine and Maximum Entropy to learn the classifiers.

In another study [13] Aerts et al. used vector space model to identify and prioritize relevant documents likely to have high cis-regulatory content. They used similarity based ranking to prioritize documents for curation purpose. In addition to this, they extracted DNA sequences from full text articles and mapped the DNA sequences to genome sequences in order to

identify the species, location and target gene information for annotation of cis-regulatory networks.

To make the best use of both machine learning approach and pattern based approach, some hybrid methods have been proposed. The hybrid approach combines both the aforementioned approaches in an attempt to achieve better results.

3.4 Hybrid Approach

Both machine learning and pattern based approaches have their own pros and cons. The ML-based approach is more generalized than the specifically tuned rule based approach. However in real-life scenario, rule based systems prove to be more robust than ML-based systems [33]. This led the researchers to design methods that use a combination of both machine learning and rule based approach to achieve the benefits of both methods.

AutoPat [1] is one of those systems that uses the hybrid approach. The aim is to develop a semi-supervised pattern generation module and use the patterns to extract sentences containing gene expression relationships. A gene expression pattern is assumed to contain at least one transcription factor, one gene and one key verb. The system has two major components:

1. *Pattern Generation Module* that uses supervised patterns to search and build a large pattern set. The training corpus is based on the abstracts with titles containing the seed patterns. If a pattern in the training corpus also matches to one of the established pattern templates, this pattern is selected.

2. *Interaction Extraction Module* in which the pattern set is used to extract related regulation sentences from literature. The sentences are ranked based on a number of features such as pattern matching score, distance between transcription factor and target gene, the position feature and the type of pattern template. The combined weight of a sentence is calculated as the sum of each feature weight.

Another study [11] takes advantage of bootstrapping to automatically generate patterns. Their proposed methods consists of three phases:

1. *Abstract retrieval phase*: The user specifies a TF query and abstracts related to that TF are collected from PubMed.

2. *Pattern training phase*: From the retrieved abstract, sentences containing a '(TF-TGene)-tuple' are selected and patterns are found based on a set of initial 'seed' tuples.

3. *Bootstrapping phase*: Patterns found in phase 2 are used to search other articles to extract new '(TF-TGene)-tuples'. The process continues until no new tuples can be found.

In the next section we would critically analyze all the aforementioned approaches one-by-one.

3.5 Critical Analysis

In the previous sections, we discussed various approaches to extract relationship between a transcription factor and its target genes. The section will critically analyze each technique discussing its strengths as well as limitations.

1. Co-occurrence approach suffers from low precision although they have a high recall. The underlying base assumption that two entities are related if they occur together, tends to generate a lot of false positives. Sentences in biomedical text are inherently complex. It is common that a sentence mentions multiple entities but only a fraction of them are actually related to each other. Another limitation is that the type of relationship is not extracted, making them unsuitable for applications that require extracted relations [30, 31].

2. Rule based approach [9, 10, 3] has high precision as it is more transparent and thus the required criteria can be easily enforced. It is easy to incorporate domain knowledge into the rules, making the approach more flexible. In addition to it, the cause of errors can easily traced compared to machine learning approach and thus errors are easy to fix which explains the high precision. However they are tedious and time consuming. The existing rule based approaches consider only the cases where the relationship between TF and genes is expressed through a single keyword. As a result, a limited number of relationships are extracted, resulting in low recall.

3. Machine learning based approaches [12, 13] tend to achieve high recall but suffer from low precision, just as the co-occurrence method. Moreover they only recognize the transcription factor contexts in the literature [12] or retrieve the set of relevant documents with regulatory content [13]. The type of relationship remains unidentified in the ML based methods. In a comparative study [33], ML-based approach for gene regulatory information extraction was compared with rule based approach. The results showed that under real-life conditions the rule based approach performs better than the ML based system. A lot of information in biomedical documents can only be captured in rules and the labelled data is relatively scarce for this kind of data as annotation of such highly specific documents requires specialized labour.

4. Hybrid approach [1, 11] achieves better recall than manually defined patterns. However the precision is reduced due to noisy patterns. A large number of patterns are generated automatically. Some generated patterns are so specific that they cannot match any unseen text, while other are overly generic that they match any text causing a large number of false positives. Furthermore, these system also lack at determining the type of relation between participating entities i.e, genes and TFs.

In a nutshell we can conclude that the current gene regulation extraction systems [30, 31, 12, 13, 1, 11] do not take into account an important aspect of transcriptional regulation i.e., the type of relationship between a TF and its target gene. Moreover they also suffer from low precision. The systems [9, 10, 3] have high precision but consider only a limited number of possible ways in which a relationship can be described in text. Thus they have low recall values. There is a need to define a technique that can extract TFs, target genes as well as the type of relationship between them and also has improved precision and recall values.

Summary

In this chapter an overview of existing approaches to extract gene regulatory information has been presented. The approaches that have been discussed are: co-occurrence approach, rule based approach, machine learning approach and hybrid approach. An overview of these approaches with respect to their working methodology has been discussed. In the end, a critical analysis of these techniques has been presented highlighting the drawbacks and strengths of these approaches.

Chapter 4

Proposed System Design

This chapter provides details of the the proposed system. The system consists of three main modules. (i) Pre-processing, (ii) Entity Recognition (iii) Triple Extraction. Each module will be discussed in detail in the remaining sections. The system was developed using GATE developer [34], an open source Java based IDE used for text processing. Figure 4.1 shows the architecture of the proposed system. In this architecture a corpus of biological documents is given as an input to the system. Each document is pre-processed before the entities are recognized. In entity recognition module, lists of genes, transcription factors and their relations are provided to identify the entities in documents. Disambiguation rules and extraction rules are provided to triple extraction module to produce the regulatory triples from the given documents.

4.1 Pre-processing module

Pre-processing is a key part of any Natural Language Processing (NLP) application. Pre-processing converts human understandable raw text into a well organized sequence of linguistically-meaningful and machine understandable units [35]. It identifies the fundamental units in the text, such as characters, words, sentences and morphemes. These fundamental units are then passed further to subsequent processing stages. The pre-processing module of our system includes (i) Document Resetter, (ii) Tokeniser, (iii) Sentence Splitter, (iv) POS tagger and (v) Morphological Analyser.

4.1.1 Document Resetter

A document resetter is required if a document has some already annotated contents. So before a document is processed by the subsequent modules, such as, Named Entity Recognition

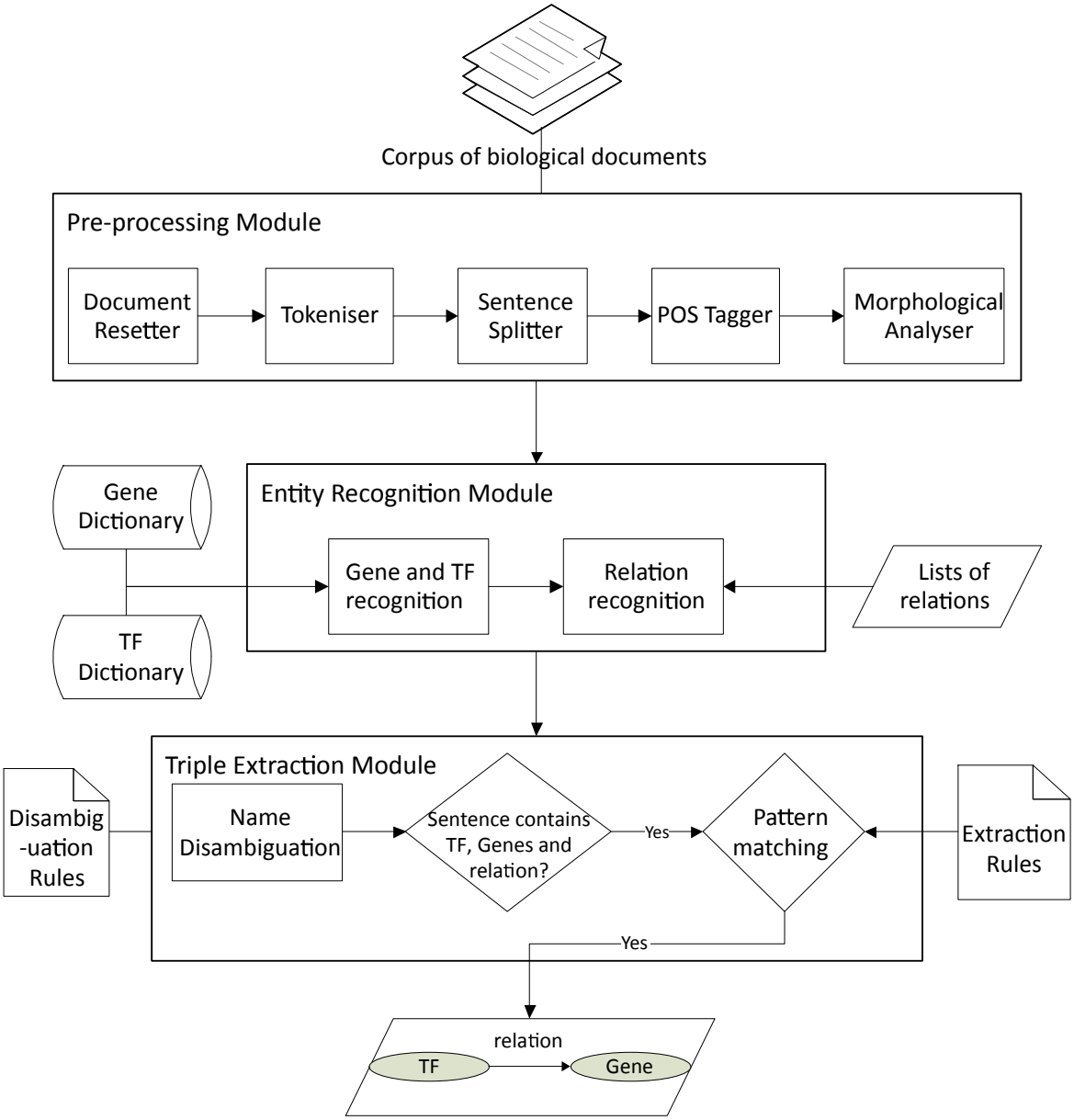

Fig. 4.1 System Architecture

(NER) and triple extraction, it is essential to remove previous annotations and to set the document to its original state. Resetting is important for two reasons. Firstly, it removes prior annotations that may mess up with rules in subsequent modules. Secondly, it avoids duplicate annotations when application is run over the same document more than once. Document resetter restores the original state of a document by eliminating all the previous annotations except for the ones that store the document format information, such as paragraph, title and body.

4.1.2 Tokeniser

Electronic text is a sequence of characters. Before any syntactic analysis of the corpus is carried out, text has to be segmented into linguistically significant units such as words, symbols, numbers and punctuation. This process is referred to as tokenisation [36]. Any type of analysis or extraction is not possible unless these basic units, i.e. tokens are clearly segregated. Errors made at this stage will propagate into later phases and cause problems. Therefore it is very important to accurately tokenise the text in the documents.

In this step, word boundaries are located. These word boundaries mark the beginning and ending of each word in text. Most often, word boundaries are easily indicated by white spaces preceding and following any sequence of characters. This rule is applicable to words that consist of alphabets only, but it does not consider punctuation characters. Use of punctuation marks, such as hyphens, commas, colon and periods, create tokenisation ambiguity. The same punctuation mark can be used for different purposes in a single sentence, as illustrated by the following example sentence.

> "After infection of MDA-MB-468 (48 h), selection was initiated in 0.5 mg/ml Geneticin (Invitrogen Corp.)."

This sentence uses periods in three ways: as decimal point (0.5), to mark abbreviation (Corp.) and to end the sentence. This problem is particularly challenging in the bio-medical text domain. In bio-medical literature words containing parentheses, hyphens, and abbreviations are used frequently. Usually these special characters are treated as separate tokens. But hyphenated words are an exception to this general rule and cause problems. For example the phrase "MT2-MMP-dependent" should ideally be tokenised into two tokens 'MT2-MMP' and 'dependent', but since hyphenated words are considered as a single unit, the whole phrase is tokenised as one token. In our system, we have used GATE's ANNIE English Tokeniser [34] to produce token annotations in input documents. The tokeniser outputs token annotations in the documents. Each token annotation has a length and string attribute that holds the value of length and string of the token respectively. These token annotations are

prerequisite for the next submodule, i.e. sentence splitting. Table 4.1 shows the detail of different types of token produced by the tokeniser.

Table 4.1 Types of token

S.No	Type	Description
1	Word	Any sequence of consecutive letters without white space. It may not have any punctuation, except for hyphen.
2	Number	Any sequence of contiguous digits.
3	Symbol	Any number of consecutive symbols like $,Â£, &
4	Punctuation	This includes any type of punctuation like ’,”,:
5	Space	All white spaces are classified into two types, space and control, depending on whether they are entered as space character or a control character.

4.1.3 Sentence Splitter

Since our triple extraction module reads only those sentences which contain genes, transcription factors and relations, it is important to accurately determine sentence boundaries. A sentence splitter segments an input text into individual sentences. Though it sounds easy, identification of sentence boundaries is a complex problem [37]. Most often, punctuation that delimits sentences is also used in abbreviations or ordinal numbers which causes ambiguity. The exclamation point and the question mark are almost always unambiguous. But the period is an extremely ambiguous punctuation mark. It is not trivial to decide when it is a full-stop, a part of an abbreviation, or both. This is illustrated in the the following example sentence, where the period following ’Ecad(108)/EboxB’ seems to be a full-stop, but is actually a part of the biological term term ’Ecad(108)/EboxB. MUT-Luc’.

> “Reporter gene activity of the Ecad(108)/EboxB. MUT-Luc construct relative to that of the Ecad(108)-Luc construct was assessed in a panel of breast cancer cell lines.”

Therefore the problem of sentence splitting reduces to disambiguating all instances of punctuation characters that may be used to end a sentence. In order to disambiguate different uses of punctuation mark, the context in which the punctuation occurs is important. In most cases the character immediately following the punctuation mark is sufficient to determine whether it is a sentence delimiting character or not. For example, a period followed by one or more spaces followed by a capitalized word; marks the boundary of a sentence, provided the

word preceding the period is not an abbreviation. So contextual factors like case distinctions, word length before and after the period, are used to assist in sentence splitting. A sentence splitter takes a tokenised document and adds “Sentence” annotation which spans the entire length of the sentence.

4.1.4 Part-of-Speech Tagger

Identification of the part of speech of a word is an important precursor to any information extraction task. The patterns used to extract regulatory triples frequently make references to POS tags. Relations between a transcription factor and gene are described in the text with specific ’words’. These words can appear with different part-of-speech (POS) such as verb, noun, adjective, adverb etc. This POS of the relational words determine the order in which the entities of our interest,i.e., genes, transcription factors and relational words appear in a sentence. Therefore POS information for each token is required by the relation extraction module. We have used Hepple Tagger [26] for POS tagging of our biological documents. Table 4.2 shows the output of POS tagging of an example sentence:

"Overexpression of CIZ upregulates the transcriptions from MMP-1 promoter"

The tokens of the sentence are listed in the first column, whereas the corresponding tags in the second and their descriptions in the third column.

Table 4.2 POS tagging a sample sentence

Token	Tag	Description
Overexpression	NN	noun-singular
of	IN	preposition or subordinating conjunction
CIZ	NNP	proper Noun- singular
upregulates	VBZ	verb- 3rd person singular present
the	DT	determiner
transcriptions	NNS	noun-plural
from	IN	preposition or subordinating conjunction
MMP-1	NNP	proper noun- singular
promoter	NN	noun-singular

4.1.5 Morphological Analyser

This is the last step in our pre-processing module, and is required for the relation recognition step. Based on the structure and formation of words, a morphological analyser identifies

morphemes in each word. A morphological analyser takes tokens and their part-of-speech tags as input and generates a root and affix of each word. A morphological analyser simplifies the task of relation recognition as its groups all the morphologically related forms of a verb down to a single root value. The words like upregulates, upregulated, upregulating can be matched with a single entry in the relation list: *upregulate*. Table 4.3 shows the root values generated from tokens of the following example sentence by a morphological analyser.

"Overexpression of CIZ upregulates the transcriptions from MMP-1 promoter."

Table 4.3 Root identification from tokens

S.No.	Token	Root
1	Overexpression	overexpression
2	of	of
3	CIZ	ciz
4	upregulates	*upregulate*
5	the	the
6	transcriptions	transcription
7	from	from
8	MMP-1	mmp
9	promoter	promoter

4.2 Entity Recognition Module

This phase identifies name of entities, such as genes, transcription factors and relationship keywords in the input text. Accurate recognition of gene and transcription factors' names is of crucial importance in our work as it leads to accurate relationship extraction between them. Therefore emphasis has been laid on identification of these entities as accurately as possible. The correct identification of genes and transcription factors in the input text is a challenging task because with the passage of time the gene and protein nomenclature has considerably evolved, which resulted in multiple names and synonym for a single entity [38]. Although several communities have provided gene/protein nomenclature paradigms, the naming of newly identified genes and proteins is not strictly according to the naming standards. Researchers are free to define names for genes and proteins in bio-medical literature [39]. Thus, homonyms and synonyms are frequent in gene and protein nomenclature which creates ambiguity. In addition to this, many gene and protein names overlap with general English terms e.g., CAN, LIGHT, FACT. Several tools have been developed and

documented in the literature which are tuned for specific entity recognition in biomedical texts. We have explored some of these tools including Abner [40], PennBioTagger [41] and Genia Tagger [42], but they do not produce desired results and are not applicable to our system's requirements. Therefore, for our named entity identification task, we decided to compile gene and transcription factor dictionaries from multiple sources and use a dictionary based approach along with some rule based disambiguation to accurately recognise genes and transcription factors in documents. In following sections, we discuss the compilation of our gene and transcription factor dictionaries.

4.2.1 Compilation of Gene Dictionary

Several freely accessible databases organize information about genes and proteins, e.g., NCBI [43] and UniProt [44]. These databases can be exploited for deriving dictionaries for named entity recognition. We have downloaded the latest updated NCBI database [43] to compile our gene dictionary.

For each gene, the database has several fields for gene ID, symbol, synonyms, chromosome information, description, type of gene, full name from nomenclature and other designations. Table 4.4 shows a section of the gene information file.

The "Symbol" and "Description" fields contain single entries and thus they were easily parsed automatically. However, the fields listing "Synonyms" and "Other designations" were difficult to parse as these contained special characters and nested parenthesis to indicate additional information like sub family and sub type etc. For example, other designations for gene BRF1, in the last row of Table 4.4 are listed in the database as follows.

> "B - related factor 1|BRF1 homolog, subunit of RNA polymerase III transcription initiation factor IIIB|TATA box binding protein (TBP)-associated factor 3C|TATA box binding protein (TBP)-associated factor, RNA polymerase III, GTF3B subunit 2|TBP - associated factor, RNA polymerase III, 90kD|epididymis secretory sperm binding protein Li 76p"

As it can be seen in above example, there are many special characters like pipe symbol (|), commas, hyphens, nested parenthesis. This information was cleaned programmatically, using AWK [45] to convert it into a uniform and machine understandable dictionary format. The following steps are performed to convert the information within the database into machine understandable gazetteers: the words 'lists' and 'gazetteers' are used interchangeably in this work here.

1. The first step is to select relevant fields from the NCBI database. Gazetteer list compilation is done using AWK- a language for processing text based data.

Table 4.4 A section of NCBI Gene Information file

Symbol	Synonyms	Description	Type of gene	Symbol from nomen-clature	Full name from nomenclature	Other designations
A2M	A2MD|CPAMD5|FWP007|S863-7	alpha-2-macroglobulin	protein-coding	A2M	alpha-2-macroglobulin	C3 and PZP-like alpha-2-macroglobulin domain-containing protein 5|alpha-2-M
A2MP1	A2MP	alpha-2-macroglobulin pseudogene 1	pseudo	A2MP1	alpha-2-macroglobulin pseudogene 1	-
BRF1	BRF|BRF-1|GTF3B|HEL-S-76p|TAF3B2|TAF3C|TAFIII90|TF3B90|TFIIIB90|hBRF	BRF1, RNA polymerase III transcription initiation factor 90 kDa subunit	protein-coding	BRF1	BRF1, RNA polymerase III transcription initiation factor 90 kDa subunit	B - related factor 1|BRF1 homolog, subunit of RNA polymerase III transcription initiation factor IIIB|TATA box binding protein (TBP)-associated factor 3C|TATA box binding protein (TBP)-associated factor, RNA polymerase III, GTF3B subunit 2|TBP - associated factor, RNA polymerase III, 90kD|epididymis secretory sperm binding protein Li 76p

2. Two text files are maintained, referred to as “Symbols” and “Names”. The ’Symbol’ dictionary includes all the single word acronyms, whereas the ’Name’ dictionary contains longer descriptions of each gene.

3. The data is manipulated programmatically so that each pipe (|) delimited symbol or name appears on a new line in the final dictionary.

4. Fields which do not contain any information, such as the ones with only a hyphen (as seen in Table 4.4), are removed.

5. Special characters used in the formatting of original information file are removed, such as double quotes, leading and trailing white spaces.

6. Single-word synonyms are expanded with a leading ’h’ (e.g. hSMRP). The leading ’h ’ before a gene symbol signifies that the gene is found in human beings. The new symbol is included along with the original one, only if it is unique in the gazetteer.

7. The original file had several repetitions, like fields with ’Symbol’ and ’Symbol from nomenclature’ contained exactly similar values (Table 4.4). Therefore once the gazetteer is constructed, all the duplicate entries are removed.

8. The gazetteers are run against the corpus in the case-insensitive manner.

Using these gazetteers, a string match procedure is used for detecting gene names and symbols in the text. Tokens in the text that are matched by the gazetteers are then annotated as “Genes”.

Table 4.5 presents the number of unique gene names, aliases as well as symbols in the final gazetteers.

Table 4.5 Gene Gazetteer Statistics

Organism	**Unique gene names**	**Gene names with aliases**	**Gene symbols with synonyms**
Homo sapiens	47816	107978	103012

4.2.2 Compilation of Transcription Factor Dictionary

The next step is to build a gazetteer of all the proteins in human body that function as transcription factors. For this purpose, we integrated this information from different sources, as listed below:

- 1987 transcription factors were included from the study published in Nature [46] which lists human transcription factors along with their Ensembl IDs and tissue specificity details.
- TFClass [47] is a comprehensive database classifying transcription factors in human genome according to their DNA binding domain. So far they have identified nine super classes with 40 classes and 111 families. Altogether 1558 human TFs have been taken from this source.
- Animal Transcription Factor Database : AnimalTFDB [48] lists a total of 1544 human transcription factors classified into 71 families. These transcription factors were also incorporated in our TF dictionary.
- To make sure that our dictionaries are complete and include aliases and synonyms, protein identifiers gathered from above sources were submitted to UniProt [44] - a comprehensive central repository of information on proteins. In this way the synonyms and aliases of transcription factors are retrieved.

As with the gazetteer of genes, AWK commands were used to transform the list of TF names and symbols integrated from above mentioned sources into machine understandable gazetteer format. We made sure that each entry appears on a separate line and that there are no special characters like quotes, white spaces, hyphens etc. After that all the duplicate entries were removed. The number of transcription factors in the final dictionaries are shown in Table 4.6

Table 4.6 TF Gazetteer Statistics

Organism	TF names with aliases	TF symbols with synonyms
Homo sapiens	6681	5842

4.2.3 Relation Identification

Once the genes and transcription factors were identified, the next task was to determine how they are related to each other. A transcription factor can modulate the expression of a gene by either activating it or suppressing its expression [5]. Based on this fact, the relationship words have been classified into three semantic categories:

Regulation words

This category includes relationship words that show a regulation relationship between a transcription factor and gene but does not show whether the gene is up-regulated or down-

regulated by it. Transcription factors bind to specific DNA regions in order to modulate the expression of a given gene [5]. Therefore, when it is stated that a particular transcription factor binds to a gene, it can be inferred that this gene is regulated by this transcription factor [5]. Examples of words in this category include: regulate, modulation, control, binding, affect, mediate, dependent etc. Tokens in the text which are matched with this list of words; are annotated as 'keywords' with type 'regulation.'

Activation words

This category includes words that show a gene is activated by a transcription factor, for example stimulate, up-regulation, promote, induction, transactivate, augment. Tokens in the text which are matched with this list of words are annotated as 'keywords' with type 'activation.'

Repression words

All those words which show an inhibitory relation between a gene and transcription factor are included in this category, such as deregulate, inhibit, prevent, silence, abrogate, attenuate. Words matched with this list are annotated as 'keywords' with type 'inhibition.'

Three lists are maintained, one for each category. Our proposed lists include a total of 132 words, excluding their inflectional variants. The inflectional variants of these words were matched by using a morphological analyser as discussed in section 4.1.5.

4.3 Triple Extraction Module

This is the final module in our extraction system. As an input it takes a POS tagged corpus labelled with named entities i.e., (i) genes, (ii) transcription factors and (iii) relationship keywords. The output of this module is a set of triples. The triples have three components, transcription factor, gene and relationship which indicates how a gene is related to a transcription factor. This module performs the main extraction task by reading all those sentences in the corpus which contain in a specific pattern of genes, transcription factors and one or more relationship words.

However, before we apply triple extraction rules, it is important to remove the ambiguities that may exist in the named entities because ambiguous entities will lead to incorrect triple extraction. These ambiguities and the disambiguation procedure is explained below:

4.3.1 Named Entity Disambiguation

Determining whether a given string represents a gene name or a protein is a challenging task because gene nomenclature and protein nomenclature are aspects of the same whole. Any symbol or name that represents a gene can also be potentially used for a protein that is encoded by that gene, and vice versa [49]. There is a lot of overlap in the representation of the two entities. However this is not always the case. When genes and their corresponding proteins were discovered in different time frames, their names did not match [49]. Another ambiguity lies within the name/symbol itself when a named entity contains as substring another entity. For example rel and ICE(rel)III are two different genes. However, with our gazetteer list for genes, ICE(rel)III will be annotated twice, one with gene symbol 'ICE(rel)III' and second with 'rel'. This pattern is also quite common in gene/protein nomenclature where a gene or protein name contains as a substring another gene/protein name. These types of ambiguities are removed with rules described in the following sections:

Disambiguation of Gene Name Entities

Certain gene disambiguation rules are applied before applying the triple extraction rules. These are explained as follows:

- Overlapping gene annotations: As said before, there are cases when a gene annotation contains another gene annotation as a substring. The rule in this case is to retain only the annotation with largest span. For example, the string 'Plasminogen activator inhibitor-1' is annotated thrice by the gazetteers, 'Plasminogen',' inhibitor-1' and 'Plasminogen activator inhibitor-1'. The rule here is to retain the annotation with the longest string. According to this rule, annotations with string 'inhibitor-1' and 'Plasminogen' are removed from the gene annotation set as the annotation with the longest span is "Plasminogen activator inhibitor-1". The process is shown in Figure 4.2

- Sometimes gene symbols coincide with abbreviations used in other contexts. For example, RT is a gene symbol but RT-PCR is an abbreviation for Reverse transcription polymerase chain reaction. To remove such disambiguities, a gene annotation immediately followed by a hyphen or number is removed from the gene annotation set. The same rule holds if a gene is followed by a number, e.g SP6 is a gene, but SP600125 is not.

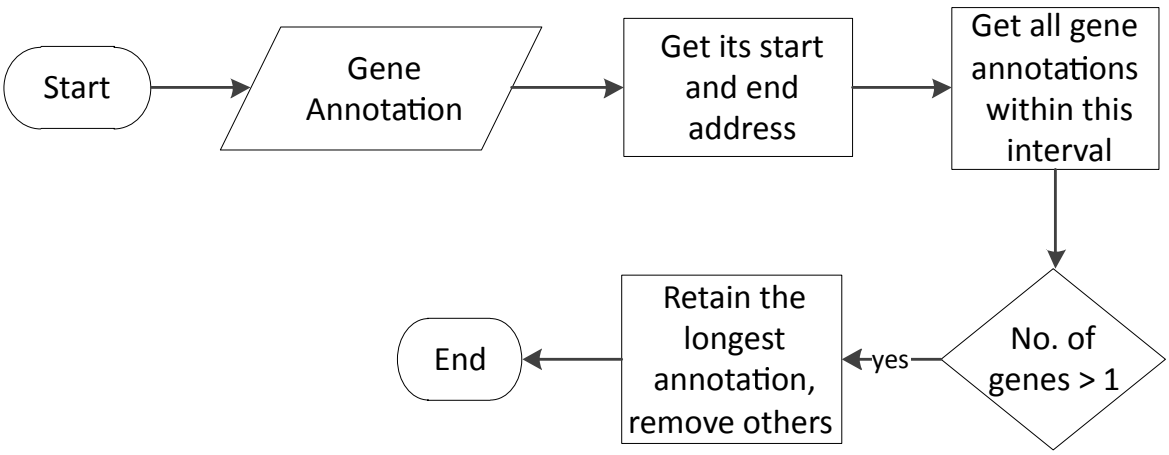

Fig. 4.2 Overlapping Genes Disambiguation

- Similar to the above rule, a gene annotation immediately preceded by a hyphen or number is also removed. A string Cotler-Fox does not represent a gene so according to the rule, Fox gene annotation is removed from string 'Cotler-Fox'.
- Gene annotation immediately followed by keyword cell or cells is removed. Consider a gene symbol NPC (an alias for Niemann-Pick disease, type C1 gene). However in 'NPC cell line', NPC refers to Nasopharyngeal carcinoma. So where NPC is followed by word cell/cells, it is removed from the gene annotation set.

Disambiguation of Transcription Factor Names

Rules for transcription factor name disambiguation are similar to those for disambiguating gene name entities.

- Overlapping transcription factor annotations: As with genes, the string with the longest span is selected as the transcription factor annotation.
- Transcription factor annotation immediately followed by a hyphen or number is removed.
- Transcription factor annotation immediately preceded by a hyphen or number is removed.
- Transcription factor annotation immediately followed by keyword cell or cells is dropped.

The following two rules prevent a gene entity from being mistagged as a transcription factor.

- Transcription factor annotation followed by keywords such as, promoter, region, sequence, is removed as this is an indication of gene entity.

- Transcription factor annotation preceded by keywords such as, promoter, region, sequence, is dropped as this is again an indication of gene entity.

Disambiguation of relationship words

Although very few disambiguities can arise in relationship words, compared to those in genes and transcription factors, it is still important to remove them to accurately determine the type of regulation. In this case, preference is given to the type which contains more specific information. As an example, consider the word 'up-regulation'. This word is annotated by the gazetteers twice, once with string 'regulation' and type 'regulation' and second with string 'up-regulation' and type 'activation'. According to rule, the first annotation is removed while second one is retained because it is more specific and informative.

After the named entities have been disambiguated, the next task is to detect relation between these entities. The following section describes rules that are used to extract regulatory triples from text of input documents.

4.3.2 Triple Extraction Rules

In order to extract gene regulatory information from a text segment, we have developed a set of context-sensitive regular expression rules. These rules are in the form: *Left-Hand-Side (LHS) –> Right-Hand-Side (RHS)*. The LHS of the rules specifies the pattern template for the named entities and the RHS contains information about triple annotation to be created if a text segment matches pattern on LHS. The rules are divided into a series of phases. Each phase contains a set of context-sensitive regular expression rules. These phases constitute a set of finite state transducers, arranged in cascades where the output of one transducer is the input for succeeding one.

The output of this module is a triple, with three attributes; gene, transcription factor and their relation. The values for gene and transcription factor attributes on the RHS are always the same as their corresponding annotations on LHS of the rule. The relation attribute can have one of three values, inhibition, activation or regulation, depending on the number of relationship keyword annotations on LHS of rule.

From a number of various documents, we observed the patterns in which different authors express the relationship between genes and transcription factors. We inspected the words used and the sequence of the words used in gene regulatory text. From the observed patterns,

we realized that in general the gene regulatory relation is expressed in one of the following ways:

- Using a single keyword: This is the most commonly used pattern.
- Using more than one keywords: Often two or even three keywords are used to express the relationship.
- Using no keywords: Sometimes the relationship is expressed without using any keyword explicitly. As we will show in following section.

Based on the observed patterns,we classified the extraction rules into following three categories:

1. One keyword rules
2. Multiple keyword rules
3. Rules that cover special cases

The following sections illustrate these categories. We have used the following conventions to describe patterns:

1. The whole pattern is enclosed in square brackets.
2. 'TF', 'Gene' and 'keyword' denote the corresponding named entity annotations, i.e., transcription factor, gene and relationship keywords respectively.
3. A keyword's POS tag is represented in parenthesis next to it.
4. A dot symbol followed by Kleene star (.*) denotes a possibly empty set of words.
5. The pipe symbol | denotes disjunction.
6. The question mark makes the preceding token in the pattern optional.

Application of each rule is illustrated by an example sentence. For each example, there are two tables.

- First table shows how the text matches to the pattern. It has three columns; the text segments that match entities in pattern is listed in first column, while the matched entities are shown in the second column. The third column shows POS tag of each text segment.
- The second table shows the extracted triple from the example sentence.

One keyword rules

These types of rules deal with sentences which contain only a single keyword that relates a gene and transcription factor. The type of regulation in this case can be easily inferred as there is only one keyword whose type attribute determines the value of relation attribute for the extracted triple. Triple annotations are generated if the sentence has gene, transcription factor and keywords in the following patterns:

Rule 1 Pattern: [TF.* keyword (Verb).* Gene]

This pattern is read as 'a transcription factor followed by zero or more number of words, succeeded by a relationship keyword whose POS tag is a verb. The keyword is followed by zero or more number of words, succeeded by a gene annotation'.

Example:

> "Hypoxia-inducible nuclear factors bind to an enhancer element located 3' to the human erythropoietin gene."

Table 4.7 Rule 1 illustrated

Text	Entity	POS tag
Hypoxia-inducible nuclear factors	TF	noun
bind	Keyword, type=regulation	verb
erythropoietin	Gene	noun

Extracted Triple

TF	relation	Gene
Hypoxia-inducible nuclear factors	regulate	erythropoietin

Rule 2 Pattern: [TF.* keyword(Noun).* Gene]

This is similar to above rule; the entities are in the same order. But in this rule the keyword is in the form of noun.

Example:

> "NF-KB has been previously identified as a regulator of CD86"

Table 4.8 Rule 2 illustrated

Text	Entity	POS tag
NF-KB	Transcription Factor	noun
regulator	Keyword, type=regulates	noun
CD86	Gene	noun

Extracted Triple

TF	**relation**	**Gene**
NF-KB	regulates	CD86

Rule 3 Pattern: [Keyword(Noun) Preposition Gene Preposition.* TF]

The first entity is a noun keyword, succeeded by a prepositional token, followed by a gene annotation, and this gene annotation is followed by another prepositional token. The last entity in the pattern is a transcription factor. Example:

"Activation of IL-4 transcription by NFAT1"

Table 4.9 Rule 3 illustrated

Text	Entity	POS tag
Activation	Keyword, type=activate	noun
of		preposition
IL-4	Gene	noun
by		preposition
NFAT1	Transcription Factor	noun

Extracted Triple

TF	**relation**	**Gene**
NFAT1	activates	IL-4

Rule 4 Pattern: [Gene.* Keyword(Noun|Verb) Preposition.* TF]

A gene entity followed by a noun or verb keyword that is succeeded by a prepositional token. Transcription factor is the last entity.

Examples:

"Cyclin D1 activation through ATF-2"

"transferrin expression is mediated by hypoxia-inducible factor-1"

Table 4.10 Rule 4 illustrated

Text	Entity	POS tag
Cyclin D1	Gene	noun
activation	Keyword, type=activate	noun
through		preposition
ATF-2	Transcription Factor	noun

Extracted Triple

TF	**relation**	**Gene**
ATF-2	activates	Cyclin D1

Multiple keywords rules

Most often, text that shows gene regulations has more than one relationship keywords. This is quite different from the case with a single keyword in the expression as the type of regulation now depends on the combined effect of *all* the keywords mentioned. Considering only one of them and ignoring others may cause the determined direction of regulation to be exact opposite of the actual one. As an example, consider the following sentence

> "Suppression of CREMα expression in SLE T cells results in increased production of IL-2".

Now here if we only consider the substring "CREMα expression in SLE T cells results in increased production of IL-2", it may lead to the wrong conclusion that CREMα activates gene IL-2. Whereas in reality, if seen as a whole, it is indirectly implied that CREMα is a down regulator of IL-2 gene. Because if the transcription factor CREMα is suppressed, then IL-2 production is increased. Such cases are quite frequent in biomedical texts, where two or even three keywords are used to describe how a transcription factor regulates a gene. Hence this category of rules deals with situations where multiple keywords express the regulatory relationship.

Rule 5 to 8 contain two keywords in the L.H.S pattern, arranged in different order w.r.t the transcription factors and genes. Let the first keyword be denoted as keyword-1 and the second be keyword-2, then type of relation in the extracted triple, for rules 5 -9 will be according to Table 4.11

Table 4.11 Relation determination in two-keyword patterns

Type of keyword-1	Type of keyword-2	Extracted relation
activation	activation	activation
activation	inhibition	inhibition
inhibition	inhibition	activation
inhibition	activation	inhibition
activation	regulation	regulation
inhibition	regulation	inhibition
regulation	regulation	regulation

Rule 5 Pattern: [TF .*Keyword-1(adjective) Keyword-2(Noun).* Gene]

The first keyword is POS tagged as adjective while the other one is a noun.

Example:

"CREMα is a negative regulator of IL-2"

Table 4.12 Rule 5 illustrated

Text	Entity	POS tag
Cremα	Transcription Factor	noun
negative	Keyword-1, type=inhibition	adjective
regulator	Keyword-2, type=regulation	noun
IL-2	Gene	noun

According to Table 4.11, if keyword-1 is has type 'inhibition' and keyword-2 is of type 'regulation', then the type of final relation extracted will be 'inhibition'.

Extracted Triple

TF	relation	Gene
Cremα	inhibits	IL-2

Rule 6 Pattern: [TF .*Keyword-1(Verb|Noun).* Keyword-2(Verb|Noun).* Gene]

The two keywords are positioned between transcription factor and gene.

Example 1:

"AML1/ETO inhibition was detected in the downregulation of the VEGFA"

Table 4.13 Rule 6 Example 1

Text	Entity	POS tag
AML1/ETO	Transcription Factor	noun
inhibition	Keyword-1, type=inhibition	noun
downregulation	Keyword-2, type=inhibition	noun
VEGFA	Transcription Factor	noun

Extracted Triple

TF	relation	Gene
AML1/ETO	activates	VEGFA

Example 2

"SOX-5 knockdown can up-regulate SPARC"

Table 4.14 Rule 6 Example 2

Text	Entity	POS tag
SOX-5	Transcription Factor	noun
knockdown	Keyword-1, type=inhibition	noun
up-regulate	Keyword-2, type=activation	verb
SPARC	Gene	noun

Extracted Triple

TF	relation	Gene
SOX-5	inhibits	SPARC

Rule 7 Pattern: [Keyword-1.*TF.* Keyword-2(Verb|Adjective).* Gene]

Example1:

"antisense CREMα resulted in increased promoter activity of CD86"

Table 4.15 Rule 7 Example 1

Text	Entity	POS tag
antisense	Keyword-1, type=inhibition	adjective
CREMα	Transcription Factor	noun
increased	Keyword-2, type=activation	adjective
CD86	Gene	noun

Extracted Triple

TF	relation	Gene
CREMα	inhibits	CD86

Example 2:

"induced Snail1 reduced cortactin"

Table 4.16 Rule 7 Example 2

Text	Entity	POS tag
induced	Keyword-1, type=activation	adjective
Snail1	Transcription Factor	noun
reduced	Keyword-2, type=inhibition	verb
cortactin	Gene	noun

Extracted Triple

TF	relation	Gene
Snail1	inhibits	cortactin

Rule 8 Pattern: [TF.*Keyword-1.* Gene.* Keyword-2]

The first entity is a TF, followed by first keyword. while the gene entity is followed by the second keyword.

Example:

"RUNX1 regulates human CD34 expression"

Table 4.17 Rule 8 illustrated

Text	Entity	POS tag
RUNX1	Transcription Factor	noun
regulates	Keyword-1, type=regulation	verb
CD34	Gene	noun
expression	Keyword-2, type=regulation	noun

Extracted Triple

TF	relation	Gene
RUNX1	regulates	CD34

Rule 9 to 12 contain three keywords in the L.H.S pattern, arranged in different order w.r.t the transcription factors and genes. Let the first keyword in the pattern be denoted as keyword-1,the second be keyword-2 and the third one be keyword-3, then the type of relation in the extracted triples, for rules 9 -12 will be according to Table 4.18

Table 4.18 Relation determination in three-keyword patterns

Type of keyword-1	Type of keyword-2	Type of keyword-3	Extracted relation
activation	activation|regulation	inhibition	inhibition
activation	activation|regulation	activation|regulation	activation
activation	inhibition	inhibition	activation
activation	inhibition	activation|regulation	inhibition
inhibition	activation|regulation	inhibition	activation
inhibition	activation|regulation	activation|regulation	inhibition
inhibition	inhibition	inhibition	inhibition
inhibition	inhibition	activation|regulation	activation
regulation	activation	activation|regulation	activation
regulation	activation	inhibition	inhibition
regulation	inhibition	inhibition	activation
regulation	inhibition	activation|regulation	inhibition
regulation	regulation	inhibition|activation|regulation	regulation

Rule 9 Pattern: [Keyword-1.*TF.*Keyword-2.*Keyword-3.*Gene]

In this rule pattern, the first keyword appears before transcription factor while the other two keywords are positioned between transcription factor and gene entity.

Example:

"Activation of HIF-1 or HIF-2 reduces the expression of E-cadherin"

Table 4.19 Rule 9 illustrated

Text	Entity	POS tag
Activation	Keyword-1, type=activation	noun
HIF-1 or HIF-2	Transcription Factor	noun
reduces	Keyword-2, type=inhibition	verb
expression	Keyword-3, type=regulation	noun
E-cadherin	Gene	noun

Extracted Triple

TF	relation	Gene
HIF-1	inhibits	E-cadherin
HIF-2	inhibits	E-cadherin

Rule 10 Pattern: [TF.*Keyword-1.*Keyword-2.*Keyword-3.*Gene]

Example:

"NFAT1-/- lymphocytes display increased expression of certain cyclin genes"

Table 4.20 Rule 10 illustrated

Text	Entity	POS tag
NFAT1	Transcription Factor	noun
-/-	Keyword-1, type=inhibition	symbol
increased	Keyword-2, type=activation	verb
expression	Keyword-3, type=regulation	noun
cyclin	Gene	noun

Extracted Triple

TF	relation	Gene
NFAT1	inhibits	cyclin

Rule 11 Pattern: [Keyword-1.*Gene.*Keyword-2(verb).*Keyword-3.*TF]

Example:

"Down-regulation of the cyclin A promoter in differentiating human embryonal carcinoma cells is mediated by depletion of ATF-1 and ATF-2"

Table 4.21 Rule 11 illustrated

Text	Entity	POS tag
Down-regulation	Keyword-1, type=inhibition	noun
cyclin A	Gene	noun
mediated	Keyword-2, type=regulation	verb
depletion	Keyword-3, type=inhibition	noun
ATF-1 and ATF-2	Transcription Factor	noun

Extracted Triple

TF	relation	Gene
ATF-1	activates	cyclin A
ATF-2	activates	cyclin A

Rule 12 Pattern: [Keyword-1.*Keyword-2.*Gene.*Keyword-3.*TF]

Example:

"increased expression of cyclin genes in lymphocytes lacking the NFAT1 protein"

Table 4.22 Rule 12 illustrated

Text	Entity	POS tag
increased	Keyword-1, type=activation	adjective
expression	Keyword-2, type=regulation	noun
cyclin	Gene	noun
lacking	Keyword-3, type=inhibition	verb
NFAT1	Transcription Factor	noun

Extracted Triple

TF	relation	Gene
NFAT1	inhibits	cyclin

Rules that cover special cases

This category deals with special cases. For example in sentences where no keyword is explicitly mentioned, but the statement does contain a regulation relationship or when keyword is there but it does not fit into patterns discussed before.

Rule 13a Pattern: [Gene.*synonym(require).*TF]

Here the word 'require' shows that a gene needs a TF for its expression, implying a regulatory relation between the two entities. "synonym(require)" denotes that the word 'require' in the pattern can be replaced with any word that is synonymous to it, e.g. need, necessitate, involve etc. This rule covers the active voice of the sentence. The passive voice is covered in the following rule:

Rule 13b Pattern: [TF.*synonym(required)(VBN).*Preposition.*Gene]

Here the verb is POS tagged as 'VBN' which represents past participle form of the verb. When this verb is followed by a preposition, this ensures the sentences is in passive voice.

Example:

"Smad2 was required for Snail1 expression"

Table 4.23 Rule 13b illustrated

Text	Entity	POS tag
Smad2	Transcription Factor	noun
required		VBN
for		preposition
Snail1	Gene	noun

Extracted Triple

TF	relation	Gene
Smad2	regulates	Snail1

Rule 14a Pattern: [TF.*Token(string="target")(Verb).*Gene]

Targeting of a gene by a TF implies that the gene is regulated by the TF [50]. The above pattern is read as ' a TF followed by a token which has string "target" and POS tag as verb, succeeded by a gene annotation'. The word 'target' in this rule is used as a verb. It can be also be used as noun, as shown in the following rule:

Rule 14b Pattern: [Gene.*Token(string="target")(Noun).*TF]

Example: "CDK4 is a target of JunD"

Table 4.24 Rule 14b illustrated

Text	Entity	POS tag
CDK4	Gene	noun
target		noun
JunD	TF	noun

Extracted Triple

TF	relation	Gene
JunD	regulates	CDK4

One of the most important mechanisms by which gene expression is controlled is the binding of TF to DNA promoter regions located upstream of gene transcription start site [51]. So if a gene's region contains a TF binding site, it can be inferred that the gene is regulated by that TF. The following rule denotes this pattern:

Rule 15a Pattern: [Gene.*synonym(sequence).*synonym(contain).*TF.*Token(string="site")?]

The words that are used synonymously with 'sequence' are promoter, regulatory element and region. Similarly the words used synonymously with 'contain' are recruit, hold, carry, has etc.

Example:

"cyclin A2 promoter contains an NFAT binding site"

Table 4.25 Rule 15a illustrated

Text	Entity	POS tag
cyclin A2	Gene	noun
contains		verb
NFAT	TF	noun
site		noun

Extracted Triple

TF	relation	Gene
NFAT	regulates	cyclin A2

The above pattern can have a keyword at the end or at the beginning. In this case, the type of keyword determines the final relation between TF and gene. This leads further to the following two rules:

Rule 15b Pattern: [Gene.*synonym(sequence).*synonym(contain).*TF.*Token(string="site")? .*Keyword]

Example:

> "The cyclin A2 promoter contains a functional NFAT responsive negative regulatory element."

Table 4.26 Rule 15b illustrated

Text	Entity	POS tag
cyclin A2	Gene	noun
contains		verb
NFAT	TF	noun
negative	Keyword, type=inhibit	adjective

Extracted Triple

TF	relation	Gene
NFAT	inhibits	cyclin A2

Rule 15c Pattern: [Keyword.*Gene.*synonym(sequence).*synonym(contain).*TF.*Token(string= "site")?]

Example:

> "Negative regulatory region at cyclin A2 promoter contains an NFAT binding site"

Table 4.27 Rule 15c illustrated

Text	Entity	POS tag
Negative	Keyword, type=inhibit	adjective
cyclin A2	Gene	noun
contains		verb
NFAT	TF	noun
site		noun

Extracted Triple

TF	relation	Gene
NFAT	inhibits	cyclin A2

Summary:

In this chapter a detailed discussion of the proposed methodology has been presented. The three main modules of the system, i.e. pre-processing module, entity recognition module and triple extraction module have been described in detail. Finally the triple extraction rules are discussed. The rules are explained with example sentences to illustrate the application of rules and the extraction of triples from the matched text.

Chapter 5

Implementation and Evaluation

This chapter has two main parts. The first part will discuss the technical details about the system implementation and the second part will focus on the evaluation of the proposed system.

5.1 System Implementation

This section is further divided into three sub-sections; (i) System Specifications, (ii) Software Specifications, (iii) Sample output of each module, illustrated through a series of screen shots.

5.1.1 System Specifications

The system specifications used in the development of the system are shown in Table 5.1

Table 5.1 System Specifications

Processor	Intel 1.7GHz Corei3 4010U
RAM	4 GB
Operating System	Windows 8.1 Pro

5.1.2 Software Specifications

The software specifications used in the development of the system are shown in Table 5.2

We have used GATE integrated development environment to develop our proposed system for a number of reasons:

Table 5.2 Software Specifications

Development Language	Java version 8 update 40
Framework	GATE Embedded
IDE	GATE Developer 8.0 build 4825

1. It is an open source free software for text engineering.
2. It has a built-in information extraction component called ANNIE [52] that we customized for our pre-processing tasks.
3. The most useful feature that was applicable to our work is JAPE,(Java Annotation Patterns Engine).
   (a) JAPE enabled us to use regular expressions over annotation graphs, instead of linear sequence of strings that traditional regex packages match.
   (b) In certain rules, we needed to execute actions on RHS of rule. For example, in case of multiple keyword rules, the final relation had to be decided if the number of keywords is more than one. JAPE allows to use customized Java code to execute these actions.

5.1.3 Sample output

As discussed in the system methodology chapter, there are three modules in our system. Figure 5.1 shows how the raw text is annotated when it has been pre-processed.

On the right side of the Figure 5.1 different annotation sets are shown. The pre-processing module creates the following annotation sets in pre-processing:

- Token
- Split
- SpaceToken
- Sentence

One of the token annotation with string '*stimulates*' is highlighted in Figure 5.1. The attributes 'kind','length' 'orth' and 'string' are added by the tokeniser, whereas the 'category' attribute is added by POS tagger. The morphological analyser reduces the token string into a root attribute as shown in the figure.

Next Figure 5.2 shows the output from entity recognition module. The methodology and rules for entity recognition have been discussed in the previous chapter.

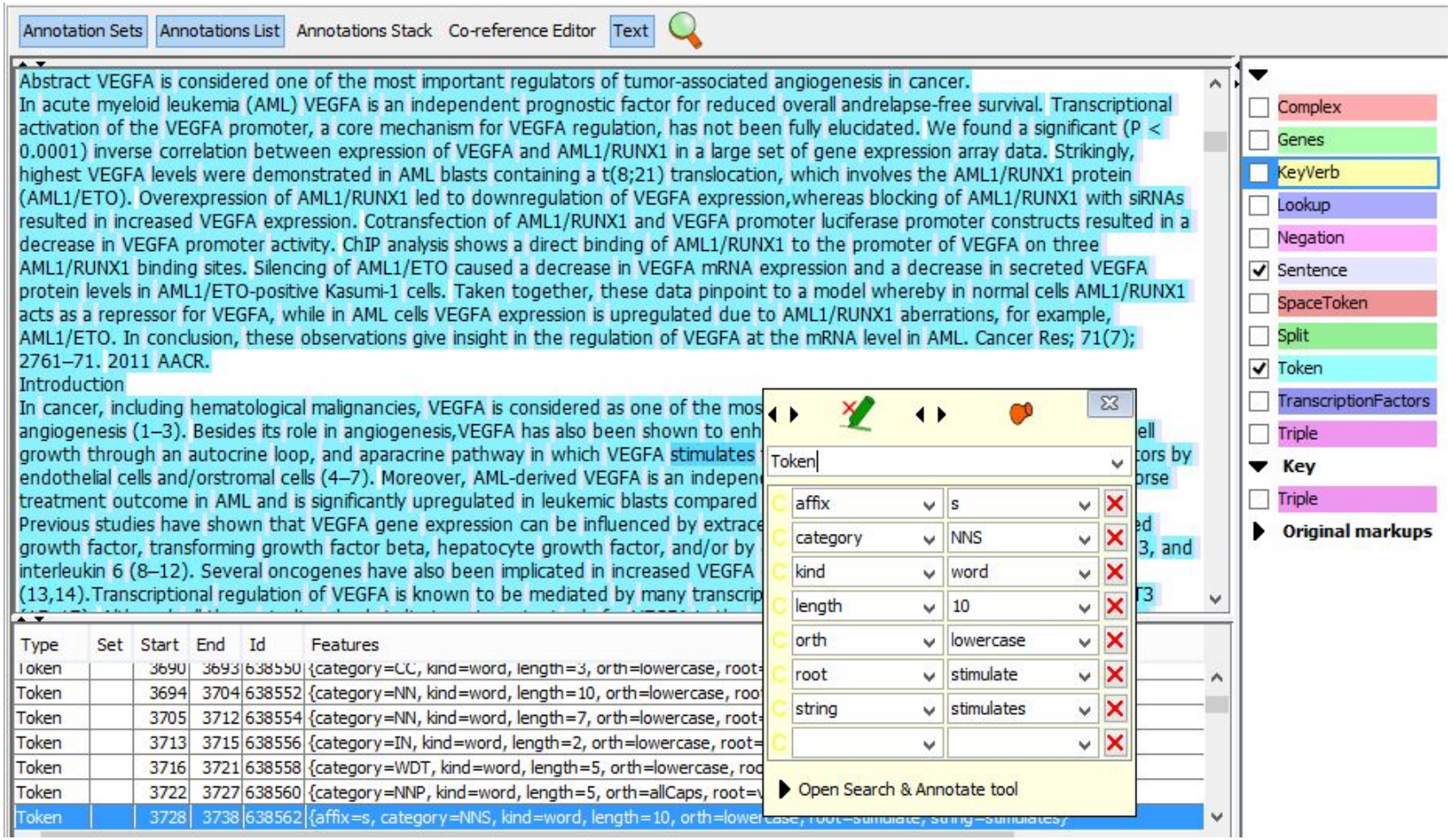

Fig. 5.1 Illustration of Pre-processing Module

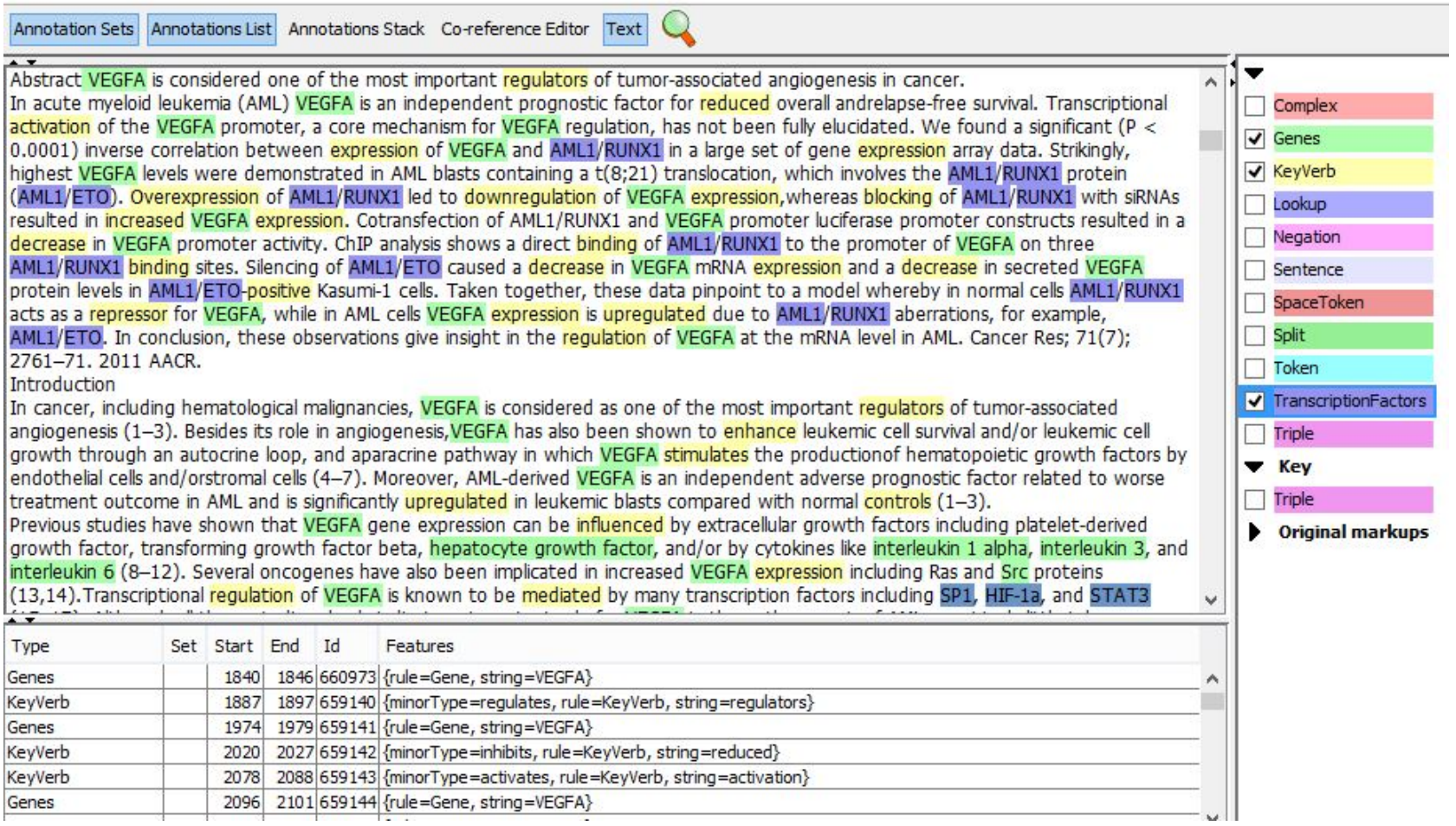

Fig. 5.2 Illustration of Entity Recognition Module

The annotation set 'Genes', 'TranscriptionFactors' and 'KeyVerb' are created by the entity recognition module. The TF and gene entities and relationship keywords are color coded in Figure 5.2. It can be seen that some gene and TF entities overlap in the figure, such as SP1, HIF-1a and STAT3.

The output from triple extraction module is shown in Figure 5.3. One of the triple annotations on the text *'increased expression of cyclin genes in lymphocytes lacking the NFAT1'* is highlighted to show its three attributes; gene, transcription factor and relation.

5.2 Evaluation Approaches

Two different approaches have been applied in the literature while evaluating gene regulatory relationship extraction systems:

5.2.1 Evaluation Approach -1

In this approach, a system identifies sentences that contain gene regulatory information. However the system does not extract the arguments in a triple format. It has been used in [12] and [53]

5.2.2 Evaluation Approach -2

There are two steps in this approach. In the first step, a system identifies sentences that contain gene regulatory information. Then in the second step, arguments of a triple are extracted from the identified sentences. This approach has been used in [10].

5.3 Evaluation Metrics

Before introducing the metrics, we first define how we interpret the terms that are used in calculating the metrics. These terms are (i) True Positive, (ii) False Positive and (iii) False Negative.

True Positive: A triple annotation generated by the system is considered a 'true positive' if the system accurately identifies each of the three components of the triple i.e., gene, transcription factor and the relationship between them. If any of these three components is missing or annotated wrongly, the triple is not considered to be true positive.

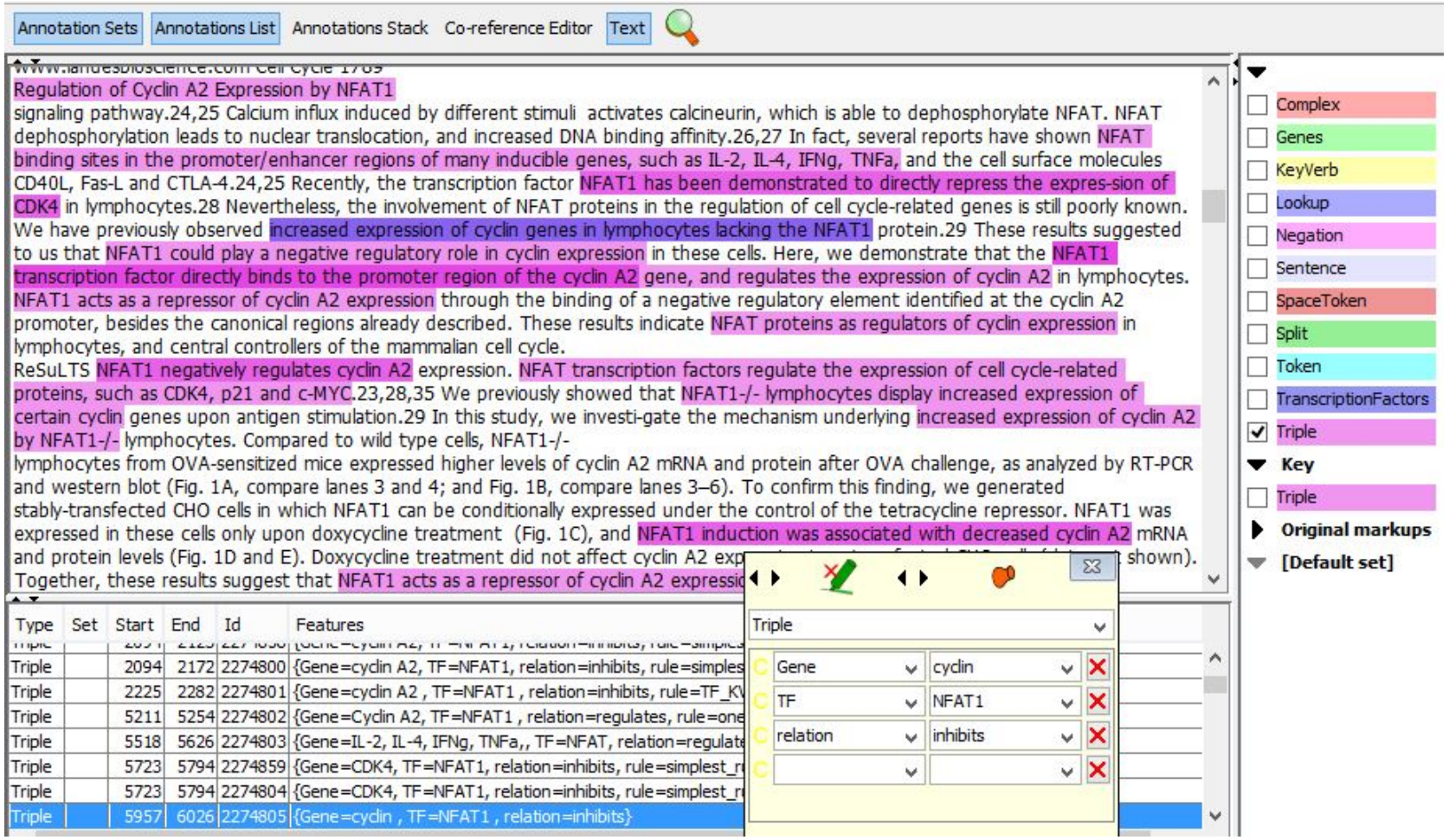

Fig. 5.3 Illustration of Triple Extraction Module

False Positive: A triple annotation is considered to be a 'false positive' if it is generated by our proposed system but is not present in the manually extracted gold standard triple annotation set.

False Negative: A triple annotation is considered as a 'false negative' when it is present in the gold standard triple annotation set but not generated by the proposed system.

We have selected the widely used measures i.e., precision, recall and f-measure to evaluate the performance of our proposed extraction system. These measures are defined below [54]:

Precision

Precision measures the number of correctly identified triples as a percentage of number of triples identified. In other words, it measures how many of the triples that the system identified are correct. Equation (5.1) shows the formula:

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives} \tag{5.1}$$

Recall

Recall measures the number of correctly identified triples as a percentage of the total number of correct triples. In other words, it measures how many of the triples that should have been identified are actually identified. Equation (5.2) shows the formula:

$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives} \tag{5.2}$$

F-measure

F-measure is the harmonic mean of precision and recall, calculated as:

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.3}$$

5.4 Dataset Specifications

In our experiments we have used three different datasets to evaluate and compare our system with existing gene regulatory relationship extraction systems. The specifications of each dataset are described below:

5.4.1 HIF1 Dataset

This is a publicly available dataset used by Tsung Tang et al. for evaluation of their system, AutoPat [53]. This dataset consists of 30 abstracts of articles related to the HIF1 transcription factor. The dataset is divided into 323 sentences, out of which 88 sentences contain information about regulation of target genes by HIF1.

5.4.2 E2F1 Dataset

Just like HIF dataset, this dataset is also used by Tsung Tang et al.[53] for evaluation of AutoPat. It consists of 142 sentences from articles related to the E2F1 transcription factor. 54 among those 142 sentences describe regulation of target genes by E2F1.

5.4.3 Miscellaneous Dataset

To date there is no publicly available dataset that contains human gene regulatory relationship annotations in the form of triples. Although there exist datasets that annotate transcription factors and their target genes [55], yet the type of interaction is missing in the available datasets. Therefore we decided to create a corpus from randomly selected documents about different genes and transcription factors involved in a variety of diseases. The corpus consists of 20 documents. The total number of sentences in the selected documents was 9182. This corpus was manually annotated by a domain expert. Since several triples may refer to the same regulatory relationship, we only consider the unique triples from each document. The total number of unique triples extracted by the domain expert from the corpus was 305. These triples were then used as gold standard to evaluate performance of our system, in terms of target gene detection, the transcription factors and the type of interaction between them.

5.5 Performance Evaluation

5.5.1 Approach-1

As mentioned before, evaluation approach-1 was used in [1]. We executed the proposed system on HIF1 and E2F1 datasets to compute precision, recall and f-measure using approach 1.Then the computed results were compared with the results of [1] on the same datasets, as shown in Figure 5.4 and Figure 5.5

The illustrations show that the proposed system achieves better results than the existing ones. The comparison among the systems on the basis of methodology is shown in Table 5.3

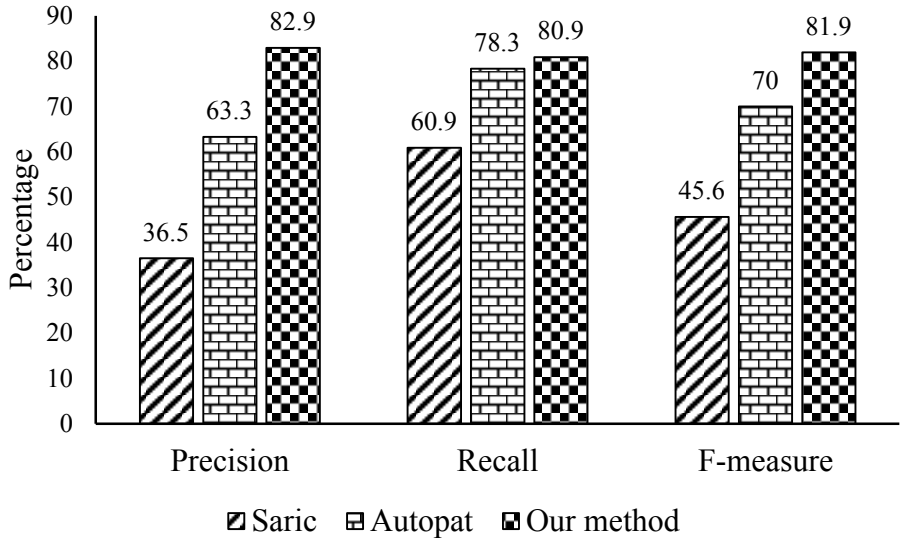

Fig. 5.4 Performance comparison using E2F1 dataset

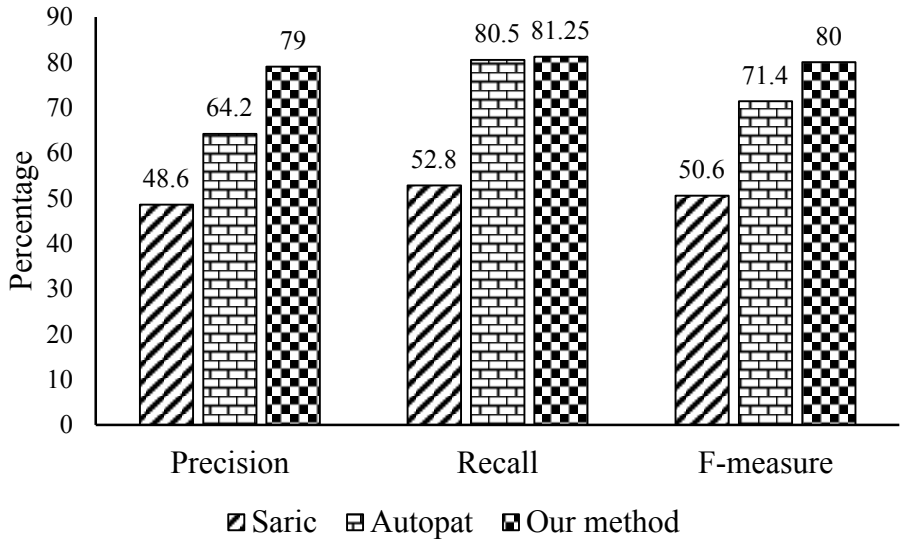

Fig. 5.5 Performance Comparison using HIF1 dataset

Table 5.3 Comparison based on methodology

System	Method	Machine learning	Manual Rules	Multiple Key-words	No Key-word
AutoPat	Pattern Discovery	Yes	Yes	No	No
Saric rule based	Rule	No	Yes	No	No
Our system	**Rule**	**No**	**Yes**	**Yes**	**Yes**

5.5.2 Approach-2

Evaluation approach-2 identified sentences that contain gene regulatory information and then extracted arguments of triples from the identified sentences, as discussed earlier. Saric [10] used this approach. Although the rule based approach of Saric [10] recognizes relation chunks between named entities, their extraction results are still embedded in the form of annotated sentences. Their rules do not describe the extraction of triples from the semantic annotations within the sentences. Therefore we cannot compare our system's triple extraction method with existing approaches. To evaluate the effectiveness of our proposed system in term of relationship extraction, we executed the proposed system on miscellaneous dataset. The total number of triples extracted by our proposed system from the miscellaneous dataset was 867. However, as discussed in section 5.4.3 we only consider the unique triples from each document. The number of unique triples extracted by our system from the corpus was 333. The details of the extracted triples are given in Table 5.4.

Table 5.4 Document Statistics and Evaluation of Extraction Results

Document No.	Manually Extracted Triples	System Generated Triples	Precision (p) (%)	Deviation about mean squared $(p-\bar{p})^2$	Recall(r) (%)	Deviation about mean squared $(r-\bar{r})^2$
1	35	35	77.14	26.6	77.14	162.8
2	9	12	75	53.3	100	102
3	8	7	85.7	11.6	75	222
4	12	15	73.3	81	91.66	3.09
5	25	22	77.27	25.3	68	479.6
6	17	24	70.8	132.3	100	102
7	9	11	81.8	0.25	100	102
8	13	16	75	53.3	92.3	5.8
9	14	17	76.4	34.8	92.8	8.4
10	12	14	78.57	13.9	91.6	2.9
11	13	12	91.6	86.5	84.6	28
12	11	14	78.57	13.91	100	102
13	20	18	94.4	146.4	85	24
14	17	17	100	313.2	100	102
15	7	6	100	313.2	85.7	17.6
16	23	29	79.3	9	100	102
17	19	22	77.2	26.01	89.4	0.25
18	11	13	76.9	29.16	90.9	1
19	11	11	100	313.3	100	102
20	19	18	77.7	21.16	73.6	265.7
Total	305	333	1646.65	1704.3	1797.77	1935.3

In the experiment precision and recall were calculated for each document. Then the average precision and recall were computed using equation 5.4 and 5.5. This macro-averaging ensures that even if the dataset varies in size, every document is given an equal weight. F-measure was computed using average precision and recall values as shown in equation 5.6. Equations 5.7 and 5.8 compute the standard deviation of precision and recall respectively. The standard deviation of precision was calculated to be 9.5, while standard deviation of recall was 10.

$$Average\,Precision = \frac{\sum_{i=1}^{20} Precision_i}{20} = \frac{1646.65}{20} = 82.3\% \tag{5.4}$$

$$Average\,Recall = \frac{\sum_{i=1}^{20} Recall}{20} = \frac{1797.77}{20} = 89.9\% \tag{5.5}$$

$$Average\,F-measure = 2 \cdot \frac{0.823 \cdot 0.899}{0.823 + 0.899} = 85.9\% \tag{5.6}$$

$$s_p = \sqrt{\frac{1}{20-1}\sum_{i=1}^{20}(p_i - \overline{p})^2} = \sqrt{\frac{1704.3}{19}} = 9.5 \tag{5.7}$$

$$s_r = \sqrt{\frac{1}{20-1}\sum_{i=1}^{20}(r_i - \overline{r})^2} = \sqrt{\frac{1935.3}{19}} = 10 \tag{5.8}$$

The average precision, recall and f-measure for our gold standard evaluation is shown graphically in Figure 5.6.

5.5.3 Result Discussion

Results for both the HIF1 and E2F1 datasets show similar pattern in terms of effectiveness. Our technique performs significantly better than AutoPat and rule based method of Saric et al. Our system improves both precision and recall at the same time. The reason for increased recall is that our rules include a more variety of patterns including multiple keyword cases as well as cases where there is no explicit relationship keyword. Another reason that makes our patterns more generalized is that the POS of the keyword is not only restricted to noun or verb, as in Saric et al. approach. Our patterns use keywords in various forms such as noun, verb, adjective as well as adverbs and this allows for greater coverage. The higher precision is caused by the disambiguation rules that ensure that the gene and TF entities are detected as accurately as possible. The rules developed by Saric et al. suffer from low precision and

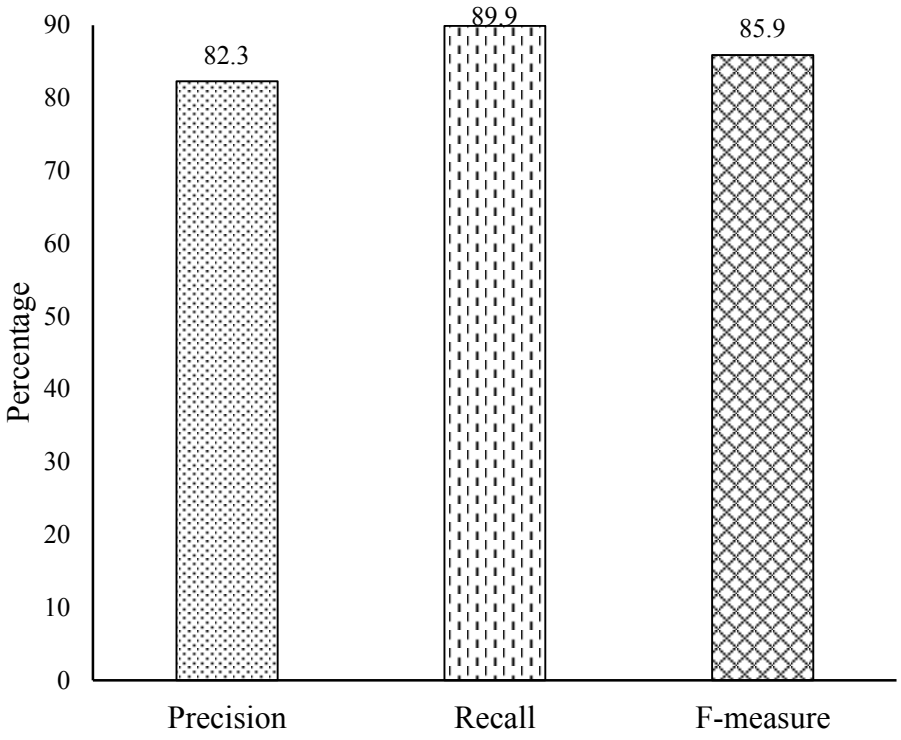

Fig. 5.6 Evaluation against Gold Standard Triples Corpus

recall. The main problem is that their system handles a limited number of patterns due to self-imposed restraints, for example, they do not consider regulatory relationships that are reported to occur after genetic modifications. The AutoPat system performs better than that of Saric et al. However, it has lower precision compared to our system and the reason is that it tends to over-generalize the criteria for extracted sentences. However, merely considering the entities and the verbs in the sentence does not guarantee that the sentence really describes a gene regulation relationship. In addition to this, their system does not handle sentences which lack key verbs and this makes their recall lower than ours. Not only does our system identify sentences with increased precision and recall, but also it can effectively extract the arguments of triples as shown by the results for the miscellaneous dataset.

Summary

In this chapter we have discussed our system's implementation and evaluation. The hardware and software specifications are described and the output of system modules are illustrated with screen shots. The datasets used for the system evaluation and comparison have been specified. The system evaluation against gold standard triple annotation set was discussed, followed by

a discussion on system's comparison with existing techniques based on methodology as well as precision, recall and f-measure.

Chapter 6

Conclusion and Future Direction

In this chapter we present a summary of the contributions of the research work documented in this thesis. Some of the fundamental limitations of our approach and an outlook of the future directions where this work can be extended are presented at the end of this chapter.

6.1 Conclusion

Transcription factors play a central role in the regulation of gene expression and the study of regulatory relationships between TFs and their genes is crucial for genome biology. Many studies have been carried out to extract molecular interactions from text in the biological domain. The current techniques use patterns or machine learning based methods to extract these relationships. The machine learning approaches are hindered by unavailability of training data in bio medical domain, whereas the existing pattern based systems are limited to single interaction keywords. Ignoring the sentences with multiple interaction keywords may result in reduced precision. An additional keyword may cause the extracted relation to be entirely opposite of what was actually intended. Moreover, the systems which assume that potential sentences must have a keyword also miss a number of correct relationships resulting in extraction errors. So the focus of this research is to extract TF-gene interactions using a rule based approach with added multi-keyword and no-keyword support to improve the overall extraction accuracy.

In this research a system has been proposed that automatically extracts interaction between TFs and their target genes from full-text articles through domain-specific dictionaries. The proposed system consists of three modules: 1) Pre-processing Module, 2) Entity Recognition Module and 3) Triple Extraction Module. We have used three kinds of relations; activation, inhibition and underspecified regulation, to represent relation between a transcription factor and gene. We have evaluated our system using two approaches and three data sets. The

first approach test accuracy of our system in terms of detection of target genes. On both of the E2F1 and HIF1 data sets, our system performed better than the existing systems. In the second approach, we tested how accurately our system extracts TF-relation-gene triples. The extraction performance in the second approach is 82.3% for precision, 89.9% for recall and 85.9% for f-measure. The experimental results show that by integration of features like multi-keyword support and named entity disambiguation , and by considering no-keyword cases, we can achieve reasonable precision and recall rates with a simple rule based technique.

6.2 Contributions

6.2.1 Named Entity Disambiguation

Entity disambiguation rules have been specifically designed to address the challenge posed by overlapping gene and protein nomenclature. Disambiguating gene and TF entities before applying triple extraction rules minimizes false positives in the final output.

6.2.2 Multi-keyword and no-keyword support

A special category of rules has been designed to address cases when the number of relationship keywords in the sentences is more than one. Similarly the special case rules cover the sentences when the relationship is expressed without using any keyword. The improved precision as well as recall can be attributed to these two features together with with named entity disambiguation.

6.2.3 Reusability

With minimal modifications, our system can be adapted to other organisms as well as extraction of other interactions such as protein-protein interactions. We only have to replace the list of genes/protein names and symbols to adapt the system for another organism. Similarly, replacing the list of keywords and minimal extension of rules would adapt our system to extract protein interactions successfully.

6.2.4 Simplicity

Compared to other approaches our system is fairly straight forward to implement and achieves competitive performance.

6.3 Limitations and Future Direction

Incomplete named entity dictionaries can limit our system performance specially when new gene/ transcription factors are discovered. Incremental enhancement in these dictionaries is time-consuming, but it is also perfectly feasible and does not represent especially difficult technical hurdles. Moreover, the current methodology focuses on text which describes gene regulation with in the boundaries of a single sentence. The approach can be further improved by co-reference resolution that will enable the system to extract relationship text that spans more than a sentence, up to a paragraph. Moreover the interpretation of tables and figures are not the focus of this research. However, these are of primitive importance as they provide multiple interactions, in a manner well suited for human readers, but enormously difficult for computer processing.

Bibliography

[1] Yi-Tsung Tang et al. “Using unsupervised patterns to extract gene regulation relationships for network construction”. In: *PloS one* 6.5 (2011), e19633.

[2] Peder Larsen and Markus Von Ins. “The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index”. In: *Scientometrics* 84.3 (2010), pp. 575–603.

[3] Carlos Rodríguez-Penagos et al. “Automatic reconstruction of a bacterial regulatory network using Natural Language Processing”. In: *BMC bioinformatics* 8.1 (2007), p. 293.

[4] Jean Villard. “Transcription regulation and human diseases”. In: *Swiss medical weekly* 134 (2004), pp. 571–579.

[5] Florian Leitner et al. “Mining cis-regulatory transcription networks from literature”. In: *Proceedings of BioLINK SIG 2013* (2013), pp. 5–12.

[6] Jung-Hsien Chiang, Hsu-Chun Yu, and Huai-Jen Hsu. “GIS: a biomedical text-mining system for gene information discovery”. In: *Bioinformatics* 20.1 (2004), pp. 120–121.

[7] Minoru Kanehisa et al. “Data, information, knowledge and principle: back to metabolism in KEGG”. In: *Nucleic acids research* 42.D1 (2014), pp. D199–D205.

[8] *DEEPER: A full parsing based approach to protein relation extraction*. Springer, 2008, pp. 36–47.

[9] Jasmin Šarić et al. “Extracting regulatory gene expression networks from PubMed”. In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2004, pp. 191–198.

[10] Jasmin Šarić et al. “Extraction of regulatory gene/protein networks from Medline”. In: *Bioinformatics* 22.6 (2006), pp. 645–650.

[11] Hei-Chia Wang et al. “Inference of transcriptional regulatory network by bootstrapping patterns”. In: *Bioinformatics* 27.10 (2011), pp. 1422–1428.

[12] Hui Yang, Goran Nenadic, and John A Keane. “Identification of transcription factor contexts in literature using machine learning approaches”. In: *BMC bioinformatics* 9.Suppl 3 (2008), S11.

[13] Stein Aerts et al. “Text-mining assisted regulatory annotation”. In: *Genome biology* 9.2 (2008), R31.

[14] Wikipedia. *DNA — Wikipedia, The Free Encyclopedia*. [Online; accessed 4-December-2014]. URL: http://en.wikipedia.org/w/index.php?title=DNA&oldid=665764099.

[15] Marshall Nirenberg. "Historical review: Deciphering the genetic code–a personal account". In: *Trends in biochemical sciences* 29.1 (2004), pp. 46–54.

[16] Wikipedia. *Gene expression — Wikipedia, The Free Encyclopedia*. [Online; accessed 4-December-2014]. URL: http://en.wikipedia.org/w/index.php?title=Gene_expression&oldid=665735081.

[17] Genetics Home Reference. *How Genes Work*. [Online; accessed 5-December-2014]. 2015. URL: http://ghr.nlm.nih.gov/handbook/howgeneswork.

[18] University of Leicester. *Gene expression and regulation*. [Online; accessed 4-December-2014]. 2015. URL: http://www2.le.ac.uk/departments/genetics/vgec/schoolscolleges/topics/geneexpression-regulation.

[19] Robert Tjian. *Gene Regulation: An Introduction*. [Online; accessed 7-December-2014]. URL: http://www2.le.ac.uk/departments/genetics/vgec/schoolscolleges/topics/geneexpression-regulation.

[20] *Gene Expression*. [accessed 6-December-2014]. URL: www.nature.com/scitable/topicpage/gene-expression-14121669.

[21] Wikipedia. *Natural language processing — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-December-2014]. URL: http://en.wikipedia.org/w/index.php?title=Natural_language_processing&oldid=665643174.

[22] Wikipedia. *Tokenization (lexical analysis) — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-December-2014]. URL: http://en.wikipedia.org/w/index.php?title=Tokenization_(lexical_analysis)&oldid=637328355.

[23] Sebastian Blohm. "Large-scale pattern-based information extraction from the world wide web". PhD thesis. 2011.

[24] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[25] Wikipedia. *Part-of-speech tagging — Wikipedia, The Free Encyclopedia*. [Online; accessed 10-December-2014]. URL: http://en.wikipedia.org/w/index.php?title=Part-of-speech_tagging&oldid=663559091.

[26] Mark Hepple. "Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers". In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2000, pp. 278–277.

[27] Eric Brill. "A simple rule-based part of speech tagger". In: *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics. 1992, pp. 112–116.

[28] University of Sheffield NLP. *Module 2: Introduction to IE and ANNIE*. [Lecture slides; accessed 10-December-2014]. 2011. URL: https://gate.ac.uk/sale/talks/gate-course-may11/track-1/module-2-ie/module-2-ie.pdf.

[29] Wikipedia. *Triplestore — Wikipedia, The Free Encyclopedia*. [Online; accessed 7-June-2015]. URL: http://en.wikipedia.org/w/index.php?title=Triplestore&oldid=642239431.

[30] Tor-Kristian Jenssen et al. "A literature network of human genes for high-throughput analysis of gene expression". In: *Nature genetics* 28.1 (2001), pp. 21–28.

[31] Benjamin J Stapley and Gerry Benoit. “Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts”. In: *Pac Symp Biocomput*. Vol. 5. 2000, pp. 529–540.

[32] Pierre Zweigenbaum et al. “Frontiers of biomedical text mining: current progress”. In: *Briefings in bioinformatics* 8.5 (2007), pp. 358–375.

[33] Udo Hahn et al. “How feasible and robust is the automatic extraction of gene regulation events?: a cross-method evaluation under lab and real-life conditions”. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*. Association for Computational Linguistics. 2009, pp. 37–45.

[34] Hamish Cunningham et al. “Getting More Out of Biomedical Documents with GATE’s Full Lifecycle Open Source Text Analytics”. In: *PLoS Comput Biol* 9.2 (Feb. 2013), e1002854. DOI: 10.1371/journal.pcbi.1002854. URL: http://dx.doi.org/10.1371/journal.pcbi.1002854.

[35] David D. Palmer. “Text Preprocessing”. In: *Handbook of Natural Language Processing*. 2nd ed. CRC Press, Taylor and Francis, 2010. Chap. 2, pp. 9–30.

[36] CraigTrim. *Language Processing. The Art of Tokenization*. Jan. 24, 2013. URL: https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization (visited on 01/16/2015).

[37] CLARIN-D/SfS-Uni. Tübingen. *WebLicht: Web-Based Linguistic Chaining Tool*. Apr. 2, 2013. URL: https://weblicht.sfs.uni-tuebingen.de/ (visited on 01/12/2015).

[38] Daniel Hanisch et al. “ProMiner: rule-based protein and gene entity recognition”. In: *BMC bioinformatics* 6.Suppl 1 (2005), S14. DOI: 10.1093/nar/gkq1237.

[39] Katrin Fundel and Ralf Zimmer. “Gene and protein nomenclature in public databases”. In: *Bmc Bioinformatics* 7.1 (2006), p. 372.

[40] Burr Settles. “ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text”. In: *Bioinformatics* 21.14 (2005), pp. 3191–3192.

[41] Ryan McDonald and Fernando Pereira. “Identifying gene and protein mentions in text using conditional random fields”. In: *BMC bioinformatics* 6.Suppl 1 (2005), S6.

[42] Yoshimasa Tsuruoka et al. “Developing a robust part-of-speech tagger for biomedical text”. In: *Advances in informatics*. Springer, 2005, pp. 382–392.

[43] D. Maglott et al. “Entrez Gene: gene-centered information at NCBI”. In: *Nucleic Acids Res.* 39.Database issue (Jan. 2011), pp. D52–57.

[44] A. Bateman et al. “UniProt: a hub for protein information”. In: *Nucleic Acids Res.* 43.Database issue (Jan. 2015), pp. D204–212.

[45] Alfred V Aho, Brian W Kernighan, and Peter J Weinberger. “Awk–a pattern scanning and processing language”. In: *Software: Practice and Experience* 9.4 (1979), pp. 267–279.

[46] Juan M Vaquerizas et al. “A census of human transcription factors: function, expression and evolution.” In: *Nature Reviews Genetics* 10.4 (2009).

[47] Edgar Wingender, Torsten Schoeps, and Jürgen Dönitz. “TFClass: an expandable hierarchical classification of human transcription factors”. In: *Nucleic acids research* 41.D1 (2013), pp. D165–D170.

[48] Hong-Mei Zhang et al. "AnimalTFDB: a comprehensive animal transcription factor database". In: *Nucleic acids research* 40.D1 (2012), pp. D144–D149.

[49] Wikipedia. *Gene nomenclature — Wikipedia, The Free Encyclopedia*. [Online]. URL: http://en.wikipedia.org/w/index.php?title=Gene_nomenclature&oldid=646964645 (visited on 01/12/2015).

[50] Weronika Sikora-Wohlfeld et al. "Assessing computational methods for transcription factor target gene identification based on ChIP-seq data". In: *PLoS computational biology* 9.11 (2013), e1003342.

[51] Bioconductor. *Finding Candidate Binding Sites for Known Transcription Factors via Sequence Matching*. [Online]. 2015. URL: http://www.bioconductor.org/help/workflows/gene-regulation-tfbs/ (visited on 06/03/2015).

[52] Hamish Cunningham et al. "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications". In: *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. 2002.

[53] *AutoPat*. 2011. URL: http://ikmbio.csie.ncku.edu.tw/AutoPat/AutoPat_dataset/datasets.html (visited on 07/16/2014).

[54] Hamish Cunningham et al. *Developing Language Processing Components with GATE Version 8*. Nov. 17, 2014.

[55] Claire Nédellec. "Learning language in logic-genic interaction extraction challenge". In: *Proceedings of the 4th Learning Language in Logic Workshop (LLL05)*. Vol. 7. Citeseer. 2005.