Gene Interaction Extraction from Biomedical Texts

Master Thesis

Přemysl Vítovec

Abstract

The presented report describes a method of text preprocessing improving the performance of sequential data mining applied in the task of gene interaction extraction from biomedical texts. The need of text preprocessing rises primarily from the fact, that the language encoded by any general word sequence is mostly not sequential. The method involves a number of heuristic language transformations, all together converting sentences into forms with higher degree of sequentiality. The core idea of enhancing sentence sequentiality results from the observation that the components constituting the semantical and grammatical content of sentences are not equally relevant for extracting a highly specific type of information. The experiments employing a simple sequential algorithm confirmed the usability of the proposed text preprocessing in the gene interaction extraction task. Furthermore, limitations identified during the result analysis may be regarded as guidelines for further work exploring the capabilities of the sequential data mining applied on linguistically preprocessed texts.

Abstrakt

Předkládaná práce popisuje metodu textového předzpracování, jež zlepšuje výkon sekvenčního data miningu v úloze extrakce genových interakcí z biomedicínských textů. Potřeba textového předzpracování vychází především ze skutečnosti, že jazyk zakódovaný obecnou slovní sekvencí nemá ve většině případů sekvenční charakter. Metoda sestává z několika heuristických jazykových transformací, jež společně převádí věty na formy s větší mírou sekvenčnosti. Hlavní myšlenka zvyšování větné sekvenčnosti je založena na pozorování, že jednotlivé komponenty, jež společně tvoří sémantický a gramatický obsah vět, nejsou stejně relevantní pro extrakci úzce specifického typu informace. Provedené experimenty využívající jednoduchého sekvenčního algoritmu potvrdily použitelnost navrženého předzpracování v úloze extrakce genových interakcí. Omezení, jež vyvstala z analýzy výsledků, lze navíc považovat za vodítka pro další studium aplikace sekvenčního data miningu na lingvisticky předzpracované texty.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 12. května 2010 ....

podpis

Poděkování

Děkuji Ing. Jiřímu Klémovi, Ph.D. za vedení, inspiraci a výbornou spolupráci.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Gene interaction extraction from textual language representation can succeed only if language is understood correctly. In general, language comprehension proceeds through interpretation of grammar, semantics and pragmatics; omission of any of these components may cause the communication to fail. Individual language variants may differ in complexity of these components; biomedical language proves to be complex in all of them. Being the complexity extremely hard, any engineering approach has to omit some aspects by making assumptions, permitting relaxations etc. In case of sequential approach, which is focused in this project, this is expressed by assumption that language is of sequential nature. To diminish the negative effect of such a simplification while keeping the full potential power and flexibility of the sequential approach unchanged, a kind of text preprocessing may be employed. The presented report describes a method of text preprocessing which aims at enhancing the results of any sequential approach in the task of gene interactions extraction from biomedical texts.

The outline of the report is as follows: (1) A bibliographical overview of methods commonly used in *named entity recognition* and *gene interaction extraction* is presented (chapter 2, page 6); (2) language components disproving the assumption of language sequentiality are identified and a method of biomedical text preprocessing dealing with these obstacles is derived based on various linguistic observations (chapter 3, page 17); (3) the impact of the designed text preprocessing method is evaluated from various points of view using a simple sequential approach, limitations of the derived method are analyzed in detail (chapter 4, page 37).

Chapter 2

Methodological Overview

2.1 Introductory Remarks

In the gene interaction extraction task, relations between specific word entities are of primary interest. Concentrating exclusively on the relation extraction task implies the operational space defined in a very specific way: being given a biomedical text, the starting point is (1) zero knowledge of the given text, except for (2) a detailed information of what word tokens are gene entities. Such disparity in knowledge level suggests a simplification being made, namely that the *gene named entity recognition* has been accomplished. However, tasks assigned by the real world do not naturally provide any kind of such exclusive and highly specific knowledge, i.e. named entities (gene names) are not known. What word tokens are the ordinary words, can be to high degree of accuracy determined using existing thesauri, i.e. large word banks covering the great part of English lexicon; in contrast, it proves to be extraordinarily difficult to determine, what words represent the entity words. Therefore, before introducing methods employed in the *gene interaction extraction task*, an overview of *gene named entity recognition* is presented.

2.2 Overview of Named Entity Recognition

2.2.1 Gene Named Entity Recognition

Named entity recognition includes three subtasks [37,40]: (1) *term recognition*, (2) *term classification* and (3) *term mapping*. The *term recognition* is a task of finding adjacent tokens representing a concept of the given domain, i. e. genes entity names in biomedical domain. The *term classification* or *term categorization* classifies extracted terms into predefined classes (e.g. *genes*, *mRNAs* etc.) [37]. The *term mapping* stands for a task of selecting a preferred term to represent the recognized concept in case that synonyms exist (they typically exist) [37,40]. The recognized term is not actually identified until the mapping has been done [37].

The performance of the NER (and in general all text mining tasks) is measured in terms of *precision* and *recall*. Assuming A stands for the set of positive units not extracted by the tested system ($\sim$ false negative), B for the set of extracted positive units ($\sim$ true positive) and

C for the set of units falsely extracted as positive ($\sim$ false positive), the *precision* and *recall* are defined [37,66] as

$$Precision = \frac{B}{B+C}, \qquad Recall = \frac{B}{A+B}. \tag{2.1}$$

The overall performance is often expressed by the *F-measure* defined [66] as

$$F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}, \tag{2.2}$$

where β is typically set to 1.

Methodological approaches applied in domain of the named entity recognition are often devided in *rule-based* (i. e. grammar-based and respective rule-based approaches) and *approaches based on machine learning and statistics* [11,51]. In the following overview, the finer classification presented by *Zhou and He* is followed [66]: approaches are devided into *computational linguistics-based approaches*, *rule-based approaches* and *machine learning and statistical approaches*. Note that sometimes one more category is distinguished, namely *dictionary-based approaches* [37]. However, this category is not considered here, since the use of dictionaries appears to be a common feature of all other listed approaches. Furthermore, one more note is to be made: existing solutions for named entity recognition tasks often combine several approaches, e.g. (i) machine learning engine and rule-based pre- and postprocessing, (ii) the use of machine learning and statistical methods in computational linguistics-based approaches etc.

2.2.2 Obstacles

The main obstacles in named entity recognition tasks can be summarized in two terms: *ambiguity* and *variability* of gene names [37]. Both difficulties, in principle, arise from the same aspect of the rapidly growing (and changing) biomedical nomenclature, namely the insufficient standardization of the biomedical terms [40].

Term ambiguity results from *term homonymy* or *homology*, i. e. one name expression is used for multiple different genes [64]. Chen et al. [8] distinguish four ambiguity types: (1) *intra-species* ambiguity, (2) ambiguity with *general English words*, (3) ambiguity with *general medical terms* and (4) *across-species* ambiguity. The *across-species* ambiguity occurs by far most frequently, however, only the *intra-species* ambiguity might by labeled negligible [8]. Moreover, it has been observed, that the complexity of the ambiguity problem varies among different organisms [64]. Abbreviations and acronyms appear to be frequent source of ambiguity [37].

Term variability, on the other hand, results from spelling ($\sim$ formal) differences [37] and *term synonymy* [8,40,64]. The formal variability was widely discussed by Cohen et al. [10]. They speek about *non-contrastive variability* and design four heuristics that cover typical problems concerning variability: (1) *equivalence of vowel sequences*, (2) *optionality of hyphens*, (3) *optionality of parenthesized material* and (4) *case insensitivity*. The term synonymy means that multiple name expressions (full names, official symbols and their synonyms) are associated with one gene [8,40,64]. Chen at al. find out that synonyms are mostly preferred to official symbols and full names [8].

2.2.3 Computational Linguistics-based Approaches

Methods discussed in this section construct grammars describing the internal structure of proper noun phrases, in both *morphological* [2,26,27,34] and/or *syntactical* [54] manner.

The fundamentals of the term identidication using grammars were summarized by Ananiadou [2]. In the proposed morphology model, a *four-level ordered* approach is applied, each level (∼ stage) employing a different term formation principle {*latinate/native compounding*, *latinate/native affixation*}. Moreover, three *boundness* levels of word roots are assumed. Distinguishing between *term* and *word* (∼ non-term) wordtype or affix values (∼ semantic categories) and being provided a lexicon of specific roots, affixes and Greek/Latin combining words, the resulting morphological unification grammar operates as follows:

```
term → word + term_suffix          word → word + word_suffix
term → term + term_suffix          term → term + word_suffix
```

Note that in the example rules new forms are generated using the *suffixation* principle; the *prefixation* and *compounding* principles behave in similar manner. For deeper view refer to [2].

Gaizauskas et al. [26,27,34] applied terminological context-free grammar in systems EMPATHIE [34] and PASTA [26,34]. The architecture of these systems splits in several modules: After preliminary *text processing*, the systems proceeds with *terminological processing* applying firstly *morphological analysis* to find out biochemical affixes (e.g. *-ase, -in*) and following with *lexical lookup* in terminological lexicons (compiled from biological recourses) to determine component categories. The core of the system is the *terminology parsing* [26,27,34]: being given e.g. [26] *casein kinase 1*, the components categories are recognized as *protein_modifier*, *protein_head* and *numeral*, the corresponding grammar rule is

```
protein → protein_modifier, protein_head, numeral.
```

Furthermore, Pyysalo et al. employ the *link grammar* formalism [54]. This grammar builds on notions of *links* (e.g. *adjective to noun, preposition to opbject*) and *linkages*: ”A linkage consists of a set of links connecting the words of a sentence so that links do not cross, no two links connect the same two words, and thew types of the links satisfy the *linking requirements given to each word in the lexicon*” [54]. Detailed description and examples are available in [54].

2.2.4 Rule-based Approaches

Approaches referred in this section define a set of rules describing relationships between (1) structural and textual elements of the given proper noun phrases or (2) more general categories of these elements (similarly in [66]). These relationships are called *patterns* [66], abstractions that can be successfully applied on unknown text. The patterns may be expressed by means of regular expressions [11].

One of the earliest rule-based systems was proposed by Fukuda et al. [23]. The PROPER system defines a rule set for each of its three stages. First, *core-terms*, the most determinative words of compound terms (e.g. *SH3, p54, SAP*), and *feature-terms*, words describing a function or characters of compound terms (e.g. *receptor, gene*) are annotated, second, *core-blocks* are built from adjacent *core-terms*, *feature-terms* or nouns/adjectives between them using concatenation rules, and finally relations between *core-blocks* are determined mainly according to the connective words (e.g. *and, of*).

Narayanaswamy et al. [48] take up the approach proposed by Fukuda et al. and extend it especially by classifying the *feature-terms* (here referred to as *function-terms*) into six semantic classes: *gene, gene parts, chemicals, chemical parts, source terms* and *general terms*. Furthermore, they apply post-processing rules including the use of adjoining context for term disambiguation: *h-terms* (*h* ∼ *help*, e.g. *expression of*) considered not to be a part of the named entity are introduced to determine the right class where *feature-terms* are missing.

Another rule-based system was presented within the BioCreAtIvE I [65] evaluation by Tamames [60]. Here sentences are tagged by {*central* ∼ *core-terms, chemical, type, location, bioword, other*} and set of rules is applied to identify gene full names. Moreover, special attention is paid to correct symbol identification: Words from the training set are scored for probability of constituting a gene name and the decision of whether the symbol is a gene name is derived from the scores of the symbol and surrounding words, taking into consideration statistically determined *risk-factor*. Furthermore, an additional matching against a dictionary of gene name is made.

Nakov and Divoli [18], also participating in the BioCreAtIvE II competition, apply a set of normalization and expansion rules to a list (∼ dictionary) of EntrezGene gene names: *strong rules* allowing for minor alterations only and *weak rules* conceding more serious alterations of the dictionary names are distinguished.

An innovative submission to the BioCreAtIvE II was made by Neves [18]. The system uses the case-based reasoning: *Cases* are automatically extracted from the training data and stored in two case bases, namely in the *known case base* and the *unknown case base*. The former is built of all words found using various features (*word itself, gene/not, frequency, the same about preceding word*). The *unknown case base* is composed of the formats of all words found using similar features (but not the word itself; format ∼ guideline for unseen word sequences). The cases are used to identify *new cases*: the case assignment is accomplished by measuring the similarity to cases contained in the both case bases.

Another approach was proposed by Hobbs [32]. Here sequential patterns of interest are represented as a kind of finite state automata (here referred to as *transducers*). To examine multiple patterns of interest, the corresponding automata are cascaded. When reaching the acceptance state of the applied automat, the given pattern is recognized. The system called FASTUS works on five levels: (1) complex words, (2) basic phrases, (3) complex phrases, (4) domain patterns, (5) merging structures.

Plantevit et al. [51] identify the limitations of both *sequential patterns* and *sequential rules* applied as single methods on gene mentions extractrion: the former suffer from low precision, the latter, in contrast, from poor recall. They overcome these limitations by introducing the *LSR pattern* defined as a triple (*l* ∼ *left neighborhood, s* ∼ *sequential pattern, r* ∼ *right neighborhood*), in which both methodological principles are combined. Furthermore, relaxed order constraints and support/confidence measures are applied to balance the good trade-off between recall and precision of the fully automated extraction process. Precise description is given in [51].

A widely discussed rule-based solution for the gene name normalization task was the system proposed by HANISH AT AL. [31] within of the BIOCREATIVE I [12] competition. The PROMINER system consists of three subsequent stages: (1) Dictionaries are built from several databases, various curation steps using regular expressions are applied, synonyms are gathered and assigned a synonym class: (i) frequent one-word synonyms (used for disambiguation, therefore augmented with specific context words), (ii) case-sensitive and (iii) case-insensitive synonyms. (2) Gene occurence detection is accomplished by matching the text against the dictionaries using *boundary* and *acceptance measures*. (3) A disambiguation filter is applied.

In the BIOCREATIVE I and II challenges multiple solutions for gene normalization were proposed [12, 17], the majority of them relying on rule based technics similar to those presented by HANISH, moreover, most of them even preserving the scheme (1) dictionary construction and curation, synonyms processing, (2) string matching and (3) disambiguation [17].

2.2.5 Machine Learning and Statistical Approaches

General Remarks

The machine learning approaches can be summarized in the following manner: given a training dataset in which each entity is classified either as positive, i. e. representing the concept of interest, or negative, i. e. representing counter-examples of the concept of interest, the system learns to recognize the positive examples from the negative ones. The ability of discriminating between positive and negative examples is proven on unknown data or testing dataset [18].

As the natural language as the domain of all tasks related to named entity recognition proves to be an extremely complex system, the problem arises of how to capture the linguistic entities. A general approach is to represent the linguistic entities (mostly words, phrases, sentences) as sets or vectors of features (*feature vectors*). Following this approach, two crucial aspects have to be taken into consideration: the *feature selection*, as the features are not equally informative, and the *feature representation*, as a numerical value is needed to express the rate of adherence of the given lexical element to that particular feature [18]. In general the following feature classes are commonly employed in the natural language domain:

- Lexical features. Words itself are often used as features [46,57], SETTLES points out the need of generalization from the concrete word forms appearing in the text [57].

- Orthographic features. This feature class focuses on the graphical representation of the lexical elements (tokens), or as summarized by ZHOU ET AL. it is concerned with capitalization, digitalization and word formation [67]. SETTLES abstracts from the current graphical representation of the words by classifying each letter into one of three classes: *capitals*, *lowercase letters* and *digits* [57]. Orthographic features are commonly expressed using various regular expressions, e.g. expressions defined in [67]: *Parenthesis, RomanDigit, GreekLetter, DigitCommaDigit, AllCaps* etc.

- Morphological features. Biomedical terms are often derived using very specific affixes. ZHOU ET AL. mention examples of typical suffixes: *-ase, zyme, -ome, -gen* etc. [67], MITSUMORI makes systematically use of both prefixes and suffixes in form of letter *uni-, bi- and trigrams* [46]. Moreover, McDonald employs letter *bi-, tri-* and *tetragrams* wherever in the word [44].

- Contextual features. The context often helps to reveal the right identity of the give token. As contextual features may be considered the *trigger words* of the second type (*TW2*) introduced in [67] and defined as the heads of the noun phrases, though not part of the gene name, e.g. *activation, stimulation* etc. Settles employs as contextual features simply the neighboring words [57].

- Semantic features. Settles provides the semantic knowledge in the form of manually formed lexicons, each representing a semantic class such as *amino acids*, *known viruses* etc. If the current word is found in one of these lexicons, it is marked as holding the corresponding feature, e.g. *amino acid.* Furthermore, clearly semantic features are the *trigger words* of the first type introduced in [67] and referred to as heads of noun phrases and at the same time parts of the gene names represented by the given phrase.

- Dictionary features. In [46] the information on whether *token uni-, bi- and -trigrams* were found in the available gene dictionary is used as feature.

- Parts of speech. The parts of speech embody the morphologic, semantic/syntactic/contextual properties of the given token, they are widely applied in many kinds of NER systems. In the biomedical domain, they appear often in adjusted form: the corpus used in the BioCreAtIvE competition only two tags were used to distinguish genes (or parts of genes) from the other tokens [18], frequent tagging approach is the *B-I-O tagging*, where *B* stands for the beginning, *I* for continuing and *O* for the outside of the gene entity [18, 46].

- Preceding class. Class (e.g. *B*, *I*, or *O*) of the token/tokens preceding the current token [46].

The machine learning and statistical approaches include *support vector machines*, *conditional random fields*, *hidden Markov models*, *maximum entropy* and *naive Bayes.*

Support Vector Machines

Given a set training examples $(\mathbf{x}_i, y_i)$, where $\mathbf{x} \in \mathbf{R}^n$ is a feature vector and $y \in \{+1; -1\}$ stands for positive or negative class, the support vector machine (SVM) separates the examples with hyperplyne maximizing the margin, i. e. the distance of the hyperplane to the nearest example vectors called *support vectors.* The hyperplane is defined by the equation

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0, \tag{2.3}$$

where $\mathbf{w} \in \mathbf{R}^n$ is a vector of weights and $b \in \mathbf{R}$, both calculated from the support vectors found. Being provided a set of testing examples $\mathbf{x}_j$, the classification follows the equation

$$\text{sign}(\mathbf{w} \cdot \mathbf{x}) + b = \pm 1. \tag{2.4}$$

Note that in case that the example units are not linearly separable, which is almost always the case, the scalar product in equations 2.3 and 2.4 is replaced by a kernel function K moving the examples into higher-dimensional space, in which they are linear [7, 67]. Detailed description of the SVM can be found in [7]. SVM-based solution of NER problems have been presented e.g. by Mitsomuri at al. [46].

Conditional Random Fields

Conditional random fields (CRF) handle the NER problem as a tagging task: Given the input token sequence $\mathbf{o} = (o_1, o_2, \ldots, o_n)$, the conditional random fields estimate the conditional probability of the candidate tag sequence $\mathbf{t} = (t_1, t_2, \ldots, t_n)$:

$$P(\mathbf{t}|\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \exp \sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i(s_j(\mathbf{t}), o_j). \tag{2.5}$$

where $s_j(\mathbf{t}) = (t_{j-k+1}, \ldots, t_j)$ represents the state as k-gram at position j; f_i encodes the i-th *feature function* from the set of m *feature functions* available; the *feature weight* λ_i refers to the weight of the corresponding *feature function* favoring the tags correlated with the value of the current feature; $Z(\mathbf{o})$ is the normalization factor [44, 57]. As a result, the selected tag sequence maximizes $P(\mathbf{t}|\mathbf{o})$. For more detailed description check [38]. A solution based on CRF has been proposed e.g. by Settles [57].

Maximum Entropy

Being C a set of classes (labels), $\mathbf{x}$ a vector of features (input data), $\mathbf{c}_-$ a vector of previous classifications, f_j a feature function of the j-th feature (i. e. *that we assume the j-th feature for now*) and λ_j the weight of this feature, the maximum entropy approach (ME) determines the probability of the class $c \in C$ as

$$P(c|\mathbf{x}, \mathbf{c}) = \frac{\exp \sum_j \lambda_j f_j(\mathbf{x}, \mathbf{c}_-, c)}{\sum_{c \in C} \exp(\sum_j \lambda_j f_j(\mathbf{x}, \mathbf{c}_-, c))}. \tag{2.6}$$

Thus, the maximum entropy principle operates with the probability distribution over the set of classes C, e.g. $C \in \{gene, nongene\}$ [20,21]. The approach is also referred to as *maximum entropy Markov models* [43]. A system based on the ME approach has been reported e.g. by Chieu et al. [9].

Hidden Markov Models

Assuming the token sequence $T^n = (t_1, t_2, \ldots, t_n)$, the traditional hidden Markov model (HMM) finds the state ($\sim$ tag) sequence $S^n = (s_1, s_2, \ldots, s_n)$ that maximizes the probability

$$\log P(S^n|T^n) = \log P(T^n|S^n) + \log(S^n), \tag{2.7}$$

$$P(T^n|S^n) = \prod_{i=1}^{n} p(t_i|s_i) \quad \text{and} \quad P(S^n) = \prod_{i=1}^{n} p(s_i|s_{i-1}), \tag{2.8}$$

where the Bayes rule was applied [41]. In addition to this traditional HMM, several extensions have been proposed: *mutual information HMM* [41], *consese HMM* [41], *discriminative HMM* [67] and *dictionary HMM* [36]. HMM-based system has been described e.g. by Zhou et al [67].

Naive Bayes

The naive Bayes assigns to a token occurance the class $c \in C$ that maximizes the probability $P(c|f_1, f_2, \ldots, f_k)$, where f_i are features. Following the Bayes rule, $P(c|f_1, f_2, \ldots, f_k)$ can be rewritten as

$$P(f_1, f_2, \ldots, f_k|c) \cdot \frac{P(c)}{P(f_1, f_2, \ldots, f_k)}, \tag{2.9}$$

where $P(c)$ and $P(f_1, f_2, \ldots, f_k)$ are prior probabilities of $c \in C$ and feature configuration, respectivelly. Though, $P(f_1, f_2, \ldots, f_k)$ for large k is typically impossible to estimate, therefore we assume f_i independent (*naive* Bayes) [63]. In the NER domain, the naive Bayes approach was applied by Nobata and Tsujii [49].

2.3 Overview of Gene Interaction Extraction

2.3.1 Task Overview

Gene interaction extraction may be seen as a complete set of tasks. In the BioCreAtIvE II competition four subtasks were distinguished by the organizers [19]:

1. *interaction article subtask*: ranking the PubMed abstracts, based on whether they are relevant for protein interaction annotation;

2. *interaction pair subtask*: extraction of binary protein-protein interaction pairs from the full-text articles;

3. *interaction method subtask*: extraction of the interaction extraction method used to characterize the extracted interactions;

4. *interaction sentence subtask*: retrieving the textual evidence passage describing the interaction.

2.3.2 Computational Linguistics-Based Approaches

Shallow Parsing Approaches

Shallow or *partial parsing* provides only partial decomposition of the sentence structure [66]. Hammerton et al. summarize the *shallow parsing* procedure in three subsequent steps [30]:

1. *part-of-speech tagging*: each word is assigned a morphosyntactic class referred to as *tag* (e.g. *verb, noun*, etc.);

2. *chunking*: tagged words are grouped into non-overlapping *chunks* (e.g. *verb phrase*, *noun phrase*, etc.);

3. *relation finding*: relations of chunks to the main verb are found (e.g. *subject, object*).

To extract gene interactions, a set of *relation patterns of interest* are often defined and applied [39]. The *finite state automata* appear to be a useful modelling tool for these relation patterns [66]:

the *nodes* of the underlying graph represent states, one of which is the *accept* state, and the *directed edges* specify transit into another state according to the next encountered element class. Reaching the *accept* state means that the relation (∼ interaction) captured by the automat has been identified in the given sentence [39, 66]. Relations are typically indicated by prepositional and conjunctional expressions, special attention is paid to verb nominalizations (e.g. *regulate sth.* → *regulation of sth*). Two examples of such approach will be given.

Pustejovsky et al. [52] construct a parse tree from each sentence, using five separate finite state automata, each of them operating on different level in the relational hierarchy. Starting from ground up, these levels involve (1) noun chunking, (2) non-prepositional noun chunks and relation chunks, (3) coordinated chunks, (4) *of*-prepositional phrases, and (5) subordinated clauses. Relation of interest are sought both on nominal and sentence level; the relation extraction proceeds as identification of arguments and relation elements.

The approach described by LEROY ET AL. [39] uses closed class English words (namely prepositions, negation elements, conjunctions) to determine the structure of biomedical texts. Due to semantic stability and essential role in controlling relations between individual entities, these words are used as the core of abstract relation templates represented by finite state automata. Four automata, one for capturing basic sentence structures and three for detecting structures built around three English prepositions, are employed (separately or chained) to extract interactions from preprocessed text. Moreover, they are combined with heuristic principles for handling negation and word coordinations.

Deep Parsing Approaches

In contrast to the *shallow parsing*, the *deep* or *full parsing* considers the entire structure of the sentence [66]. Depending on whether the underlying grammar is extracted manually or automatically, *deep parsing approaches* can be divided into (1) *rationalist methods* and (2) *empiricist methods* [66].

Rationalist methods. The *rationalist methods* employ various grammar formalisms, including combinatory categorial grammars, context-free grammars or link grammars [66]. Two distinct example systems will be briefly introduced.

AHMED ET AL. combine a full parse with linguistically sound rules. First, sentence is split into simple clauses, which are assigned a typed syntactic structure, i.e. a characteristic set of labelled links connecting word pair; these links (∼ syntactic roles) are obtained from a link parser. The interaction extraction complies with the following procedure: starting gradually by subject, verb and object/modifier, the algorithm follows the role links until all predefined, linguistically relevant structures expressed by the link grammar syntactic roles are detected.

FRIEDMAN ET AL. [22] define a large set of semantic classes associated with actions, processes and other relations and extract relations in the form of frames (∼ semantic patterns) consisting of *type*, *value*, possibly followed by another frame; e.g. the result from sentence *Raf-1 activates Mek-1* is *[action,activate,[protein,Raf-1],[protein,Mek-1]]*. In the interaction extraction task, this semantic/syntactic grammar is combined with a grammar parser.

Empiricist methods. Many systems based on *empiricist approaches* make use of some kind of hidden Markov models (see section 2.2.5, page 12) [66]. Skounakis et al. [58] use parse trees generated by a shallow parser to construct the input representation for the hidden Markov models (HMMs). The concept of the HMMs had to be adjusted to this specific representation, which requires HMMs to accept grammatical information at multiple scales.

Another approach builds on *dependency parse trees.* In the RelEx system proposed by Fundel et al. [25], first *dependency parse tree* is generated from the given sentence, then simplified *noun-phrase chunk dependency tree* is constructed. Finally three rules are applied for the respective identification: *effector-relation-effectee*, *relation-effectee-by-effector* and *relation-between-effector-and-effectee.*

2.3.3 Rule-Based Approaches

Rule-Based Approaches

Rule-based systems for extracting gene interactions employ manually or automatically generated textual rules or patterns encoding relationships between entities [66]. According to [66] rule-based approaches suffer from insufficient portability to other domains and inability to successfully process more complicated statements.

In the system proposed by Blaschke and Valencia [4], separate sentences are matched against a list of predefined frames representing templates for gene interactions, reliability scores for both frames and detected interactions are computed based on distance and frequency measures. Proux et al. [55] combine shallow parsing with a knowledge processing approach: syntactic dependencies detected by a shallow parser are used to construct a dependency graph, which are searched for predefined generic request scenarios, such as *gene interacts with gene*, *gene acts as modifier of gene*, *gene induces the expression of gene product* etc.

Systems capable of defining rules automatically have been proposed e.g. by Huang et al. [33] or Phuong et al. [50]. Huang et al. [33] use dynamic programming to extract frequent patterns which meet specific structural requirements. The system operates on part-of-speech level; parts-of-speech representing modifiers (adjectives, adverbs etc.) and determiners are removed from tagged sentences, since they lower the generalization power of candidate rules; three filtering rules are defined to control the structure of resulting rules. Phuong et al. [50] work with link-parsed sentences. Starting from each keyword of any annotated interaction, the shortest link paths to all annotated gene words are found; thus the link paths represent the rules describing the interaction structure. The set of specific rules is then passed to a generalization algorithm producing more generic rules. Hakenberg et al. [29] generate patterns by sentence clustering and multiple sentence alignment.

2.3.4 Machine Learning and Statistical Approaches

This class of methods includes statistical term co-occurrence analysis [66], bayesian classifiers, support vector machines and hidden Markov models (section 2.2.5, page 10).

Kraven and Kumlien [13] use Naive Bayes to classify sentences according to whether they contain gene interaction or not. Documents are represented as *bags of words*, i.e. the word positions are considered not to be important. Stapley and Benoit [59] start from the premise, that two genes co-occur more frequently in biomedical literature, if they have a related biological function. Assuming a fixed set of gene entities, every gene pair is investigated for co-occurrence, both joint and individual occurrence statistics are used to compute a (dis)similarity distance between the involved genes. Donaldson et al. [15] employ support vector machines to distinguish those biomedical abstracts describing gene interactions. The feature set is constructed from single words and two-word sequences with the highest positive information gain.

Bunescu and Mooney [6] designed a general text mining method consisting in construction of a specific kernel which is passed to SVM classifier. The approach as built around a hypothesis that the relation between two entities can be estimated from the shortest path between them in the dependency graph. Using both words and various automatically generated metalingual generalizatios as features, the value of kernel function is computed as a number of common features between two relations. Airola et al. [1] adopt this approach specifically for the gene interaction extraction task: the kernel (graph kernel) is constructed by employing the word sequence order and syntactic information from the sentence parse.

Chapter 3

Method Description

3.1 Instrumentarium

Before introducing the ideas of the designed text preprocessing, we provide definitions of several key terms used throughout the remaining text. Terms specific to individual components will be explained at corresponding places.

Word. Tag. Tag class. We define *word* as a character sequence delimited by space or any delimiter. Grammar *tag* is a code of metalingual class assigned to the word according to its morphological, syntactical and semantical properties. The *tags* used in this project originate from the Penn Treebank Tagset [62], which has been extended by new *tags*. Similar *tags* constitute *tag classes*: *noun*={*NN, NNS, NNP, NNPS, GENE*}, *verb*={ *V[BHV], V[BHV]Z, V[BHV]P, V[BHV]D, V[BHV]N, V[BHV]G, MD*} etc.

Gene entity word. Ordinary word. We define *gene entity word* as a gene name which needs to be invariant to all transformations applied to the word sequence. *Ordinary word*, in opposite, is a word which is not a *gene entity word*.

Verbal noun. Adjectival noun. *Verbal nouns* are nouns derived from verbs, in context of this project only those expressing action: *-tion*, *-sion*, *-ment*, *-age*, *-al*, *-ence*, *-ance*, *-ery*, *-ry* (e.g. *activation, expression*). *Adjectival nouns* are nouns derived from adjectives by adding *-ment*, *-ness*, *-ity*, *-ance*, *-ence*, *-ency*, *-ship* or *-hood* (e.g. *capability, efficiency*).

Interaction kernel. Interaction operands. We define *interaction kernel* as a word which binds together two *gene entity words* as *interaction operands*. The *interaction operand* can be either *agent operand* (agent $\sim$ affecting) or *patient operand* (patient $\sim$ affected). Semantically, the *interaction kernel* is the predicate of the interaction: it expresses, how the *agent operand* affects the *patient operand* (transitive relation) or how two agents *operands* change their common state (intransitive relation). The *interaction kernel* appears most commonly in the form of verb, verbal noun or adjective; these types will be referred to as *verb kernel*, *noun kernel* and *adjective* *kernel*; *noun* and *adjective kernels* are both *nominal kernels*. Note that *interaction kernel* (∼ predicate) may require prepositions to bind *interaction operands* (∼ arguments). Nominal verb forms without auxiliary verb (ex. 5) are treated as *nominal* rather than *verb kernels*, since they appear within nominal phrases.

(1) G1@GENE[agent] activates@VVZ[verb kernel] G2@GENE[patient]

(2) G1@GENE[patient] activated@VVN[verb kernel] by@IN G2@GENE[agent]

(3) G1@GENE[agent] activation@NN[noun kernel] of@IN G2@GENE[patient]

(4) G1@GENE[patient] activation@NN[noun kernel] by@IN G2@GENE[agent]

(5) G1@GENE[agent] -activated@JJ[adjective kernel] G2@GENE[patient]

3.2 Reducing Cluster Complexity

3.2.1 Problem Specification

(1)

G
the G gene
the G gene expression
the G gene expression in the cell
the activation of the G gene expression in the cell
the activation of the G gene expression in the eucaryotic cell

The above example demostrates that altering a simple phrase by adequate language components causes the phrase to grow both to the left and to the right. Even though there are limitations of such growth given by the demand of understandability, i.e. the communicational dimension of the language prevents phrases by stylistic rules from growing throughout arbitrarily, the space of all possible forms of phrases remains infinite. We name this problem *arbitrary phrase space complexity*. To address another problem assume the following example sentences:

(6) G1@GENE activates@VVZ G2@GENE

(7) the@DT expression@NN of@IN G1@GENE activates@VVZ G2@GENE

In sentence 6 *G1* binds to predicate *activates* (i.e. to the right), whereas in sentence 7 *G1* binds to verbal noun *expression* (i.e. to the left). As a result, the distance between *G1* and *activates* in sentence 6 is not equal to the distance between the same elements in sentence 7. We name this problem *arbitrary element directionality*.

Thus, being given a task to investigate the semantic relation between two lexical elements of such a sequence, we have to face two difficulties as a result of the problems discussed above: (1) the distance between the assumed elements may be arbitrarily long; (2) this distance is hard to estimate.

3.2.2 Linguistic Observations

Consider the following simple noun phrase:

(8) gene@NN G@GENE in@IN cells@NNS

In this phrase, *G* is the *head* of the phrase, i.e. the word holding the core meaning, *gene* stands for the *attribute*, i.e. word preceding the *head* word, and *in cells* is the *appositional adjunct*, i.e. word sequence following the *head* word. The importance of the *head* of the phrase comes from its ability to represent the whole phrase without fatal change in the meaning. Thus the word *G* can replace the whole phrase in the given sentence and simplify its structure. Now assume another noun phrases:

(9) mouse@NN cell@NN gene@NN G@GENE

(10) mouse@NN cell@NN gene@NN G@GENE investigated@JJ in@IN our@PP$ experiments@NNS

Similarly to the phrase 8, *G* stands for the *head* of the phrase in examples 9 and 10. Notice that the *head* word of English noun phrases is always the last noun before the *appositional adjunct* or the last word of the phrase, if no *appositional adjuncts* are present. By gradually removing the attributes and appositional adjuncts all three phrases presented above can be reduced to the single-word form *G*. As we are primarily interested in gene names, this result is highly valuable: instead of the gene name nested in a complicated word structure we now operate with an almost equivalent single word element without paying anything for such a reduction.

However, not all noun phrases containing a gene name can be replaced by this gene name without non-zero cost. Assume the following two noun phrases:

(11) G@GENE expression@NN in@IN cells@NNS

(12) expression@NN of@IN G@GENE in@IN cells@NNS

In contrast to phrases 8 through 10, the gene name *G* does not stand for the *head* in phrases 11 and 12, thus the simple removal of attributes and appositional adjuncts would result in losing the gene name, element of our interest. To avoid such a loss, we need to propagate the gene name into the head of the phrase. In the phrase 11, we propagate *G* to the right by removing *expression*, whereas in the phrase 12 we propagate *G* to the left by replacing *expression of G* by *G*. I is obvious, that such changes cause a shift in the semantic structure of the phrase. However, these shifts are linguistically measurable, and therefore controllable.

Similar transformations can be applied to verb sequences (not verb phrases); the example 13 demonstrates the growing complexity of verb sequences. It can be shown that despite the arbitrary length and complexity of the verb sequences, the core meaning is always held by the last verb of such sequences.

(13) activates@VVZ; has@VHZ been@VBN activated@VVN; has@VHZ been@VBN said@VVN to@TO activate@VV

Finally, notice that words may show strong affinity to the preceding word (ex. 14) or to the following word (ex. 15).

(14) of@IN G@GENE (G shows affinity to the left)

(15) G@GENE activation@NN (G shows affinity to the right)

3.2.3 Main Principles

The methodological approach of resolving the problems identified in section 3.2.1 complies with the following general principles.

Sentence structure reduction. The language sentence may be considered as a projection of a multidimensional, non-sequential language structure into a sequence of lexical elements. As shown in section 3.2.1, the backward mapping (i.e. word sequence interpretation) may be extremely difficult without fully qualified language knowledge. The objective is to modify the language structure behind the original lexical sequence so, that it is more reliably mirrored by the new lexical sequence resulting from a projection of the modified structure into a sequence of lexical elements. The presented linguistic observations suggest that such a modified sentence structure may be retrieved by ruled sentence simplification. The resulting structure is called *sentence skeleton*, since it is supposed to hold the core meaning of the original sentence. Accordingly, the process of deriving sentence skeleton is called *sentence skeletonization*.

Operation atomicity. To achieve the sentence structure reduction, we do not work with the sentence as a whole, since we would need to face the potential complexity of a general sentence. Instead of that, we concentrate on low-level, elementary transformations with the simplifying effect, i.e. we rely on *what we almost certainly know about the language components contained in the word sentence and their relations*. By applying repeatedly a set of elementary transformations, we gradually get a sentence with higher sequentiality factor. As the transformations are linguistically relevant, we can also qualify, quantify and register the additive semantic shifts (smaller or bigger) caused by these transformations, which enables us to define a distance measure *the overall semantic shift*, the sum of costs required by the given transformations.

Gene name propagation. Simplifying a word sequence can not proceed without removing words, which are considered irrelevant from the general syntactical and semantical point of view. However, our task requires us to work with gene and protein names; what if a gene or protein name holds such a position, which should be considered irrelevant from that general point of view? The fact that we concentrate on a specific semantic subclass of nouns does not imply, that the entities of our interest are equally important in the sentence structure. By simply allowing for removal of gene and protein names from irrelevant positions, we would suffer information loss with fatal consequences. To avoid this, we need to propagate the entities of our interest to more stable positions. However, this proceeding causes non-negligible shift in the semantic space of the given sentence. To control the resulting semantic shift, we use the distance measure defined in the previous paragraph.

Proximity assumption. Due to declared operation atomicity, the word sequence is never seen as a whole, but always locally. As a result, conjunctional elements may be ambiguous: since we are given only the immediate neighborhood, we are not capable of determining, what subsequences of the sentence actually constitute the arguments of the conjunctional element. However, in case that both left and right neighboring words are of the same or related tag class, the following principle is applied: unless there is special reason for not treating them as arguments of the conjunctional element, they are treated as such.

Assumption of stylistic correctness. Clearly ambiguous sentences remain at least equally ambiguous after being skeletonized. However, we assume that texts written by experts are stylistically correct, and therefore not ambiguous.

To anchor the suggested method into the context of common methodologies introduced in section 2.3 (page 13), the following needs to be said: The skeletonization resides somewhere between rule-based approaches and approaches based on shallow parsing: it comprises of a small set of predefined rules operating on the micro-syntactic level, which are employed to reveal a very limited set of structures on the macro-syntactic level. Allowing such incompleteness of the output macro-syntactic information results from the observation, that components constituting the grammatical (and also semantical) content of the given sentences are not equally relevant for extracting a highly specific type of information.

3.2.4 Instrumentarium

Additional terms have to be explained to conceive the ideas of resolving specifically the problems identified in section 3.2.1. The structural units defined in this section are built around similar concepts used in the shallow parsing methodologies, especially in [52].

Cluster. Minimal cluster. We define *cluster* as a sequence of words of the same *tag class* (*noun class* or *verb class*), either without arguments or with non-prepositional arguments: the *noun cluster* contains primarily nouns, but also adjectives, numerals, adverbs or determiners (ex. 16); the *verb cluster* contains primarily verbs, but also adverbs and prepositions (ex. 17). A *cluster* consisting of only one word is called *minimal cluster*.

Cluster sequence. Head cluster. We define *cluster sequence* as a set of subsequent *clusters* of the same class separated by prepositions or conjunctions (ex. 18). Note that clusters sequence is not equal to *phrase*. Furthermore, the *head cluster* is a *minimal cluster* which is not prepositional argument of any other *cluster* of the given *cluster sequence* and which holds the core meaning of the whole *cluster sequence*.

Cluster head, cluster prefix. We define *cluster head* as a word at the right-most position in the given *cluster*. The *head* word holds the core meaning of the whole *cluster*. Consequently *prefix* of the *cluster* is the word or *cluster* of the same class preceding the *cluster head*. *Cluster prefix* shrinks the semantic space covered by the *cluster head*.

(16) the@DT very@RB active@JJ gene@NN G1@GENE

(17) has@VHZ been@VBN proven@VVN to@TO activate@VV

(18) activation@NN (head cluster) of@IN [G2@GENE] and@CC [G3@GENE] in@IN [mouse@NN cells@NN]

(19) cell@NN gene@NN

Left gene pole. Right gene pole. We say that a *gene entity word* has *left pole* (LP) if it has a strong affinity to the nearest left neighboring word. *Left pole* appears in case that a *gene entity word* follows a preposition (ex. 20) or adjective (ex. 22). Similarly, we say that a *gene entity word* has *right pole* if it has a strong affinity to the nearest right neighboring word. *Right pole* appears in case that a *gene entity word* precedes an adjective (ex. 22) or *verbal noun* (ex. 21).

(20) of@IN G@GENE(LP)

(21) G@GENE(RP) activation@NN

(22) G1@GENE(RP) activated@JJ G2@GENE(LP)

3.2.5 Noun Cluster Sequence Reduction

Reduction of *noun cluster sequences* is achieved using the following four elementary transformations.

Left removal. Assume a two-word *cluster*, where both *prefix* and *head* positions are held either by *ordinary* words (ex. 23) or by *gene entity* words (ex. 25), or where the *prefix* position is held by an *ordinary* word and the *head* position by a *gene entity* word (ex. 24). We define *left removal* as removal of the word at the *prefix* position. The shift in semantic space is considered negligible, therefore *cost* = 0.

(23) cell@NN gene@NN → gene@NN

(24) gene@NN G{0,0}@GENE → G{0,0}@GENE

(25) G1{0,0}@GENE G2{0,0}@GENE → G2{0,0}@GENE

Forward propagation. Assume a two-word *cluster*, where the *prefix* position is held by a *target* word and the *head* position by an *ordinary* word, which is not required as a *special separator*. We define *forward propagation* as removal of the word holding the *head* position, i.e. moving the *target* word to the *head* position (ex. 26). The *forward propagation* causes a non-negligible shift in the semantic space, therefore *cost* = 1.

(26) G{0,0}@GENE activation@NN → G{1,0}@GENE

Right removal. Assume two *minimal clusters* connected by a preposition, where both *clusters* are represented by words of the same importance class or where the left *cluster* is represented by a *target* word and the right cluster by an *ordinary* word. We define *right removal* as removal of

Original pattern	Reduced pattern	Example sequence
gene1{0,0} nv IN gene2{0,0}	(gene1{1,0},gene2{0,2})	G1 activation of G2
nv IN gene1{0,0} IN gene2{0,0}	(gene1{0,2},gene2{0,2})	activation of G1 by G2
nv between gene2{0,0} and gene1{0,0}	(gene1{0,2},gene2{0,2})	interaction between G1 and G2
gene1{0,0} JJ gene2{0,0}	gene2{0,0}	G1 -induced G2
gene1{0,0} JJ-ing gene2{0,0}	gene1{0,0}	G1 inducing G2
gene1{0,0} JJ IN gene2{0,0}	gene1{0,0}	G1 required for G2

Table 3.1: Reduction of specific nominal structures. Legend: nv ∼ verbal noun, JJ-ing ∼ *ing*-form in role of adjective.

the right *cluster* and the connecting preposition (ex. 27). The shift in the semantic space caused by the *right removal* is considered negligible, therefore *cost* = 0.

(27) G{0,0}@GENE in@IN cell@NN → G{0,0}@GENE

Backward propagation. Assume two *minimal clusters* connected by a preposition, where the left *cluster* is represented by an *ordinary* word not required as a *special separator* and the right *cluster* is represented by a *target* word. We define *backward propagation* as removal of the left *cluster* and the connecting preposition, i.e. moving the right *cluster* to the position of the left *cluster* (ex. 28). The shift in the semantic space is considered significant, therefore *cost* = 2.

(28) activation@NN of@IN G{0,0}@GENE → G{0,2}@GENE

Adjectives are removed from noun *clusters*, unless they are considered to be *kernel* candidates. Moreover, *nominal kernels* build together with gene intities a limited number of specific structures, which may be patternalized and reduced in the way shown in table 3.1.

3.2.6 Verb Cluster Reduction

Assume a verb cluster consisting of at least two words, where the *head* position is held by a verb and the *prefix* position is held by single verb (ex. 29), couple verb + *to* (ex. 30) or triples *be* + adjective + *to* (ex. 31) or *have* + noun + *to* (ex. 32). We define *verb reduction* as removal of all words at the *prefix* position. The shift in semantic space is considered negligible, therefore we do not count any cost.

(29) have@VHP activated@VVN → activated@VVN

(30) known@VVN to@TO activate@VV → activate@VV

(31) is@VBZ able@JJ to@TO activate@VV → activate@VV

(32) has@VHZ ability@NN to@TO activate@VV → activate@VV

Adverbs are removed except for the negation element *not*, which does not allow full *verb reduction* (ex. 33).

(33) have@VVP not@RB activated@VVN → have@VVP not@RB activated@VVN

3.2.7 Appositions and Coordinations

Appositions and coordinations are another factors contributing significantly to the sentence complexity. They are reduced using a specific rule set.

Appositions. Assume a couple of *minimal noun clusters*, the second of which is separated from both sides by a delimiter. Being both *clusters* represented by ordinary noun, the first one is selected to replace the appositional structure (ex. 34); gene entity is, however, always preferred to *ordinary* noun (ex. 35); being both *clusters* gene entities, the appositional structure is replaced by a concatenation of both of them (ex. 36). Since both *clusters* are coordinated, we do not count any cost.

(34) gene@NN ,@, protein@NN ,@, → gene@NN

(35) gene@NN ,@, G@GENE ,@, → G1@GENE

(36) G1@GENE ,@, G2@GENE ,@, → G1,G2@GENE

Coordination. Assume a couple of *minimal noun clusters* separated by a delimiter. To reduce such a structure, we apply similar rules as used with appositions (ex. 37, 38). Gene entities are, however, concatenated only if they have not opposite *poles* (*proximity assumption, ex. 39*), otherwise no change is applied (ex. 40).

(37) gene@NN and@CC protein@NN → gene@NN

(38) G@GENE and@CC gene@NN → G@GENE

(39) G1@GENE and@CC G2@GENE → G1,G2@GENE

(40) of@IN G1@GENE(LP) and@CC G2@GENE(RP) [inhibition@NN] → [of@IN] G1@GENE and@CC G2-@GENE [inhibition@NN]

3.3 Clause Skeleton

3.3.1 Example Derivation

We demonstrate the *skeleton* idea on an example sentence. Note that the example sentence is intentionally complex, since we want to demonstrate the general case; by selecting a simple example some operations would seem needless.

(41) the@DT expression@NN of@IN G1@GENE gene@NN and@CC the@DT G2@GENE induction@NN of@IN gene@NN G3@GENE in@IN B-cells@NPS proved@VVD recently@RB to@TO activate@VV the@DT G4@GENE phosphorylated@JJ genes@NN G5@GENE and@CC G6@GENE

Redundant words (determiners, adverbs) are removed from the sequence, *noun* and *verb clusters* are marked for easier orientation:

(42) [expression@NN] of@IN [G1@GENE gene@NN] and@CC [G2@GENE induction@NN] of@IN [gene@NNS G3@GENE] in@NN [B-cells@NPS] [proved@VVD to@TO activate@VV] [G5@GENE phosphorylated@JJ genes@NN G5@GENE] and@CC [G6@GENE]

Left removal and *forward propagation* are applied leaving candidate *nominal kernels* untouched. Furthermore, poles of gene entities are determined:

(43) [expression@NN] of@IN [G1{1,0}@GENE(L)] and@CC [G2{0,0}@GENE(R) induction@NN] of@IN [G3{0,0}@GENE(L)] in@IN [B-cells@NPS] [proved@VVD to@TO activate@VV] [G4{0,0}@GENE(R) phosphorylated@JJ G5{0,0}@GENE(L)] and@CC [G6{0,0}@GENE]

Appositions and *coordinations* are resolved. Since *G1* and *G2* have opposite *poles*, they are not considered coordinated. In contrast, *G5* and *G6* are coordinated:

(44) [expression@NN] of@IN [G1{1,0}@GENE(L)] and@CC [G2{0,0}@GENE(R) induction@NN] of@IN [G3{0,0}@GENE] in@IN [B-cells@NPS] [proved@VVD to@TO activate@VV] [G4@GENE{0,0}(R) phosphorylated@JJ G5{0,0},G6{0,0}@GENE(L)]

Notice that the resulting sequence allows for extraction of all nominal interactions contained in the precessed clause: *G2@GENE induction@NN of@IN G3@GENE* and *G4@GENE phosphorylated@JJ G5,G6@GENE*. We name this sequence *nominal skeleton*.

Now assume we are seeking for interactions built around *verb kernel*. First, verb structure is reduced:

(45) [expression@NN] of@IN [G1@GENE{1,0}(L)] and@CC [G2@GENE{0,0}(R) induction@NN] of@IN [G3{0,0,}@GENE] in@IN [B-cells@NPS] [activate@VV] [G4@GENE{0,0}(R) phosphorylated@JJ G5{0,0},G6{0,0}@GENE(L)]

The verb *activate* requires two arguments, one on the left, one on the right side, i.e. *cluster sequences* preceding and following the verb need to be reduced. Complex structures are resolved in preference according to table 3.1:

(46) [expression@NN] of@IN [G1@GENE{1,0}(L)] and@CC [G2{1,0},G3{0,2,}@GENE] in@IN [B-cells@NPS] [activate@VV] [G5{0,0},G6{0,0}@GENE(L)]

The *right removal* and *backward propagation* are applied:

(47) [G1@GENE{1,2}] and@CC [G2{1,0},G3{0,0}@GENE] [activate@VV] [G5{0,0},G6{0,0}@GENE]

Again, *coordinations* are resolved; this time, however, regardless of the gene *poles*:

(48) [G1{1,2},G2{1,0},G3{0,0}@GENE] [activate@VV] [G5{0,0},G6{0,0}@GENE]

The resulting sequence allows for extraction of the interaction built around a verb kernel. Therefore, we name this sequence *verb skeleton*.

The process of finding the clause skeletons can be roughly summarized into four steps: (1) reduce *noun clusters* into *minimal clusters* using the *left removal* and *forward propagation*; resolve *appositions* and *coordinations* according to the gene *poles*, which results to a *nominal skeleton*. The remaining two steps are specific to *verb skeletons*: (3) resolve nominal structures, mainly using the *right removal* and *backward propagation*; (4) resolve *appositions* and *coordinations* regardless of the gene *poles*. Following the path of abstraction, the above four steps may be further summarized in two steps: (I) investigate in details the internal structure of *noun cluster sequences*; (II) reduce the *noun cluster sequences* (if possible) to such forms which can be passed as arguments to clause verb predicate.

3.3.2 Nested Nominal Kernels

Nominal structures are resolved and passed as arguments to clause predicates, i.e. nominal structures are subordinated to verb predicates. However, *nominal kernels* are also predicates, i.e. they bind arguments: nouns, *gene entity words* or other nominal predicates. Nominal structures built around *nominal kernels* are saved in nominal skeletons before they are dissolved to become verb arguments. We need to define a storage device, where to save subordinated nominal structures built around *nominal kernels*, before they are dissolved to become arguments of their superior nominal predicates.

To resolve this problem the following procedure is applied: (i) nested nominal structures are detected - they are indicated e.g. by sequences adjective + verbal noun 49 or by *gene entity words* having both left and right *poles* 53; (ii) mark candidate *kernels* either as superior (the left one) or subordinated (the right one); (iii) subordinated structures are reduced to a *minimal cluster* (table 3.1) leaving superior structures untouched (first skeleton)a; (iv) superior structures are removed leaving subordinated structures untouched (second skeleton). Note also, that when creating verb arguments, first subordinated structures and then superior structures are reduced.

We demonstrate this procedure on two simple examples. In example 49 *noun kernel* is an argument of an *adjective kernel*:

(49) G1{0,0}@GENE(R) -induced@JJ activation@NN of@IN G2{0,0}@GENE(L) by@IN G3{0,0}@GENE(L)

Following the above steps, candidate *kernels* are detected (ex. 50), first skeleton in derived by reducing subordinated structure (ex. 51), second skeleton is derived by removing superior structure (ex. 52).

(50) G1{0,0}@GENE(R) -induced@JJ(superior) activation@NN(subordinated) of@IN G2{0,0}@GENE(L) by@IN G3{0,0}@GENE(L)

(51) G1{0,0}@GENE -induced@JJ G2{0,2},G3{0,2}@GENE

(52) activation@NN of@IN G2{0,0}@GENE by@IN G3{0,0}@GENE

In example 53 an *adjective kernel* is an argument of a *noun kernel*:

(53) G1{0,0}@GENE activation@NN of@IN G2{0,0}@GENE -induced@JJ G3{0,0}@GENE

Following the above steps, candidate *kernels* are detected (ex. 54), first skeleton is derived by reducing subordinated structure (ex. 55), second skeleton is derived by removing superior structure (ex. 56).

(54) G1{0,0}@GENE(R) activation@NN of@IN G2{0,0}@GENE(L,R) -induced@JJ G3{0,0}@GENE(L)

(55) G1{0,0}@GENE(R) activation@NN of@IN G3{0,0}@GENE(L)

(56) G2{0,0}@GENE(L,R) -induced@JJ G3{0,0}@GENE(L)

The above procedure assumes two-level nominal structures, i.e. once being subordinated, nominal kernel does not take another nominal kernel as its argument. Nominal structures of higher order are not only improbable, but they also do not meet the requirement of understandability, i.e. they considered stylistically incorrect. To confirm this statement, a number of queries were made in a large set of biomedical abstracts.

3.3.3 Skeleton Tasks

Consider that we are given a sentence containing an interaction built around a *verb kernel*. The expected structure of the sentence is as follows:

- *noun cluster sequence* containing *gene entity word G1*,
- *noun cluster sequence* containing *gene entity word G2*,
- *verb cluster*,
- word environment surrounding the three above components.

The interaction requires two *operands* and an *kernel*. The required transformations may be defined in the following manner:

- *noun cluster sequence* containing *gene entity word G1* → *head cluster* (*gene entity word* only),
- *noun cluster sequence* containing *gene entity word G2* → *head cluster* (*gene entity word* only),
- *verb cluster* → *minimal cluster*.

Consider that we are given a sentence containing an interactions built around a *nominal kernel*. The expected structure of the sentence is as follows:

- *nominal cluster sequence* containing the *gene entity word G1*, *gene entity word G2* and a *nominal kernel*,
- environment surrounding the component above.

The interactions requires two *operands* and a *kernel*. The required transformation may be defined in the following manner:

- *nominal cluster sequence* → sequence containing *nominal kernel*, whose arguments are *minimal nominal clusters*.

3.4 Resolving Language Pointers

3.4.1 Problem Specification

To address another problem regarding the sequential text mining task first assume the following sentence triple:

(57) it@PP [G1] activates@VVZ G2@GENE

(58) its@PP$ [G1] activation@NN of@IN G2@GENE

(59) which@WDT [G1] activates@VVZ G2@GENE

Our observations will focus on pronouns contained in each of the three above sentences (*it*, *its* and *which*). Pronouns represent a structural element (word, clause) without actually holding its semantic value: being given only the above sentence stubs, we are able to guess the grammatical meaning, but not the semantic contents. Using the pointer/value opposition, pronouns point to the structural element actually holding the semantic value; therefore, we name them *language pointers*. Personal and possessive pronouns (examples 57 and 58) point either to a noun word in any preceding clause (i.e. within the same sentence), or to a noun word in any (but typically close) preceding sentence; relative pronouns (example 59) point most typically to the last word of the preceding clause. Hence, predicate argument is not represented by any element holding the semantic value, but rather by a pointer referring to this element. Now consider one more sentence stub:

(60) and@CC [G1] activates@VVZ G2@GENE

In example 60, the subject argument is represented neither by element holding the semantic value, nor by pointer referring to that element. To be more precise, there is a *language pointer* (since transitive verb simply requires two arguments), though not explicit, but rather implicit. We will call it *implicit language pointer*; consequently the personal pronouns, possessive pronouns and relative pronouns will be referred to as *explicit language pointers*.

The difficulty arising from the existence of *language pointers* is obvious: Being given a task to determine, whether two lexical entities build a semantic relation, we first have to resolve the element the pointer refers to. This problem will be named *the existence of language pointers*.

3.4.2 Main Principles

The methodological approach of resolving the problem of language pointers complies with the following general principles.

Mapping language pointers to corresponding values. Treating pointers as values naturally generates errors in sentence interpretation. In case that a pointer stands for a gene entity, we typically miss at least one semantical relation (possibly an interaction) concerning this entity. Substituting pointers by their values naturally avoids this problem, however, it requires a correct mapping of language pointers to their lexical values. In fact, we determine the most probable mapping rather than correct mapping, since this task proves to be very difficult.

Restriction to gene values. As we restrict our interest exclusively to those relations describing gene interactions, we leave unresolved all pointers, which are unlikely to refer to any gene entity. If at least one relation operand is not a gene entity, we do not miss any interaction. This principle is analogy to the gene propagation principle (section 3.2.3, page 20).

Assumption of left pointer orientation. Language pointers may refer both to the left and to the right (i.e. pointers referring to a subject clause). However, we take into account only those pointers pointing to the left and thus following the common textual principle: an entity is referred to (as clause/sentence rheme) not until it has been introduced (as sentence/clause theme).

Operation minimality. In contrast to the cluster simplification task, pointer mapping can not be evaluated using a reliable language based measure, since the context we are working with is too large and therefore too versatile. To minimize the probability of making errors, we only apply a minimum number of steps/transformations to find the most probable values of the given pointer.

3.4.3 Pronoun Mapping

Two restrictions are set for resolving explicit language pointers: (1) Relative pronouns are excluded from the pronoun mapping since they, following contiguously the word they are pointing to, do not actually break the sentence sequentiality. (2) The considerations focus only on those personal and possessive pronouns preceding the clause predicate, i.e. being part of the noun cluster sequence containing the clause subject, since occurrences in noun cluster sequences at object or prepositional positions prove to be very rare.

In the majority of cases both personal and possessive pronouns refer to an entity, which plays a role of subject in one of the previous clauses. Such situations are demonstrated in example 61 and 62 for both pronoun types.

(61) G1@GENE consists@VVZ of@IN three@CD exons@NNS and@CC it@PP (→ G1@GENE) activates@VVZ G2@GENE

(62) G1@GENE consists@VVZ of@IN three@CD exons@NNS and@CC its@PP$ (→ G1@GENE) expression@NN activates@VVZ G2@GENE

In some cases, on the other hand, personal and possessive pronouns refer to an entity, which is employed as object or prepositional verb argument (e.g. *interacts@VVZ with@IN G1@GENE*) in one of the previous clauses. Most typically, however, such entity is mentioned right in the previous clause. Mapping to object is shown in examples 63 and 64 for both pronoun types.

(63) We@PP reported@VVD the@DT phosphorylation@NN of@IN G1@GENE and@CC it@PP (→ G1@GENE) associates@VVZ with@IN G2@GENE

(64) We@PP reported@VVD the@DT phosphorylation@NN of@IN G1@GENE and@CC its@PP$ (→ G1@GENE) association@NN with@IN G2@GENE

Furthermore, ambiguous cases, in which both subject and object (or prepositional argument) of the previous clause are held by *gene entity words*, also appear in biomedical literature (examples 65 and 66). This ambiguity is resolved by mapping the pronoun to the entity holding the subject role, since this interpretation is stylistically (i.e. with respect to sentence understandability) more appropriate in most cases.

(65) G1@GENE activates@VVZ G2@GENE ,@, but@CC it@PP (→ G1@GENE) associates@VVZ more@RBR

strongly@RB with@IN G3@GENE

(66) G1@GENE activates@VVZ G2@GENE ,@, but@CC its@PP$ (→ G1@GENE) association@NN with@IN G3@GENE is@VVZ considered@VVN more@RBR strong@JJ

In all examples above, only singular pronouns are employed. By analogy, we could rewrite all these sentences using only plural pronoun forms, which are mapped to plural gene entities. Note that entity names appear very rarely in plural forms, however, coordinate singular entity names are also considered to be plural entity forms. The problem complexity rises when pronoun and entities have different numerus. To resolve these situations, we define the following principles: (1) singular pronoun is never mapped to plural entity names; (2) plural entity names may be mapped to singular entity names, if no plural entity name is available, since often not all entity names have been marked as entity names by data curators; (3) plural pronouns are preferably mapped to plural entity names.

3.4.4 Verb Coordinations

Coordinate verb with no explicit subject is resolved by making a sentence fork, i.e. by creating a new, shorter sentence, in which all tokens preceding this verb are removed unless finally a verb with an explicit subject is met. Examples 67 through 71 demonstrate this procedure, considering all possible forms of coordinate verbs with no explicit subject: coordinate verbs following a conjunction (ex. 67), comma (ex. 68), gene or noun being an object (ec. 69) or prepositional (ex. 70) argument of the preceding clause verb predicate, and an *ing*-form following a preposition indicating a verb argument[1] (ex. 71).

(67) G1@GENE activates@VVZ G3@GENE and@CC interacts@VVZ with@IN G3@GENE ↪ G1@GENE interacts@VVZ with@IN G3@GENE

(68) G1@GENE activates@VVZ G3@GENE ,@, interacts@VVZ with@IN G3@GENE ↪ G1@GENE interacts@VVZ with@IN G3@GENE

(69) G1@GENE which@WDT activates@VVZ G2@GENE interacts@VVZ with@IN G3@GENE ↪ G1@GENE which@WDT interacts@VVZ with@IN G3@GENE

(70) G1@GENE which@WDT interacts@VVZ with@IN G2@GENE activates@VVZ G3@GENE ↪ G1@GENE which@WDT activates@VVZ G3@GENE

(71) G1@GENE activates@VVZ G2@GENE by@IN associating@VVG with@IN G3@GENE ↪ G1@GENE associating@VVG with@IN G3@GENE

3.5 Sentence Skeleton

In section (page 24) construction of clause skeletons has been discussed (i.e. skeletons on clause level); in this section, skeleton concept is extended to cover whole sentences (i.e. skeletons on sentence level).

Sentence skeleton is created by simply concatenating clause skeletons in the same order, in which corresponding clauses appear in the original sentence, using the same connectives as in

[1] This phenomenon will be explained in section 3.6.4, page 33.

the original sentence. On sentence level, word *clusters* continue to play the role of candidate components of possible interactions, i.e. there is no abstraction towards higher level structures (e.g. *cluster sequences*, clauses). Predications, however, may either stay within one clause or extend over multiple neighboring clauses. To demonstrate this difference, consider the following two example sentences:

(72) G1@GENE interacts@VVZ with@IN G2@GENE ,@, which@WDT also@RB activates@VVZ G3@GENE

(73) G1@GENE interacts@VVZ with@IN G2@GENE ,@, but@CC activates@GENE also@RB G3@GENE

Both two example sentences contain two predications, one staying within the first clause (*G1@GENE interacts@VVZ with@IN G2@GENE*) and one extending to both sentence clauses (built around the word *activates@VVZ*). Predications extending over multiple clauses require *language pointers*, either explicit or implicit, in their implementation: *which@WDT* in example 72 and the implicit pointer in example 73.

In addition to concatenating clause skeletons, sentence skeleton construction involves the pronoun mapping and resolving verb coordinations in the following distribution: nominal skeleton construction is extended only by possessive pronoun mapping, while verb skeleton construction includes resolving verb coordinations and both personal and possessive pronoun mapping.

Note that predications may exceed also over multiple sentences. In this case we are already moving on textual level, which is not covered by the presented system. Similarly to sentence level, predications are implemented using language pointers - *It@PP* in example 74.

(74) It@PP interacts@VVZ with@IN G2@GENE ,@, but@CC it@PP also@RB activates@VVZ G3@GENE

3.6 Keeping Semantical Integrity

3.6.1 Problem Specification

To address one more problem concerning sequential (but not limited to) text mining tasks, consider the following sentence triple:

(75) G1@GENE necessary@JJ for@IN G2@GENE interacts@VVZ with@IN G3@GENE

(76) G1@GENE -activated@JJ G2@GENE interacts@VVZ with@IN G3@GENE

(77) G1@GENE activating@VVG G2@GENE interacts@VVZ with@IN G3@GENE

Starting from example 75, the verb *interacts* splits the above sentence in two parts: subject argument and prepositional argument, both of which are noun clusters. In general, the predicative power of verb allows it to operate as top level node which divides clause in two regions containing arguments of the given verb - either noun clusters or another clauses. However, examples 76 and 77 demonstrate, that verbs may occur also within these regions, behaving similarly to those verbs constituting clause predicate - they bind the same types of arguments using the same syntactic rules: *activated* as past participle and *activating* as *ing*-form.

An error in determining, which verb holds the role of clause predicate, may lead towards loss of the clause/sentence semantical integrity. As a result, we speak about the *problem of clause/sentence integrity*. The difficulty of finding the correct solution for this problem may vary significantly from case to case, from trivial ones to those being extremely difficult.

3.6.2 Main Principles

The methodological approach for ensuring semantical integrity complies with the following general principles:

Mapping to potential interaction kernels. To preserve the clause/sentence integrity, selected nominal verb forms are mapped to potential interaction kernels: verbs (i.e. no change), nouns or adjectives with respect to current local context. The mapping follows rules extracted manually from random subsets of biomedical abstracts.

Minimality. Similarly to language pointers, there is no higher chance to define a reliable language based measure to determine the quality of applied transformations. Therefore, in ambiguous cases we use the most universal one to minimize the probability of errors.

3.6.3 Linguistic Observations

Assume the following example sentence:

(78) G1@GENE necessary@JJ for@IN G2@GENE interacts@VVZ with@IN [G3@GENE complex@NN] and@CC inhibits@VVZ [G4@GENE]

The sentence 78 contains three interacting gene pairs: *G1@GENE and G2@GENE*, *G1@GENE and G3@GENE* and *G1@GENE and G4@GENE*. Now we inject *ing*-forms into sentence 78, replacing *necessary@JJ for@IN* by *activating@VVG*, *complex@NN* by *binding@VVG* and *and@CC interacts@VVZ* by *by@IN inhibiting@VVG*:

(79) G1@GENE activating@VVG [G2@GENE] interacts@VVZ with@IN [G3@GENE] binding@VVG by@IN inhibiting@VVG [G4@GENE]

Despite the applied lexical changes, the semantic structure of the resulting sentence 79 remains very similar to that of sentence 78; all relations between gene entities are preserved, one indirect relation is added: *G3@GENE* is now affected by *G4@GENE*. However, all *ing*-forms are treated as verbs (e.g. by TREETAGGER [56] used in this project), even though some of them play role of another part-of-speech: in sentence 79, *activating* works as *adjective@VVG*, *binding@VVG* as *noun*. Mapping all *ing*-forms to a single part-of-speech type causes information loss, which makes us unable to extract the complete and valid set of semantical relations from the given sentence: in our example, we fail to identify pairs *G1@GENE and G3@GENE* and *G1@GENE and G4@GENE*, and apart from this, we extract incorrect pair *G2@GENE and G3@GENE*. The way out of this problem is to reveal the actual part-of-speech of all *ing*-forms employed in the given sentence, i.e. to map them to *nouns*, *adjectives* and *verbs* (possible predicates):

(80) G1@GENE activating@(VVG → JJ) G2@GENE interacts@VVZ with@IN [G3@GENE binding@(VVG → NN)] by@IN inhibiting@(VVG → VVG) [G4@GENE]

In general, semantical integrity is most typically harmed by nominal verb forms, i.e. *ing*-forms, past participles and infinitives. *Ing*-forms are discussed in section 3.6.4, past participles in section 3.6.5; infinitives are not covered by the presented system.

3.6.4 Resolving *ing*-forms

The overview of *ing*-form resolving is given as a list of characteristic patterns in which *ing*-forms are encapsulated.

Noun/gene + *ing*-form + noun/gene. In case *ing*-form is surrounded by nouns or genes (ex. 81 - 83), we interpret it as an adjective. Considering this triple to be a noun cluster, we define the left-side argument to hold the role of the *head* of the cluster. This is needed to correctly resolve the case, when both arguments are genes (ex. 81); in general case (ex. 82 and 83), this interpretation is not necessarily correct, however, as not two genes are involved, the error is not important.

(81) G1@GENE[head] activating@(VVG → JJ) G2@GENE

(82) protein@NN[head] containing@(VVG → JJ) G1@GENE

(83) G1@GENE[not head] signaling@(VVG → JJ) pathway@NN

Noun/gene + *ing*-form + preposition. The most appropriate solution for this pattern is mapping the *ing*-form to verbal noun (*NN*), as shown in ex. 84. In some rare cases, this interpretation seems linguistically inappropriate, however, it does not lead to errors in detecting correct interacting pairs (ex. 85).

(84) G1@GENE binding@(VBG → NN) in@IN myometrium@NN

(85) G1@GENE interacting@(VVG → NN) with@IN G2@GENE

Adjective + preposition + *ing*-form. This pattern covers two frequent situations shown in examples 86 and 87. The sentence in example 86 represents a *predicative sentence* with *capable* in role of *predicative*. In this case, we simply remove the adjective and preposition, which leads to a compound verb form. In the example 87, on the other hand, only a noun phrase is given, even though *capable* remains semantically *predicative*. By removing the adjective and preposition we obtain the structure of *ing*-form surrounded by two nouns or genes, which is resolved according to the corresponding rule.

(86) G1@GENE is@VBZ capable@JJ of@IN inhibiting@VVG G2@GENE → G1@GENE is@VBZ inhibiting@VVG G2@GENE

(87) G1@GENE[head] capable@JJ of@IN inhibiting@VVG G2@GENE → G1@GENE[head] inhibiting@VVG G2@GENE

Verb + preposition + *ing*-form. In case that the *ing*-form is bound directly to a verb, we only remove the verb and preposition as shown in example 88

(88) G1@GENE competes@VVZ for@IN binding@VVG G2@GENE → G1@GENE binding@VVG G2@GENE

Noun/gene + preposition indicating verb argument + *ing*-form. Some prepositions bind to the sentence predicate rather than to the preceding noun: *by, via, though, upon, after, before.* In context of this paper, we name them *prepositions indicating verb argument*. Being given an *ing*-form following the sequence of noun and such a preposition (ex. 89 and 90), we do not touch this structure. It will be further resolved as coordinate verb.

(89) G1@GENE controls@VVZ DNA@NP synthesis@NN by@IN regulating@(VVG → VVG) G2@GENE

(90) G1@GENE activates@VVZ G2@GENE by@IN interacting@(VVG → VVG) with@IN G3@GENE

Adjectival or phraseologically bound noun + preposition not indicating verb argument + *ing*-form. In case of prepositions which do not indicate verb argument we concentrate on patterns, where the preposition follows either adjectival noun or a noun which constitutes together with some verb a phraseological construct (e.g. *play role*). I this noun is connected to a verb, we leave the structure untouched and resolve it further as coordinate verb (ex. 91 and 92). If it, on the other hand, follows a possessive pronoun, we are working with poorly nominal construct. Since we still do not know if it is possible to map the possessive pronoun to some gene entity, the safest way to resolve the *ing*-form is to map it to verbal noun in the way shown in examples 93 and 94.

(91) G1@GENE shows@VVZ efficiency@NN in@IN activating@(VVG → VVG) G2@GENE

(92) G1@GENE plays@VVZ role@NN in@IN activating@(VVG → VVG) G2@GENE

(93) G1@GENE and@CC its@PP$ role@NN in@IN activating@VVG G2@GENE → G1@GENE and@CC its@PP$ role(activating)@NN in@IN G2@GENE

(94) G1@GENE and@CC its@PP$ efficiency@NN in@IN activating@VVG G2@GENE → G1@GENE and@CC its@PP$ efficiency(activating)@NN in@IN G2@GENE

In other cases *ing*-forms are interpreted as verbal nouns, as demonstrated in examples 95 through 98. It is not necessarily the best interpretation (ex. 97 and 98), however, it does not lead to errors in detecting interacting pairs.

(95) the@DT signaling@(VVG → NN) pathway@NN

(96) active@JJ binding@(VVG → NN) gene@NN

(97) four@CD binding@(VVG → NN) genes@NNS

(98) of@IN preexisting@(VVG → NN) gene@NN

3.6.5 Resolving Past Participles

The grammatical ambiguity caused by past participles is trivial to resolve. After removing redundant words (adverbs, determiners etc.), a simple rule is applied: if the past participle follows a verb form, it constitutes the clause verb predicate (ex. 99), therefore it continues to be a verb;

otherwise it is assigned the adjective class (ex. 100.

(99) G1@GENE was@VVD activated@(VVN → VVN) by@IN G2@GENE

(100) G2@GENE activated@(VVN → JJ) G1@GENE

3.7 Improved Sentence Skeleton

3.7.1 Skeleton Grammatical Patches

Improving semantical integrity is achieved through grammatical patches applied to ambiguous or incorrectly classified grammatical phenomena, i.e. the grammatical interpretation determines the semantical interpretation. Such dependence of language comprehension on the grammar interpretation is specific to machine language interpreters, being far from human mechanism of language comprehension. Skeleton derivation is a custom implementation of such a machine interpreter, therefore it requires grammatical patches to be applied before the concerned grammatical phenomena are used for mining the sentence semantics. Taking this into account, semantical integrity is solved right at the very beginning of the skeleton algorithm. The complete skeleton derivation is summarized in algorithm 1.

Algorithm 1: Deriving sentence skeletons

```
Data: Tagged token sequence
Result: Verb and nominal skeletons
Remove redundant words, make various input corrections, select relevant adjectives, resolve
past participles;
Do left removal and forward propagation with regard to all candidate nominal kernels;
Resolve ing-forms, repeat the previous step;
Register gene poles;
Resolve gene coordinations and appositions with respect to gene poles;
Apply right removal and backward propagation;
Resolve possessive pronouns;
Detect nested nominal structures;
Derive nominal skeletons using algorithm 2;
Derive verb skeletons using algorithm 3;
```

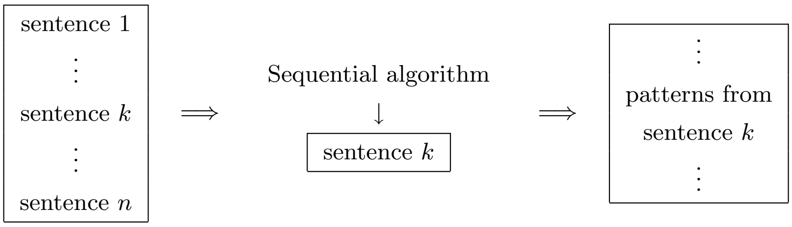
Algorithm 2: Deriving nominal skeletons

```
Data: Output sequence of algorithm 1
Result: Nominal skeletons
Reduce subordinated nominal structures in the input sequence → nominal skeleton 1;
Remove superior nominal structures from the input sequence → nominal skeleton 2;
```

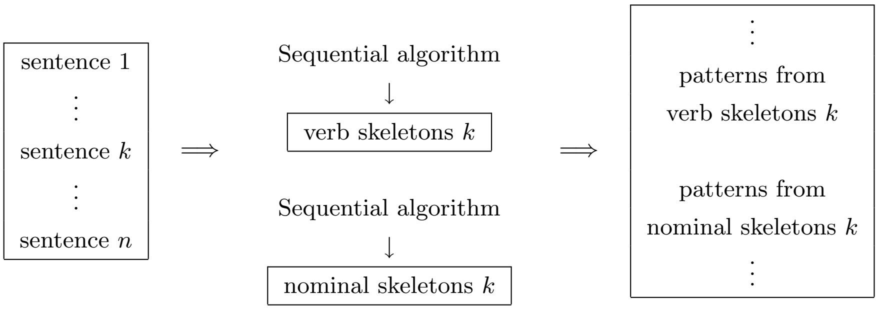
```
Algorithm 3: Deriving verb skeletons
  Data: Output sequence of algorithm 1
  Result: Verb skeletons
  Reduce verb cluster to a minimal cluster;
  Resolve subordinated nominal structures;
  Apply the left removal and forward propagation regardless of candidate nominal kernels;
  Resolve personal pronouns, repeat the previous step;
  Detect genes being objects or prepositional arguments of finite verbs;
  Apply the right removal and backward propagation with respect to gene poles;
  Resolve appositions and coordinations with respect to gene poles → verb skeleton 1;
  for each coordinate verb with no explicit subject do
      Create an additional skeleton by mapping the coordinate verb to the first verb with an
      explicit subject → additional verb skeleton;
```

3.7.2 Skeleton Application

Assume we are given a sequential algorithm which is supposed to be applied on a set of sentences (i.e. language corpus). Instead of sentences, we apply this tool on a set of skeletons. Schemes 3.2 and 3.3 illustrate this paradigm change.



Scheme 3.2: Applying sequential algorithm on sentences



Scheme 3.3: Applying sequential algorithm on sentence skeletons

Chapter 4

Experiments and Results

4.1 Testing Method

A simple sequential approach has been used to evaluate the effect of sentence skeletonization (i.e. improvement of sentence sequentiality) in the gene interaction extraction task: manually created, grammatically relevant patterns representing predication between two gene entities are matched against sentence skeletons, matching subsequences of sentence skeletons are considered to express interactions between the involved gene entities. Two features of this approach are essential:

(I) *Syntagmatic rigidity*: As the resulting sequentiality is the actual target of testing, the reference basis (i.e. what is certainly of sequential nature) represented here by the predefined sequential patterns should mirror the sequential principle in the clearest possible form in order to provide the most informative evaluation. Therefore, the time span between each two subsequent elements of all sequential patterns are set to one, i.e. neighboring tokens of a pattern have neighboring counterparts in the sentence skeleton, no time relaxation is allowed.

(II) *Paradigmatic latitude*: Instead of lexical elements, the sequential patterns are built (almost) exclusively from metalingual components, thus focusing on grammar rather than on the actual semantics. As shown in section 3.6, the grammar, largely employing sequential principles, is often a fundamental prerequisite for semantic integrity; therefore, being forced to keep the pattern set relatively small, such simplification seems acceptable (as approximation). The elements of sequential patterns result from double abstraction: e.g. *noun*-token (i.e. second-level abstraction) of a sequential pattern covers four noun tags (i.e. NN, NNS, NP, NPS; first level abstraction) actually assigned to any English noun word by a language tagger; i.e. any noun may be substituted for the *noun*-token.

The set sequential patterns consists of 29 patterns, 23 with a *verb kernel*, 3 with a *noun kernel* and 3 with an *adjective kernel*, e.g.:

gene verb gene
gene noun preposition gene
gene adjective gene

	AiMed	Brun	Hprd50	IEPA	LLL05	BC-PPI
Sentences	2202	1939	145	486	77	1000
Interactions	1031	2509	148	327	157	294

Table 4.1: Testing corpora. Numbers of interactions may slightly differ from the official ones due to applied normalization principles.

For the complete list of sequential patterns refer to section B, page 53.

4.2 Experimental Data

The resulting sequentialty was evaluated on six biomedical corpora annotated both for gene entites and gene interactions: AiMed [47], Christine Brun Corpus [5], HPRD50 [42], IEPA [3], LLL05 [61] and BC-PPI [28]. The corpus selection was largely inspired by [53] and [14]. Basic features of the testing corpora are given in table 4.1.

All six corpora were handled in the same way acoording to the following four principles: (I) sentences are stemmed and assigned grammar tags using Treetagger [56]; (II) interactions employing more than two gene entities are converted into corresponding number of binary interactions (e.g. one ternal interaction corresponds to three binary interactions); (III) interacting gene pair, being detected in a corpus sentence, is counted only ones into performance measures (precision, recall, F-measure) regardless of how many times it is actually expressed in the sentence; (IV) a triple of two interacting genes and a binding *kernel* is counted only ones in the pattern analysis regardless of how many times it actually appears in the sentence.

4.3 Experiments

4.3.1 Introductory Remarks

The overall performance of the presented approach in terms of *precision*, *recall* and *F-measure* is given in table 4.2. To provide a detailed insight in the actual impact of the sentence skeletonization, to identify its limitations and possible improvements, six experiments have been designed. They are described in the following order: first, analysis of false negatives is given (page 38); second, analysis of false positives is provided (page 39); third, components of sentence skeletonization are studied in cooperation (page 40); fourth, the effect of individual components of sentence skeletonization is discussed (page 41); fifth, performance of the system in dependence on the maximum allowed penalty is analyzed (page 42); sixth, performance of pattern classes and individual patterns are identified (page 43).

4.3.2 False Negatives

False negatives result mostly from the insufficient sequentiality of skeletonized sentences. Two corpora, LLL05 (providing excellent results) and BC-PPI (providing poor results), were analyzed in

	AiMed	Brun	Hprd50	IEPA	LLL05	BC-PPI
Precision	0.49	0.62	0.81	0.74	0.87	0.36
Recall	0.46	0.47	0.61	0.59	0.72	0.65
F-measure	0.48	0.54	0.69	0.65	0.79	0.46

Table 4.2: Precision, recall and F-measure for all testing corpora

	Category	Explanation
1	Incorrect tagging	E.g. *G1 binds@NNS to G2*
2	Distance too long	E.g. multiple nested clauses before interaction is completed
3	Front-end arguments	E.g. *in addition to G2, G1 interacts with G3*
4	Nested *ing*-forms	E.g. *... by activating G2 encoding G3*
5	Higher level non-verb coordinations	E.g. *G1 interacts (with G2) and (with G3)*
6	Unresolved pointers	E.g. *high concentration of G1 induces G2, but low concentration(!) activates G3*
7	Misleading interpunction	E.g. *G1 and G2, interact with G3*
8	Different language forms	E.g. *complex of G1 and G2; G1 and G2 interact [with each other]*

Table 4.3: Analysis of false negatives: unhandled structures, confusing factors

detail to identify both (a) the structures not covered by the sentence skeletonization, and (b) the factors causing the skeletonization to fail to improve the sentence sequentiality. A classification of such phenomena is given in table 4.3. Some of the listed problems could be possibly solved by simply employing a more advanced sequential algorithm (items 5, 7; e.g. algorithm described in [45]), the other require additional preprocessing steps to be included into the sentence skeletonization.

4.3.3 False Positives

False positives result either from (a) shortcommings of the sentence skeletonization, or (b) shortcommings of the sequential algorithm. Similarly to analysis of false negatives, two corpora, LLL05 (giving excellent results) and BC-PPI (giving poor results), have been selected to identify the main factors generating false positives.

(a) Provided that stylistical correctness is guaranteed, the sentence complexity rises together with the complexity of the idea held by this sentence; thus, reducing the sentence complexity naturally distorts the underlying idea. All components of the sentence skeletonization cause errors. However, it proves to be difficult to determine the principal of the fault, since the level of component interconnection is high. The most distinguishable fault contributions are made by components resolving appositions and coordinations (ex. 101), verb coordinations (ex. 102) and *ing*-forms (ex. 103); also pronoun mapping (ex. 104) is well distinguishable fault contributer, though not very

Category	Examples
Mediate view on sth.	*indicate, be observed, be analyzed, reveal, resemble*
Relationship	*correspond to, be related to, be correlated with*
Part of sth.	*be member/subunit of, include, contain, be inserted to, locate, participate in complex*
Role of sth.	*serve as, act as*
External action trigger	*using G1 as G2, examine G1 for [e.g. activating G2]*
Negation	*in absence of, block [e.g. activity of], prevent G1 [e.g. from activating G2]*
Result of an action	*lead to, result from*

Table 4.4: Analysis of false positives: semantically confusing expressions

frequent.

(101) G1 interacts with G3 but G2 not. $\xrightarrow{!}$ G1 interacts with (G3, G2) not.

(102) G1 activates G2, if G2 is present, and also inhibits G3. $\xrightarrow{!}$ G2 inhibits G3. [fork]

(103) G1 activates G2, and G3 activation by binding@(VVG $\xrightarrow{!}$ VVG) G4 inhibits...
→ G1 binding G4. [fork]

(104) The above mentioned gene interacts with G1, and it ($\xrightarrow{!}$ G1) also activates G2.

Furthermore, a destructive multiplicative effect of coordinations has been observed: assuming all coordinations have been correctly resolved, an error in any other component distributes the error to all participants of the given coordination, which may lead to more than five errors in a single sentence. Whether the improvement of the sentence skeletonization would decrease the error rate significantly, is the question for further work.

(b) Errors of the testing algorithm rise mostly from the ommission of semantics: not every word holding the position of a *kernel* is trully a *kernel*. Based on a detailed analysis, several semantic classes frequently confused with interaction *kernels* have been identified; they are available together with examples in table 4.4. The overall performance on various corpora (table 4.2) depends strongly upon the frequency of such confusing *kernel* candidates. This kind errors could probably be to great extent reduced by employing a more advanced sequential algorithm, e.g. the algorithm described in [45] (hypothesis for further work).

4.3.4 Effect of Gradually Adding Components

The impacts of preprocessing components can be hardly additive: since each change in the sentence structure modifies the underlying idea, every other change works with different idea than the previous one. As a result, the particular components constitute a complicated interference network, which is hard to estimate. Though, at least an increase in performance measures is required with any component addition. The experiment documented in table 4.5 lies in gradually adding the preprocessing components in the same order as they have been described in this report. Most of the components behave as expected: recall (R) and F-measure (F) increase, while precision (P)

			CC			LP		SI	
corpus	value	basic	+ncs	+vcs	+gca	+prm	+vco	+ing	+ptc
AiMed	P	0.47	0.51	0.52	0.50	0.50	0.49	0.50	0.49
	R	0.08	0.19	0.22	0.38	0.40	0.46	0.47	0.46
	F	0.13	0.28	0.31	0.43	0.44	0.48	0.48	0.48
Brun	P	0.64	0.69	0.71	0.64	0.64	0.63	0.63	0.62
	R	0.09	0.21	0.27	0.39	0.42	0.48	0.48	0.47
	F	0.16	0.33	0.39	0.49	0.51	0.54	0.54	0.54
Hprd50	P	0.73	0.81	0.81	0.86	0.85	0.82	0.82	0.81
	R	0.11	0.24	0.29	0.45	0.49	0.54	0.61	0.61
	F	0.19	0.37	0.43	0.59	0.62	0.65	0.70	0.69
IEPA	P	0.71	0.78	0.75	0.76	0.76	0.73	0.74	0.74
	R	0.16	0.33	0.38	0.51	0.52	0.57	0.59	0.59
	F	0.26	0.46	0.50	0.61	0.62	0.64	0.66	0.65
LLL05	P	0.77	0.79	0.81	0.86	0.86	0.86	0.87	0.87
	R	0.06	0.31	0.38	0.59	0.59	0.65	0.71	0.72
	F	0.12	0.44	0.52	0.70	0.70	0.74	0.78	0.79
BC-PPI	P	0.52	0.45	0.46	0.42	0.41	0.38	0.36	0.36
	R	0.11	0.28	0.36	0.55	0.57	0.63	0.63	0.65
	F	0.18	0.34	0.40	0.48	0.48	0.47	0.46	0.46

Table 4.5: Precision (P), recall (R) and F-measure (F) in dependence on increasing number of applied skeletonization components. Legend: CC ∼ resolving cluster complexity, LP ∼ resolving language pointers, SI ∼ keeping semantical integrity; basic ∼ no transformations, ncs ∼ reduction of noun cluster sequence, vcs ∼ verb cluster reduction, gca ∼ gene appositions and coordinations, prm ∼ pronoun mapping, vco ∼ resolving verb coordinations, ing ∼ resolving *ing*-forms, ptc ∼ resolving past participles.

varies, depending on how much the underlying idea is likely to be modified by the given operation. However, the last component, past participle resolving, shows often a decrease in both recall and F-measure.

4.3.5 Impact of Individual Components

Noun cluster sequence reduction and verb cluster reduction are considered to be the core components of the sentence skeletonization, since they construct the backbone of the sentence skeleton. The other components, in contrast, have been designed as improvements of this basis, as a result of which they are inherently incomparable with the core components in degree of their contribution to the resulting performance. However, they may be more easily compared between each other, when applied on the skeleton backbone. Such an experiment is documented in table 4.6 in terms of increments or decrements of the observed performance qualities held by the skeleton backbone. Focusing on the recall and F-measure, there is mostly an increment or at least stagnation in these

corpus	value	ncs+vcs	ncs+vcs +gca	ncs+vcs +prm	ncs+vcs +vco	ncs+vcs +ing	ncs+vcs +ptc
AiMed	P	0.52	-0.02	+0.00	-0.01	-0.01	+0.01
	R	0.22	+0.15	+0.01	+0.03	+0.01	+0.01
	F	0.31	+0.12	+0.01	+0.03	+0.01	+0.01
Brun	P	0.71	-0.07	+0.00	-0.00	-0.00	-0.00
	R	0.27	+0.12	+0.02	+0.03	+0.01	+0.00
	F	0.39	+0.09	+0.02	+0.03	+0.01	+0.00
Hprd50	P	0.81	+0.05	-0.01	-0.02	-0.02	+0.00
	R	0.29	+0.16	+0.03	+0.02	+0.01	+0.01
	F	0.43	+0.16	+0.03	+0.02	+0.01	+0.01
IEPA	P	0.75	+0.01	-0.00	-0.02	-0.00	-0.01
	R	0.38	+0.13	+0.01	+0.03	+0.01	-0.01
	F	0.50	+0.11	+0.01	+0.02	+0.01	-0.01
LLL05	P	0.81	+0.05	+0.00	-0.00	-0.01	+0.00
	R	0.38	+0.20	+0.01	+0.04	+0.01	+0.00
	F	0.52	+0.18	+0.01	+0.04	+0.01	+0.00
BC-PPI	P	0.46	-0.04	-0.01	-0.04	-0.01	+0.01
	R	0.36	+0.19	+0.01	+0.04	+0.01	+0.03
	F	0.40	+0.07	+0.00	+0.00	+0.00	+0.02

Table 4.6: Precision (P), recall (R) and F-measure (F) increments of additional skeletonization components applied individually to ncs (∼ reducing noun cluster sequence) and vcs (∼ verb cluster reduction). Legend: gca ∼ gene appositions/coordinations, prm ∼ pronoun mapping, vco ∼ resolving verb coordinations, ing ∼ resolving *ing*-forms, ptc ∼ resolving past participles.

rates; however, in one case a decrement can be observed, which lowers the confidence in the concerned operation (namely resolving past participles) as component of the sentence skeletonization.

4.3.6 Penalty Analysis

All the experiments except for the present one are conceived assuming the maximum allowed penalty to be set to infinity, i.e. the penalties are ignored. The subject of the present experiment is the influence of the maximum allowed penalty on the overall performance measures. The resulting characteristics are depicted in figures 4.1 through 4.6. The initial decrease of precision results from enlarging the extent of the language stuff accepted as sequential and thus it may be considered as price for the rapid initial growth of recall and F-measure. Starting with four, however, all three measures become constant, which implies that (a) ignoring the penalties in other experiments did not lower the performance rates, and (b) higly penalized gene pairs may be ignored (which may be useful for more advanced sequential algorithm). However, since the frequency of growing penalty values undergoes exponential decay, high penalties (say higher than 6 or 8) appear quite rarely. As a result, the penalty characteristics do not provide any optimal penalty values, which raises

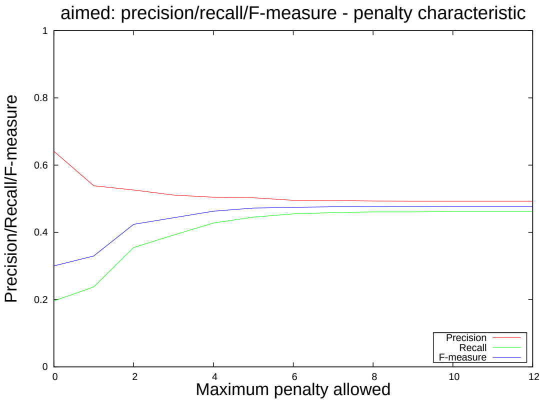


Figure 4.1: AiMed: penalty characteristic

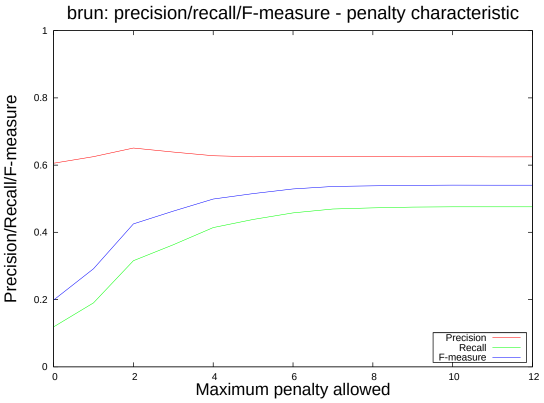


Figure 4.2: Brun: penalty characteristic

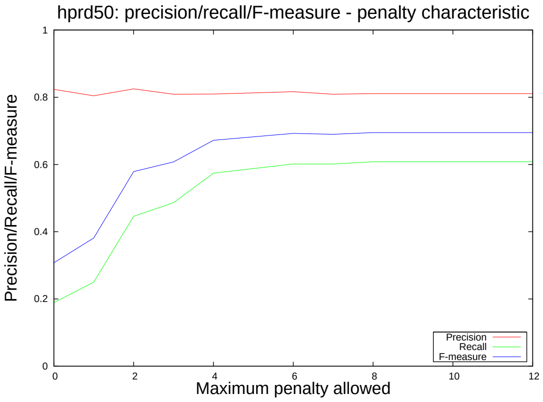


Figure 4.3: HPRD50: penalty characteristic

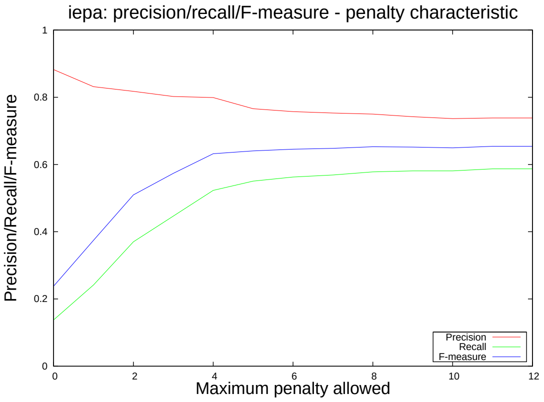


Figure 4.4: IEPA: penalty characteristic

doubts about actual usability of the designed penalty measure and calls for its redefinition.

An additional note is to be made about how the gene pair penalty is actually counted. Assume the following sentence and its skeleton:

(105) G1@GENE activates@VVZ the@DT expression@NN of@IN G2@GENE which@WDT induces@VVZ G3@GENE → G1{0,0}@GENE activates@VVz G2{0,2}@GENE which@WDT induces@VVZ G3{0,0}@GENE

The penalty of the pair G1-G2 is 2 due to the backward propagation, whereas the penalty of the pair G2-G3 is 0, since there was no seamntic shift concerning G2 with respect to G3. Thus, whether the penalty for backward propagation is counted or not, is immanent property of each sequential pattern (for details refer to B, page 53).

4.3.7 Pattern Analysis

The sequential patterns, as listed in section B (page 53), differ in how successful they are in detecting gene interactions. Moreover, only a subset of these patterns proved to by successful.

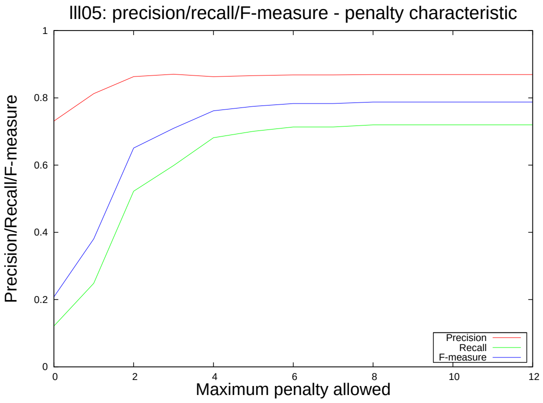


Figure 4.5: LLL05: penalty characteristic

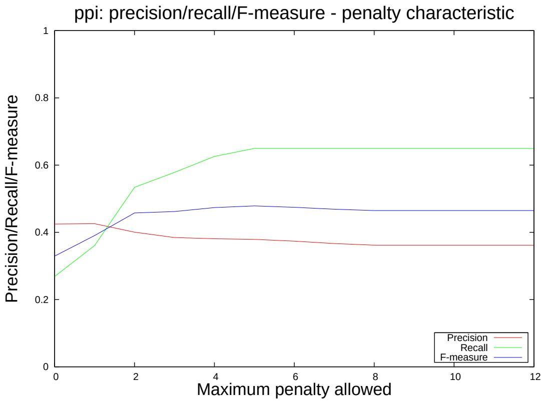


Figure 4.6: BC-PPI: penalty characteristic

Table 4.7 summarizes the efficiency of pattern groups (verb based, noun based, adjective based) on each of the testing corpora, showing that verb based patterns outperform the other patterns. Table 4.8, on the other hand, concentrates on the efficiency of individual patterns, based on the results from all testing corpora. Such characteristics may be useful for designing and tuning a more advanced sequential algorithm employed in the gene interaction extraction task. The most successful patterns define indirectly a subset of nodes, in which the projection of the language stuff into a token sequence is the closest approximation of the actual, multidimensional language reality.

Corpus	VK eff.	NK eff.	AK eff.	$\mathrm{TP}_{\mathrm{VK}}/\mathrm{TP}$	$\mathrm{TP}_{\mathrm{NK}}/\mathrm{TP}$	$\mathrm{TP}_{\mathrm{AK}}/\mathrm{TP}$
AiMed	0.55	0.49	0.51	0.71	0.18	0.11
Brun	0.69	0.64	0.57	0.72	0.18	0.10
Hprd50	0.78	0.81	0.86	0.69	0.25	0.06
IEPA	0.85	0.69	0.82	0.60	0.17	0.24
LLL05	0.58	0.90	0.90	0.79	0.06	0.15
BC-PPI	0.40	0.35	0.55	0.74	0.15	0.11

Table 4.7: Efficiency of various kernel types. Legend: VK $\sim$ verb kernels, NK $\sim$ noun kernels, AK $\sim$ adjective kernels, TP $\sim$ number of true positives, TP_{x} $\sim$ number of true positives built around kernel type x.

4.3.8 Result Summary

Based on the experimental analysis of the skeletonization impact, following conclusional remarks can be made: (i) Excluding semantics from sequential approach does not prevent it from influencing the evaluation results. Therefore, a successful match requires both sequentiality (*Seq*) and semantics (*Sem*) to agree. As a result, a match is (a) *true positive* $\Rightarrow$ $SeqOk == True$ and $SemOk == True$; (b) *false positive* $\Rightarrow$ $SeqOk == False$ or $SemOk == False$; (c) *true negative* $\Rightarrow$ $SeqOk == True$; (d) *false negative* $\Rightarrow$ $SeqOk == False$. (ii) Only a subset of lan-

Pattern	Precision	$\mathrm{TP_p}/\mathrm{TP_{vp}}$	$\mathrm{TP_p}/\mathrm{TP}_{all}$
gene+verb+gene	0.59	0.46	0.33
gene+verb+prep+gene	0.60	0.35	0.25
gene+verb+adje+prep+gene	0.49	0.04	0.03
gene+rel+verb+gene	0.53	0.03	0.02
gene+comma+rel+verb+gene	0.50	0.03	0.02
gene+prep+verb+gene	0.40	0.02	0.01
gene+comma+rel+verb+prep+gene	0.52	0.02	0.01
gene+prep+verb+prep+gene	0.54	0.02	0.01
gene+rel+verb+prep+gene	0.59	0.02	0.01
gene+verb+adje+prep+verb+gene	0.92	0.01	0.01
Pattern	Precision	$\mathrm{TP_p}/\mathrm{TP_{np}}$	$\mathrm{TP_p}/\mathrm{TP}_{all}$
noun+prep+gene+prep+gene	0.65	0.44	0.08
gene+noun+prep+gene	0.69	0.36	0.06
noun+prep+gene+conj+gene	0.53	0.20	0.04
Pattern	Precision	$\mathrm{TP_p}/\mathrm{TP_{ap}}$	$\mathrm{TP_p}/\mathrm{TP}_{all}$
gene+adje+gene	0.64	0.62	0.07
gene+adje+prep+gene	0.54	0.36	0.04

Table 4.8: Efficiency of individual patterns based on results from all testing corpora. Label: TP $\sim$ number of true positives, p $\sim$ pattern, [v,n,a]p $\sim$ patterns built around verb/noun/adjective kernel, all $\sim$ all patterns. Note: results only for patterns having $\mathrm{TP_p}/\mathrm{TP_{all}} \geq 0.01$.

guage phenomena breaking the sentence sequentiality is covered by the sentence skeletonization. (iii) Skeletonization components may fail to improve sentence skeletonization due to grammatical generalizations. (iv) The impact of the individual skeletonization components is hard to estimate due to mutual dependencies and interferences with each other. (vi) The penalty quantifying measurable semantical error proved not to be enough informative.

4.4 Comparison with Other Approaches

Providing an informative, relevant result comparison proves to be difficult, since the evaluation methodologies differ significantly among research teams. In this section, several performance comparisons of the approach proposed in this report with another approaches are presented.

Pyysalo, Airola et al. [1,53] use very similar approach to evaluate extraction performance on five corpora, four of which are used also in our experiments: AiMed, HPRD50, IEPA and LLL05. In [53], differences between these corpora are studied using (1) a naive cooccurence method and (2) the RelEx system proposed by Fundel [24] and mentioned also in this report (section 2.3.2, page 14). In [1] the same corpus collection is used to evaluate a graph kernel method, which has been also mentioned in this report (section 2.3.4, page 15). The comparison of all these three approaches with the method proposed in this report is given in table 4.9.

		AiMed	HPRD50	IEPA	LLL05
P	Graph kernel	0.529	0.643	0.696	0.725
	Cooccurences	0.17	0.38	0.41	0.50
	RelEx	0.40	0.76	0.74	0.82
	Skel. + seq.	0.49	0.81	0.74	0.87
R	Graph kernel	0.618	0.658	0.827	0.872
	Cooccurences	0.95	1.0	1.0	1.0
	RelEx	0.50	0.64	0.61	0.72
	Skel. + seq.	0.46	0.61	0.59	0.72
F	Graph kernel	0.564	0.634	0.751	0.768
	Cooccurences	0.29	0.55	0.58	0.66
	RelEx	0.44	0.69	0.67	0.77
	Skel. + seq.	0.48	0.69	0.65	0.79

Table 4.9: Performance comparison 1. Legend: Graph kernel ∼ results from [1], Cooccurences ∼ results from [53], RelEx ∼ results from [53], Skel. + seq. ∼ the approach described in this project.

Corpus	Method	P	R	F
AiMed	SVM	0.7752	0.4351	0.5561
	Skel. + seq.	0.49	0.46	0.48
Brun	SVM	0.8515	0.8479	0.8496
	Skel. + seq.	0.62	0.47	0.54

Table 4.10: Performance comparison 2. Legend: SVM ∼ selected results from [16], Skel. + seq. ∼ the approach described in this project.

Erkan et al. [16] evaluate their system based on deep parsing and machine learning on two corpora, AiMed and Brun; the comparison is given in table 4.10. Katrenko and Adriaans [35] evaluate their system built around alignment kernels also on two corpora, LLL05 and BC-PPI; the comparison is given in table 4.11.

Corpus	Method	P	R	F
LLL05	Alignment kernel	0.7425	0.8794	0.8051
	Skel. + seq.	0.87	0.72	0.79
BC-PPI	Alignment kernel	0.7556	0.7972	0.7756
	Skel. + seq.	0.36	0.65	0.46

Table 4.11: Performance comparison 3. Legend: Alignment kernel ∼ selected results from [35], Skel. + seq. ∼ the approach described in this project.

Obviously, a lot of approaches outperform the system proposed in this report. However, it must be taken into consideration that the approach based on applying ultrasimple sequential algorithm on skeletonized text was targeted only to evaluate the effect of sentence skeletonization (∼ text preprocessing); it was not seriously meant as a full featured system for gene interaction extraction.

Chapter 5

Summary

Since natural language is not sequential, linguistic preprocessing for sequential data mining (not limited to biomedical literature) can be understood as improving sentence sequentiality.

Based on a detailed analysis of biomedical texts, three classes of language phenomena breaking the sentence sequentiality have been identified: (1) arbitrary sentence complexity, (2) existence of language pointers and (3) existence of forms affecting the semantical integrity. To deal with these obstacles, seven heuristic transformations have been designed, all of which are employed to convert a sentence into a form called *sentence skeleton*. The sentence skeleton may be regarded as simplified form of the original sentence or sentence approximation (both grammatical and semantical), thus not being fully equivalent with the original sentence.

The impact of the resulting sentence skeletonization has been evaluated using an intentionally simple, clearly sequential algorithm. By applying this algorithm in the gene interaction extraction task on skeletonized sentences from various biomedical corpora, limitations of the sentence skeletonization have been identified. Furthermore, the usability of mining sequential patterns from sentence skeletons have been confirmed, provided that further improvements in sentence skeletonization will be made and a more advanced sequential algorithm will be used.

Chapter 6

Further Work

The current system consists of two components: (1) sentence skeletonization and (2) sequential data mining. Exploring capabilities of the sequential data mining has remained beyond the scope of the current project, therefore, an analysis of these capabilities would be the first task of the further work.

The sentence skeletonization, the core of the current project, has been shown have several weak points. Furthermore, another limitation of the current scope must be taken into consideration, namely that the gene interaction extraction implemented in this project and mostly expected by the corpus curators is sentence oriented. A lot of new problems, however, become relevant, when moving from sentence level to text level. The shift of interest towards full texts can be observed in the task definitions of the BioCreative challenge [19]. Focusing on continuous texts is necessary, since the real tasks are always text oriented.

Below we present a list of ideas possibly improving the preprocessing phase of the presented system. Note that the list below (1) contains only those improvements applicable to the current solution and (2) it is not complete.

Cross-clause (sentence level) interactions. Due to enormous complexity of biomedical language, sentence structure has to be determined more precisely to detect interactions overlapping single clause boundaries. Possible solution: By exploiting further the sentence structure simplification, sentence schemes could be retrieved (ex. 106). From these generalized sentence patterns, rules capable of determining, which clause pairs should be analyzed for possible cross-clause interactions, could be learned semi-automatically. After applying such rules onto a given sentence, additional verb skeletons would be added into the verb skeleton pool. A short demonstration on the example 106:

(106) We investigated G1, which interacts with G2, and analyzed G3, which activates G4.
→ verb gene , rel verb gene , conj verb gene , rel verb gene
→ [independent_1(gene)] [subordinate_1(gene)] [independent_2(gene)] [subordinate_2(gene)]

Only selected clause pairs would be investigated for cross-clause interactions: *independent_1 and subordinate_1*, *independent_2 and subordinate_2* and *independent_1 and independent_2*.

Cross-sentence (text level) interactions. In real text, interactions may extend over multiple sentences, i.e. we need to operate also on the textual level. Possible solution: After retrieving patternalized sentence schemes, paragraph schemes could be constructed by simply chaining the sentence schemes. Most probably, only two subsequent sentences would be reasonable to analyze for cross-sentence interactions. The theme - rheme principle would play the essential role (ex. 107). Interpretational rules could be learned similarly to those on sentence level.

(107) Our research focused on G1. It proved to activate G2.
→ {[independent_1(gene)]} {[(pronoun)independent_1(gene)]}
→ {[independent_1(gene)]} {[(gene)independent_1(gene)]}

Higher-level coordinations. Apart from gene names and ordinary words, also non-minimal clusters (not necessarily reducible to minimal clusters) may build coordinate pairs (ex. 108). Moreover, clusters of different complexity may be connected coordinatively, which can easily mess up the whole sentence comprehension. Possible solution: Symmetric higher level coordinations could be resolved by exploring the symmetry of components on the left side and on the right side of the given coordinative element. No clear suggestions for asymmetric coordinations yet.

(108) G1 activates G2 (in presence of G3) and (in absence of G4)

Modality issues. Modality is a powerful player; standing mostly in background, it is a crucial language control component, acting as means of pragmatics. In the following, problems concerning all three main modality types will be shortly introduced.

Subjective epistemic modality. The pragmatic interpretational pattern of this modality type is as follows: *The writer believes with some degree of confidence that something is true.* Such beliefs/evaluations of the current knowledge are expressed either by verbs (ex. 109) or by adverbs (ex. 110). All adverbs are, however, presently removed since they do not allow for sufficient generalizations of the resulting patterns.

(109) G1 might[We assume that G1] interact with G2 (!)

(110) G1 hardly@RB interacts with G2 → G1 interacts with G2 (!)

Objective epistemic modality. The pragmatic interpretational pattern of this modality type is as follows: *The writer claims that something is true because there are (from his point of view) reasonable arguments supporting it.* An example of a misleading sentence is given in ex. 111.

(111) X. Y. claims that G1 interacts with G2 (!)

Not-epistemic modality. From the variety of non-epistemic modality we select the cases represented by example . An example of a misleading sentence is given in ex. 112.

(112) We investigated whether G1 interacts with G2 (!)

Possible solution: Employing a custom dictionary, integration into the sentence scheme analysis suggested above.

Disambiguation. Disambiguation has been shown to be an extremely difficult task (especially *ing*-forms), the importance of resolving this problem is obvious. Possible solution: Resolving ambiguous cases by simply investigating all interpretations (ex. 113), evaluating the resulting structures within the sentence scheme analysis suggested above.

(113) interacting@VVG → interacting@verb; interacting@VVG → interacting@noun(verbal noun); interacting@VVG → interacting@adjective

Negation. Detection of negative clauses is currently limited to those containing the particle *not*. However, the negation can take a lot of other forms, such as particles *no, none*, compound conjunctions *neither* - *nor*, nominal suffixes *in-, im-, a-, dis-* etc. Furthermore, often it appears to be closely adherently related to sentence modality (the control power of negation is fairly comparable with the power of modality), thus being the instrument of pragmatics. Possible solution: exploiting more precisely word morphology, employing closed word dictionary, integration into sentence scheme analysis suggested above.

Appendix A

Penn Treebank Tagset

Number	Abbrebiation	Description
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential *there*
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	*to*
26	UH	Interjection

Continued on next page

Number	Abbrebiation	Description
27	V[BHV]	Verb [*to be*, *to have*, other], base form
28	V[BHV]D	Verb [*to be*, *to have*, other], past tense
29	V[BHV]G	Verb [*to be*, *to have*, other], gerund or present participle
30	V[BHV]N	Verb [*to be*, *to have*, other], past participle
31	V[BHV]P	Verb [*to be*, *to have*, other], non-3rd person singular present
32	V[BHV]Z	Verb [*to be*, *to have*, other], 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	Wh-adverb
37	SENT	Right sentence boundary
38	,	Comma
39	:	Collon
40	;	Semicollon
41	GENE	Gene

Table A.1: The extended Penn Treebank Tagset

Appendix B

Sequential Patterns

Pattern scheme	Penalty for left gene BP	Kernel index
gene+infix+verb+gene	yes if infix is empty	1+infix_inc
gene+infix+verb+prep+gene	yes if infix is empty	1+infix_inc
gene+infix+verb+adj+prep+gene	yes if infix is empty	2+infix_inc
gene+infix+verb+adj+prep++verb+gene	yes if infix is empty	2+infix_inc
verb+gene+prep+gene	yes	0
gene+prep+verb+gene	yes	2
gene+prep+verb+prep+gene	yes	2
gene+noun+prep+gene	no	1
noun+prep+gene+prep+gene	yes	0
noun+prep+gene+conj+gene	yes	0
gene+adj+gene	no	1
gene+adje+prep+gene	no	1
gene+noun+adj+prep+gene	no	2

Table B.1: List of sequential patterns for evaluation of the text preprocessing. Legend: prep ∼ preposition, adj ∼ adjective, conj ∼ conjunction, rel ∼ relative pronoun, relp ∼ possessive relative pronoun, infix_inc ∼ increment of the given infix (table B.2), BP ∼ backward propagation.

Infix type	Kernel index increment
(empty)	0
rel	1
comma+rel	2
relp+noun	2
comma+relp+noun	3

Table B.2: Infixes for verb based patterns. Legend: rel ∼ relative pronoun, relp ∼ possessive relative pronoun.

Bibliography

[1] Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9(Suppl 11):S2, 2008.

[2] Sophia Ananiadou. A methodology for automatic term recognition. In *Proceedings of the 15th conference on Computational linguistics*, pages 1034–1038, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[3] Daniel Berleant. *IEPA Corpus.* University of Arkansas at Little Rock, http://class.ee.iastate.edu/berleant/s/IEPA.htm. Accessed March 2010.

[4] Christian Blaschke and Alfonso Valencia. The Potential Use of SUISEKI as a Protein Interaction Discovery Tool. *Genome Informatics*, 12:123–134, 2001.

[5] Christine Brun. *Christine Brun Corpus.* http://www.biocreative.org/accounts/login/?next=/-resources/. Accessed March 2010.

[6] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[7] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[8] Lifeng Chen, Hongfang Liu, and Carol Friedman. Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21:248–256, 2005.

[9] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 160–163, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[10] K. Bretonnel Cohen, George K. Acquaah-Mensah, Andres E. Dolbey, and Lawrence Hunter. Contrast and variability in gene names. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain*, pages 14–20, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[11] K. Bretonnel Cohen and Lawrence Hunter. *Artificial Intelligence Methods And Tools For Systems Biology*, chapter Natural Language Processing and Systems Biology, pages 147–173. Springer Netherlands, 2004.

[12] Marc E. Colosimo, Alexander A. Morgan, Alexander S. Yeh, Jeffrey B. Colombe, and Lynette Hirschman. Data preparation and interannotator agreement: Biocreative task 1b. *BMC Bioinformatics*, 6(Suppl 1):S12, 2005.

[13] Mark Craven and Johan Kumlien. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *Proceedings of the 7th Interactions conference on intelligent systems for molecular biology*, pages 77–86, 1999.

[14] Critical Assessment of Information Extraction systems in Biology, http://www.biocreative.org/accounts/login/?next=/resources/. *BioCreAtIvE resources.* Accessed March 2010.

[15] Ian Donaldson, Joel Martin, Berry de Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D. Bader, Katerina Michalickova1, Tony Pawson, and Christopher W. V. Hogue. PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4, 2003.

[16] G. Erkan and D. R. Radev. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning*, pages 228–237. Association for Computational Linguistics, 2007.

[17] Alexander A. Morgan et al. Overview of biocreative ii gene normalization. *Genome Biology*, 9(Suppl 2):S3, 2008.

[18] Larry Smith et al. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9(Suppl 2):S2, 2008.

[19] Martin Krallinger et al. Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome Biology*, 9(Suppl 2):S4, 2008.

[20] Yi feng Lin, Tzong han Tsai, Wen chi Chou, Kuen pin Wu, Ting yi Sung, and Wen lian Hsu. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pages 56–61, 2004.

[21] Jenny Finkel, Shipra Dingare, Christopher, Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. Exploring the boundaries: gene and protein identification in biomedical text. *BMC Bioinformatics*, 6(Suppl 1):S5, 2005.

[22] Carol Friedman, Pauline Kra, Hong Yu, Michael Krauthammer, and Andrey Rzhetsky. Genies: a natural-language processing system for the extraction of molecular pathways from. *Bioinformatics*, 17(Suppl 1):74–82, 2001.

[23] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. *Pacific Symposium on Biocomputing*, pages 707–718, 1998.

[24] Katrin Fundel, Daniel Gttler, Ralf Zimmer, and Joannis Apostolakis. A simple approach for protein name identification: prospects and limits. *BMC Bioinformatics*, 6(Suppl 1):S15, 2005.

[25] Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[26] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: The pasta system. *Bioinformatics*, 19:135–143, 2003.

[27] Robert Gaizauskas, George Demetriou, and Kevin Humphreys. Term Recognition and Classification in Biological Science Journal Articles. In *In Proc. of the Computional Terminology for Medical and Biological Applications Workshop of the 2 nd International Conference on NLP*, pages 37–44, 2000.

[28] Jörg Hakenberg. *BC-PPI Corpus*. Humboldt-Universität zu Berlin - Institut für Informatik, http://www2.informatik.hu-berlin.de/ hakenber/corpora/. Accessed March 2010.

[29] Jörg Hakenberg, Conrad Plake, Loic Royer, Hendrik Strobelt, Ulf Leser, and Michael Schroeder. Gene mention normalization and interaction extraction with context models and sentence motifs. *Genome Biology*, 9(Suppl 2):S14, 2008.

[30] James Hammerton, Miles Osborne, Susan Armstrong, and Walter Daelemans. Introduction to special issue on machine learning approaches to shallow parsing. *The Journal of Machine Learning Research*, 2:551–558, 2002.

[31] Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.

[32] Jerry R. Hobbs. Information extraction from biomedical text. *Journal of Biomedical Informatics*, 35(4):260–264, 2002.

[33] Minlie Huang, Xiaoyan Zhu, Yu Hao, Donald G. Payan, Kunbin Qu, and Ming Li. Discovering patterns to extract protein–protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612, 2004.

[34] K. Humphreys, G. Demetriou, and R. Gaizauskas. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Proceedings of Pacific Symposium on Biocomputing*, pages 506–516, 2000.

[35] Sophia Katrenko and Pieter Adriaans. A local alignment kernel in the context of NLP. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[36] Zhenzhen Kou, William W. Cohen, and Robert F. Murphy. High-recall protein entity recognition using a dictionary. *Bioinformatics*, 21(1):266–273, 2005.

[37] Michael Krauthammer and Goran Nenadic. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6):512–526, 2004.

[38] John Lafferty and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.

[39] Gondy Leroy, Hsinchun Chen, and Jesse D. Martinez. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of Biomedical Informatics*, 36(3):145–158, 2003.

[40] Ulf Leser and Jörg Hakenberg. What makes a gene name? named entity recognition in the biomedical literature. *Briefings in bioinformatics*, 6:357–369, 2005.

[41] Tyne Liang and Ping-Ke Shih. Empirical Textual Mining to Protein Entities Recognition from pubmed corpus. In *Natural Language Processing and Information Systems*, pages 56–66, 2005.

[42] Ludwig-Maximilians-Universität München, Lehr- und Forschungseinheit für Bioinformatik, Institut für Informatik, http://code.google.com/p/priseinsttechuwt/source/-browse/trunk/PRISE/src/java/DEEPERsource/DEEPERsource/source/resource/-hprd50.xml?spec=svn3&r=3. *HPRD50 Corpus.* Accessed March 2010.

[43] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

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

[45] Nicolas Méger and Christophe Rigotti. Constraint-based mining of episode rules and optimal window sizes. In *PKDD '04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 313–324, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[46] Tomohiro Mitsumori, Sevrani Fation, Masaki Murata, Kouichi Doi, and Hirohumi Doi. Gene/protein name recognition based on support vector machine using dictionary as features. *BMC Bioinformatics*, 6(Suppl 1):S8, 2005.

[47] Raymond J. Mooney. *AiMed.* University of Texas at Austin, https://wiki.inf.ed.ac.uk/TFlex/AiMed. Accessed March 2010.

[48] M. Narayanaswamy, K. E. Ravikumar, and K. Vijay-Shanker. A biological named entity recognizer. *Pacific Symposium on Biocomputing 8*, pages 427–438, 2003.

[49] Chikashi Nobata, Nigel Collier, and Jun ichi Tsujii. Automatic Term Identification and Classification in Biology Texts. In *In Proc. of the 5th NLPRS*, pages 369–374, 1999.

[50] Tu Minh Phuong, Doheon Lee, and Kwang Hyung Lee. *Advances in Knowledge Discovery and Data Mining*, chapter Learning Rules to Extract Protein Interactions from Biomedical Text, page 568. Springer Berlin/Heidelberg, 2003.

[51] M. Plantevit, T. Charnois, J. Klema, C. Rigotti, and B. Cremilleux. Combining sequence and itemset mining to discover named entities in biomedical texts: A new type of pattern. *International Journal of Data Mining, Modelling and Management*, 1:119–148, 2009.

[52] J. Pustejovsky, J. Castafio, J. Zhang, M. Kotecki, and B. Cochran. Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations. In *Proceedings of the Pacific symposium on biocomputing*, pages 362–373, 2002.

[53] Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6, 2008.

[54] Sampo Pyysalo, Tapio Salakoski, Sophie Aubin, and Adeline Nazarenko. Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, 7(Suppl 3):S2, 2006.

[55] Claude Roux, Denys Proux, Francois Rechenmann, and Laurent Julliard. An Ontology Enrichment Method for a Pragmatic Information Extraction System gathering Data on Genetic Interactions. In *Proceedings of the eight International conference on intelligent systems for molecular biology*, pages 279–285. AAAI Press, 2000.

[56] Helmut Schmid. *Treetagger.* Institute for Computational Linguistics of the University of Stuttgart, http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/. Accessed March 2010.

[57] Burr Settles. Biomedical Named Entity Recognition using Conditional Random Fields and Rich Feature Sets. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 107–110, Geneva, Switzerland, August 28th and 29th 2004. COLING.

[58] Marios Skounakis, Mark Craven, and Soumya Ray. Hierarchical Hidden Markov Models for Information Extraction. In *IJCAI*, pages 427–433, 2003.

[59] B. J. Stapley and G. Benoit. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. In *Processing of the Pacific symposium on biocomputing*, pages 529–540, 2000.

[60] Javier Tamames. Text detective: a rule-based system for gene annotation in biomedical texts. *BMC Bioinformatics*, 6(Suppl 1):S10, 2005.

[61] Unité Mathématique, Informatique et Génome, http://genome.jouy.inra.fr/texte/LLLchallenge/. *LLL05 Corpus.* Accessed March 2010.

[62] University of Pennsylvania, http://www.cis.upenn.edu/ treebank/. *The Penn Treebank Tagset.* Accessed March 2010.

[63] Hatzivassiloglou V., Dubou P. A., and Rzhetsky A. Disambiguating proteins, genes, and rna in text: a machine learning approach. *Bioinformatics*, 17(Suppl 1):S97, 2001.

[64] Hua Xu, Jung-Wei Fan, George Hripcsak, Eneida A. Mendonça, Marianthi Markatou, and Carol Friedman. Gene symbol disambiguation using knowledge-based profiles. *Bioinformatics*, 23(8):1015–1022, 2007.

[65] Alexander Yeh, Alexander Morgan, Marc Colosimo, and Lynette Hirschman. Biocreative task 1a: gene mention finding evaluation. *BMC Bioinformatics*, 6(Suppl 1):S2, 2005.

[66] Deyu Zhou and Yulan He. Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*, 41:393–407, 2008.

[67] GuoDong Zhou, Dan Shen, Jie Zhang, Jian Su, and SoonHeng Tan. Recognition of protein/gene names from text using an ensemble of classifiers. *BMC Bioinformatics*, 6(Suppl 1):S7, 2005.