Mining Protein-Protein Interactions from Published Abstracts with MontyLingua[1]

Maurice HT Ling
School of Chemical and Life Sciences
Singapore Polytechnic, Singapore
Department of Zoology
The University of Melbourne, Australia

Christophe Lefevre
Institute for Technology Research and Innovation
Deakin University, Australia

Kevin R Nicholas
Institute for Technology Research and Innovation
Deakin University, Australia

1 Introduction

PubMed currently indexes more than 20 million papers with more than one million papers added annually. A simple keyword search in PubMed showed that nearly 900 thousand papers on mouse and more than 1.3 million papers on rat research had been indexed in PubMed to date, and in the last four years, more than 150 thousand papers have been published on each of mouse and rat research. This trend of increased volume of research papers indexed in PubMed over the last 10 years makes it difficult for researchers to maintain an active and productive assessment of relevant literature. Information extraction (IE) has been used as a tool to analyze biological text to derive assertions on specific biological domains (Rebholz-Schuhmann et al., 2005), such as protein phosphorylation (Hu et al., 2005) or entity interactions (Abulaish and Dey, 2007).

A number of IE tools used for mining information from biological text can be classified according to their capacity for general application or tools that considers biological text as specialized text requiring domain-specific tools to process them. This has led to the development of specialized part-of-speech (POS) tag sets (such as SPECIALIST (National Library of Medicine, 2003)), POS taggers (such as

[1] This chapter is based on Ling, MHT, Lefevre, C, Nicholas, KR, Lin, F. 2007. Re-construction of Protein-Protein Interaction Pathways by Mining Subject-Verb-Objects Intermediates. In J.C. Ragapakse, B. Schmidt, and G. Volkert (Eds.), Proceedings of the Second IAPR Workshop on Pattern Recognition in Bioinformatics (PRIB 2007). Lecture Notes in Bioinformatics 4774. (pp. 286-299) Springer-Verlag., Ling, MHT, Lefevre, C, Nicholas, KR. 2008. Parts-of-Speech Tagger Errors Do Not Necessarily Degrade Accuracy in Extracting Information from Biomedical Text. The Python Papers 3 (1): 65-80, and Ling, MHT. 2009. Understanding Mouse Lactogenesis by Transcriptomics and Literature Analysis. Doctor of Philosophy. Department of Zoology, The University of Melbourne, Australia.

MedPost (Smith et al., 2004)), ontologies (Daraselia et al., 2004), text processors (such as MedLEE (Friedman et al., 1994)), and full IE systems, such as GENIES (Friedman et al., 2001), MedScan (Novichkova et al., 2003), MeKE (Chiang and Yu, 2003), Arizona Relation Parser (Daniel et al., 2004), GIS (Chiang et al., 2004), Jang et al. (Jang et al., 2006) and PIE (Kim et al., 2008). On the other hand, an alternative approach assumes that biological text are not specialized enough to warrant re-development of tools but adaptation of existing or generic tools will suffice. To this end, BioRAT (David et al., 2004) had modified GATE (Cunningham, 2000), MedTAKMI (Uramoto et al., 2004) had modified TAKMI (Nasukawa and Nagono, 2001), originally used in call centres, Santos (Santos et al., 2005) had used Link grammar parser (Sleator and Temperley, 1991), Feng et al. (Feng et al., 2007) used a purpose-built rule-dictionary hybrid for chemical name identification and GATE (Cunningham, 2000).

Although both systems demonstrated similar performance, either developing these systems or modifying existing systems were time-consuming (Jensen et al., 2006). Although work by Grover (Grover et al., 2002) suggested that native generic tools may be used for biological text, a recent review had highlighted successful uses of a generic text processing system, MontyLingua (Eslick and Liu, 2005; Liu, 2004), for a number of purposes (Ling, 2006). For example, MontyLingua has been used to process published economics papers for concept extraction (van Eck and van den Berg, 2005). The need to modify generic text processors had not been formally examined and the question of whether an un-modified, generic text processor can be used in biological text analysis with comparable performance, remains to be assessed.

In this study, we evaluated a native, generic text processing system, MontyLingua (Liu and Singh, 2004), in a two-layered generalization-specialization architecture (Novichkova et al., 2003) where the generalization layer processes biological text into an intermediate knowledge representation for the specialization layer to extract genic or entity-entity interactions. This system demonstrated 86.1% precision using Learning Logic in Languages 2005 evaluation data (Cussens, 2005), 88.1% and 90.7% precisions in extracting protein-protein binding and activation interactions respectively. Our results were comparable to previous work which modified generic text processing systems which reported precision ranging from 53% (Malik et al., 2006) to 84% (Chiang et al., 2004), suggesting this modification may not improve the efficiency of information retrieval.

2 System Description

We have developed a biological text mining system, known as Muscorian, for mining protein-protein interrelationships in the form of subject-relation-object (for example, protein X bind protein Y) assertions. Muscorian is implemented as a 3-module sequential system of entity normalization, text analysis, and protein-protein binding finding, as shown in Figure 1. It is available for academic and non-profit users through http://www.sourceforge.net/projects/muscorian

2.1 Entity Normalization

Entity normalization is the substitution of the long form of either a biological or chemical term with its abbreviated form. This is essential to correct part-of-speech tagging errors which are common in biological text due to multi-worded nouns. For example, the protein name “phosphatase and tensin homolog deleted on chromosome 10” has to be recognized as a single noun and not a phrase. In this study, we attempt to mine protein-protein interactions and consolidate this knowledge to produce a map. Therefore, the naming convention of the protein entities must be standardized to allow for matching. However, this is not the case for biological text and synonymous protein names exist for virtually every protein. For example,

"MAP kinase kinase", "MAPKK", "MEK" and "MAPK/Erk kinase" referred to the same protein. Both of these problems could be either resolved or minimized by reducing multi-worded nouns into their abbreviated forms.

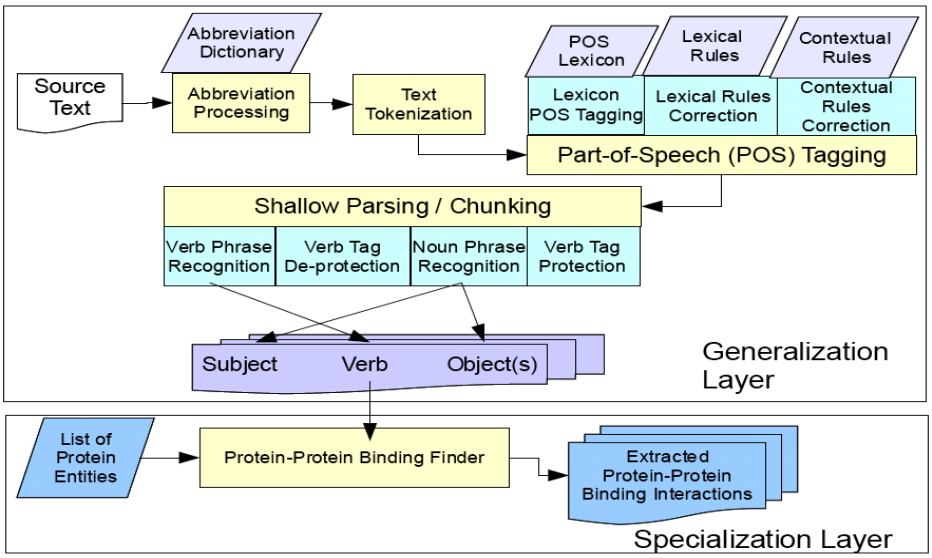

Figure 1: Schematic Diagram Illustrating the Operations of Musorian.

A dictionary-based approach was used for entity normalization to a high level of accuracy and consistency. The dictionary was assembled as follows: firstly, a set of 25000 abstracts from PubMed was used to interrogate Stanford University's BioNLP server (Chang et al., 2002) to obtain a list of long forms with its abbreviations and a calculated score. Secondly, only results with the score of more than 0.88 were retained as it is an inflection point of ROC graph (Chang et al., 2002), which is a good balance between obtaining the most information while reducing curation efforts. Lastly, the set of long form and its abbreviations was manually curated with the help of domain experts.

The domain experts curated dictionary of long forms and its abbreviated term was used to construct a regular expression engine for the process of recognition of the long form of a biological or chemical term and substituting it with its corresponding abbreviated form.

2.2 Text Analysis

Entity normalized abstracts were then analyzed textually by an un-modified text processing engine, MontyLingua (Eslick and Liu, 2005), where they were tokenized, part-of-speech tagged, chunked, stemmed and processed into a set of assertions in the form of 3-element subject-verb-object(s) (SVO) tuple, or more generally, subject-relation-object(s) tuple. Therefore, a sequential pattern of words which formed an abstract was transformed through a series of pattern recognition into a set of structurally-definable assertions.

Before part-of-speech tagging is possible, an abstract made up of one or more sentences had to be separated into individual sentences. This is done by regular expression recognition of sentence delimiters,

such as full-stop, ellipse, exclamation mark and question mark, at the end of a word (regular expression: ([?!]+|[.][.]+)$) with an exception of acronyms. Acronyms, which are commonly represented with a full-stop, for example “Dr.”, are not denoted as the end of a sentence and were generally prevented by an enumeration of common acronyms.

Individual sentences were then separated into constituent words and punctuations by a process known as tokenization. Tokenization, which is essential to atomize a sentence into atomic syntactic building blocks, is generally a simple process of splitting of an English sentence in words using whitespaces in the sentence, resulting in a list of tokens (words). However, there were three problems which were corrected by examining each token. Firstly, punctuations are crucial in understand a written English sentence, but typographically a punctuation is usually joined to the presiding word. Hence, punctuation separation from the presiding word is necessary. However, it resulted in incorrect tokenization with respect to acronyms and decimal numbers. For example, “... an appt. for ...” will be tokenized to “... an appt . for ...” and “$4.20”" will be “$ 4 . 20”. This problem was prevented by pre-defining acronyms and using regular expressions, such as “^[$][0-9]{1,3}[.][0-9][0-9](?[.]?)$”. Lastly, common abbreviated words, such as “don't”, were expanded into two tokens of “do” and “n't”. Despite the above error correction measures, certain text such as mathematical equations, which might be used to describe enzyme kinetics in biological text, will not be tokenized correctly. In spite of this limitation, the described tokenization scheme is still appropriate as extraction of enzyme kinetics or mathematical representations are not the aims of this study.

Each of the tokens (words and punctuations) in a tokenized sentence is then tagged using Penn TreeBank Tag Set (Marcus et al., 1993) by a Brill Tagger, trained on Wall Street Journal and Brown corpora, which operates in two phases. Using a lexicon, containing the likely tag for each word, each word is tagged. This is followed by a phase of correction using lexical and contextual rules, which were learnt using training with a tagged corpora, in this case, Wall Street Journal and Brown corpora. Lexical rules uses a combination of preceding tag and prefix or suffix of the token (word) in question. For example, the rule “NN ing fhassuf 3 VBG” defines that if the current token is tagged as a noun (NN) and has a 3-character suffix of “ing”, then the tag should be a verb (VBG). On the other hand, contextual rules uses only the preceding or proceeding tags and hence, must be applied after lexical rules for effectiveness. The contextual rule “RB JJ NEXTTAG NN” defines that an abverbial tag (RB) should be changed to an adjective (JJ) if the next token was tagged as a noun (NN). A table of Penn Treebank Tag Set (Marcus et al., 1993) without punctuation tags is given in Table 1.

By tagging, the complexity of an English sentence (ie, the number of ways an English sentence can be grammatically constructed with virtually unlimited words and unlimited ideas) was collapsed into a sequence of part-of-speech tags, in this case, Penn TreeBank Tag Set (Marcus et al., 1993), with only about 40 tags. Therefore, tagging reduced the large number of English words to about 40 “words” or tags.

Tag	Description	Tag	Description
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determinant	RBR	Adverb, comparative
EX	Existential *there*	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to

Tag	Description	Tag	Description
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBN	Verb, past participle
NN	Noun, singular or mass	VBG	Verb, gerund or present participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Table 1: Penn Treebank Tag Set without Punctuation Tags (Adapted from (Marcus et al., 1993))

Generally, an English sentence is composed of a noun phrase, a verb, and a verb phase, where the verb phrase may be reduced into more noun phrases, verbs, and verb phrases. More precisely, the English language is an example of subject-verb-object typology structure, which accounts for 75% of all languages in the world (Crystal, 1997). This concept of English sentence structure is used to process a tagged sentence into higher-order structures of phrases by a process of chunking, which is a precursor to the extraction of semantic relationships of nouns into SVO structure. Using only the sequence of tags, chunking was performed as a recursive 4-step process: protecting verbs, recognition of noun phrases, unprotecting verbs and recognition of verb phrases. Firstly, verb tags (VBD, VBG and VBN) were protected by suffixing the tags. The main purpose was to prevent interference in recognizing noun phrases. Secondly, noun phrases were recognized by the following regular expression pattern of tags:

```
((((PDT )?(DT |PRP[$] |WDT |WP[$] )(VBG |VBD |VBN |JJ |JJR |JJS |, |CC |NN |NNS |NNP
|NNPS |CD )*(NN |NNS |NNP |NNPS |CD )+)|((PDT )?(JJ |JJR |JJS |, |CC |NN |NNS |NNP
|NNPS |CD )*(NN |NNS |NNP |NNPS |CD )+)|EX |PRP |WP |WDT )POS )?(((PDT )?(DT |PRP[$]
|WDT |WP[$] )(VBG |VBD |VBN |JJ |JJR |JJS |, |CC |NN |NNS |NNP |NNPS |CD )*(NN |NNS
|NNP |NNPS |CD )+)|((PDT )?(JJ |JJR |JJS |, |CC |NN |NNS |NNP |NNPS |CD )*(NN |NNS
|NNP |NNPS |CD )+)|EX |PRP |WP |WDT )
```

Thirdly, the protected verb tags in the first step were de-protected by removing the suffix appended onto the tags. Lastly, verb phrases were recognized by the following regular expression:

```
(RB |RBR |RBS |WRB )*(MD )?(RB |RBR |RBS |WRB )*(VB |VBD |VBG |VBN |VBP |VBZ )(VB
|VBD |VBG |VBN |VBP |VBZ |RB |RBR |RBS |WRB )*(RP )?(TO (RB )*(VB |VBN )(RP )?)?
```

After chunking, each word (token) was stemmed into its root or infinite form. Firstly, each word was matched against a set of rules for specific stemming. For example, the rule “dehydrogenised verb dehydrogenate” defines that if the word “dehydrogenised” was tagged as a verb (VBD, VBG and VBN tags), it would be stemmed into “dehydrogenate”. Similarly, the words “binds”, “binding” and “bounded” were stemmed to “bind”. Secondly, irregular words which could not be stemmed by removal of prefixes

and suffixes, such as “calves” and “cervices”, were stemmed by a pre-defined dictionary. Lastly, stemming was done by simple removal of prefixes or suffixes from the word based on a list of common prefixes or suffixes. For example, “regards” and “regarding” were both stemmed into “regard”.

Given the general nature of an English sentence is an aggregation of noun phrase, a verb, and a verb phase, where the verb phrase may be reduced into more noun phrases, verbs, and verb phrases, each verb phrase may be taken as a sentence by itself. This allowed for recursive processing of a chunked-stemmed sentence into SVO(s) by a 3-step process. Firstly, the first terminal noun phrase, delimited by “(NX” and “NX)” was taken as the subject noun. Secondly, proceeding from the first terminal noun phrase, the first terminal verb would be taken as the verb in the SVO. Lastly, the rest of the phrase was scanned for terminal noun phrases and would be taken as the object(s). The recursive nature of SVO extraction also meant that the subject, verb, and object(s) will be contiguous, which had been demonstrated to have better precision than non-contiguous SVOs (Masseroli et al., 2006).

2.3 Protein-Protein Binding Finding

The protein-protein binding finder module is a data miner for protein-protein binding interaction assertions from the entire set of subject-relation-object (SVO) assertions from the text analysis process using apriori knowledge. That is, the set of proteins of interest must be known, in contrast to an attempt to uncover new protein entities, and their binding relationships with other protein entities, that were not known to the researcher.

Protein-protein binding assertions were extracted in a three step process. Firstly, a set of SVOs was isolated by the presence of the term “bind” in the verb clause resulting in a set of “bind-SVOs” assertions. Non-infinite forms of “bind” (such as, “binding” and “binds”) were not used as verbs were stemmed into their infinite forms during text processing. Secondly, the set of bind-SVOs were further characterized for the presence of protein entities in both subject and object clauses by comparing with the desired list of protein entities. A pairwise isolation of bind-SVOs for protein entities resulted in a set of bind-SVOs, “entity-bind-SVOs”, containing SVOs describing binding relationship between the protein entities. Lastly, entity-bind-SVOs were cleaned so that the subject and object clauses only contains protein entities. For example, “MAPK in the cytoplasm” in the object clause will be reduced to just the entity name “MAPK”, the full subject and object clauses could be used in other information extraction tasks, such as determining protein localization, but is not explored in this study. This step is required to allow for the construction of network graphs, such as using Graphviz, without reference to the list of protein names during construction. Given that protein_entities is the list of desired proteins, table SVO contains the SVO output from MontyLingua and table entity_bind_SVO contains the isolated and cleaned SVOs, the pseudocode for Protein-Protein Binding Finding module is given as:

```
for subject_protein in protein_entities1 to n
       for object_protein in protein_entities1 to n
              insert (pmid, subject_protein, object_protein) into entity_bind_SVO
                     from select pmid
                     from (select * from SVO where verb = 'bind')
                     where subject is containing subject_protein
                     and object is containing object_protein
```

2.4 Evaluating POS Tagging and Information Extraction Performance

The performance of MontyLingua’s part-of-speech tagger, MontyTagger, was evaluated using MedPost

corpus (Smith et al., 2004). The accuracy is determined as the percentage of the number of correctly tagged tokens (words and punctuations) in the total number of tokens (n=182399). MedPost tagger was swapped in place of MontyTagger by modifying MontyLingua's *jist()* and *jist_predicates()* functions to *mpjist()* and *mpjist_predicates()*, giving MedPost-MontyLingua Muscorian:

```
def jist(self,text):
    sentences = self.split_sentences(text)
    tokenized = map(self.tokenize,sentences)
    tagged = map(self.tag_tokenized,tokenized)
    chunked = map(self.chunk_tagged,tagged)
    extracted = map(self.extract_info,chunked)
    return extracted

def jist_predicates(self,text):
    infos = self.jist(text)
    svoos_list = []
    for info in infos:
        svoos = info['verb_arg_structures_concise']
        svoos_list.append(svoos)
    return svoos_list
```

to

```
def mpjist(self,text):
    sentences = self.split_sentences(text)
    tokenized = map(self.tokenize,sentences)
    sourcefilename = random.random()*1000000000
    outfilename = random.random()*1000000000000
    source = open('temp' + os.sep + str(sourcefilename), 'w')
    source.writelines(tokenized)
    source.close()
    os.popen(os.getcwd() + os.sep + 'medpost/medpost -text  \
    -token -penn < temp' + os.sep + str(sourcefilename) + '> temp' +
    os.sep + str(outfilename))
    mpout = open('temp' + os.sep + str(outfilename), 'r')
    tagged = mpout.readlines()
    mpout.close()
    chunked = map(self.chunk_tagged,tagged)
    extracted = map(self.extract_info,chunked)
    return extracted

def mpjist_predicates(self,text):
    infos = self.mpjist(text)
    svoos_list = []
    for info in infos:
        svoos = info['verb_arg_structures_concise']
        svoos_list.append(svoos)
     return svoos_list
```

MedPost-MontyLingua Muscorian's IE performance was evaluated using Learning Languages in Logic 2005 test data (Cussens and Nedellec, 2005) in the same manner as Muscorian (Ling et al., 2007) and the performances were compared (Figure 2).

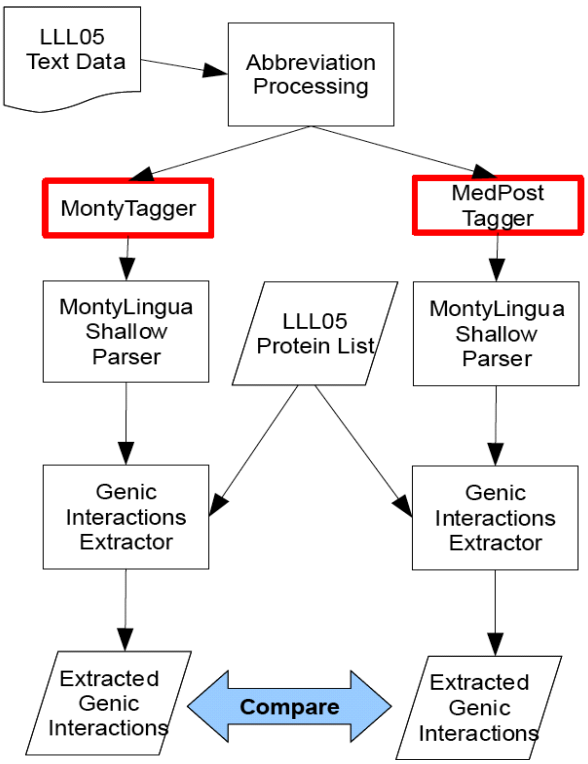

Figure 2: Flowchart of Evaluation Procedure for Muscorian with Native MontyLingua and MedPost-MontyLingua.
LLL05 test data was processed for abbreviations before feeding into each system and the extracted genic interactions (output) were evaluated for precision and recall.

2.5 Analysis of POS Tagging Errors

Wrongly tagged tokens from MontyTagger's output were first grouped by their original tags in MedPost corpus (Smith et al., 2004), then sub-grouped by MontyTagger's assigned tags (the wrong tag) and arranged in decreasing order based on the numbers of tags in both main and sub-group. First 80% of the tags in the main group where first 90% of the wrongly assigned tags were chosen for further error analysis. Each of the pairs of original tag and wrongly assigned tag were analysed with respect to the regular expressions in MontyREChunker (Ling et al., 2007), the shallow parser in MontyLingua, for the effects of the wrongly assigned tags on the operations of the shallow parser.

3 Experimental Results

Four experiments were carried out to evaluate the performance of Muscorian and demonstrate the flexibility of the two-layered generalization-specialization approach in constructing systems that could be readily be adapted to related problems. The results are summarized in Table 2.

	LLL05 Directional	*LLL05 Un-directional*	*Protein-Protein Binding*	*Protein-Protein Activation*
Precision	55.8%	86.1%	88.1%	90.7%
Recall	19.8%	30.7%	Not measured	Not measured

Table 2: Summary of the Experimental Results Comparing the Precision and Recall Measures.

3.1 Benchmarking Muscorian Performance

The performance of Muscorian, in terms of precision and recall, could only be evaluated using a defined data set with known results. For such purpose, the data set for Learning Languages in Logic 2005 (LLL05) (Cussens, 2005) was used to benchmark Muscorian on genic interactions, which is a superset of protein-protein binding interactions. LLL05 had defined a genic interaction as an interaction between 2 entities (agent and target) but the nature of interaction was not considered under the challenge task. LLL05 provided a list of protein entities found in the data set, which was used to filter subject-relation-object assertions from text analysis (MontyLingua) output where both subject and object contained protein entities in the given list. The filtered list of assertions was evaluated for precision and recall, which was found to be 55.6% and 19.8% respectively.

LLL05 required that the agent and target (subject and object) to be in the correct direction, making it a vector quality. However, this requirement was not biologically significant to protein-protein binding interactions, which is scalar. For example, “X binds to Y” and “Y binds to X” have no biological difference. Hence, this requirement of directionality was eliminated and the precision and recall was 86.1% and 30.7% respectively.

3.2 Evaluating POS Tagging and Information Extraction Performance

Evaluating MontyTagger on MedTag corpus demonstrated correct tagging in 151663 of the tags representing 83.1% tagging accuracy. Using the LLL05 evaluation corpus, Muscorian with MedPost-MontyLingua on directional relationship was found to be 56.8% precise with 24.8% recall, while nondirectional relationship was estimated to be 81.8% precise with 35.6% recall (Table 3).

	Directional Relationships		***Nondirectional Relationships***	
	MontyLingua	**mpMontyLingua**	**MontyLingua**	**mpMontyLingua**
Precision	55.6%	56.8%	86.1%	81.8%
Recall	19.8%	24.8%	30.7%	35.6%
F-Score	0.292	0.345	0.453	0.496

Table 3: Summary of Muscorian's Performances Evaluated Using Learning Languages in Logic 2005 Data (Cussens, 2005)

3.3 Analysis of POS Tagging Errors

Comparison of the reference tags (MedPost corpus) with the wrongly assigned tags from MontyTagger showed the 30736 wrongly assigned tags (52.3%, n=16067) should be tagged as nouns (tag: 'NN'), 15.8%

(n=4865) should be tagged as 'JJ' (adjectives), and the next four most common wrongly assigned tags were 'NNS' (n=1987, 6.5%), 'SYM' (n=1496, 4.9%), 'VBP' (n=1470, 4.8%), and 'VBD' (n=745, 2.4%). These six reference tags (NN, JJ, NNS, SYM, VBP, VBN) accounted for 26630 (86.6%) of the wrongly assigned tags, while the rest of the errors (n=4106) were distributed across 25 tags. Six tags (TO, :, (,), WP, ,) were correctly assigned in every instance in this evaluation. A tabulation of errors is shown in Table 4. The confusion matrix can be found at http://muscorian.sourceforge.net/data/MedPost_ MontyTagger_ confuse.txt

A deeper analysis was undertaken to examine the errors in each reference tag (tabulated in Table 5). Firstly, by grouping close POS types, for example 'NN', 'NNP', and 'NNS' were all nouns, wrong sub-type assignation, such as 'NN' assigned as 'NNP' and 'NNS' assigned as 'NNP', accounted for 55% of the errors (n=14634). Secondly, 58% (n=2818) of 'JJ' (adjective) errors were resulted by tagging as noun (NN and NNP) while 34.9% (n=1698) of the 'JJ' errors were tagged as verb (VBN and VBG). Thirdly, about 5% (n=941) of 'NN' (noun) errors were tagged as cardinal numbers (CD). Fourthly, plural nouns accounted for 51.6% (n=1026) of 'NNS' (singular noun) errors. Fifthly, 48.7% (n=729) and 39.3% (n=587) of 'SYM' (symbol) errors were either not assigned or assigned as 'NN' (noun) respectively. Lastly, 87% (n=1927) of verb errors (VBP and VBD) were due resolution of tenses, such as non-third party singular present tense (VBP) was assigned as infinite verb form (VB).

Tag	*% Corpus*	*% Error in Total Error*	*% Error in Tag*	*Tag*	*% Corpus*	*% Error in Total Error*	*% Error in Tag*
NN	28.56	52.27	30.84	VBG	0.64	0.06	1.59
IN	13.49	1.08	1.33	:	0.54	0.00	0.00
JJ	10.47	15.81	25.44	MD	0.43	0.01	0.2
DT	7.77	0.56	1.16	WDT	0.45	0.19	6.70
NNS	7.75	6.45	14.03	,	0.39	0.00	0.00
CC	6.66	1.30	3.29	PRP$	0.28	0.01	0.40
.	3.67	0.01	0.03	FW	0.26	0.96	61.39
CD	3.13	2.02	10.84	WRB	0.23	0.59	43.33
VBN	3.05	1.70	10.13	JJR	0.17	0.17	17.74
VBD	2.81	2.42	14.56	NNP	0.14	0.03	3.53
RB	2.57	1.72	9.49	EX	0.08	0.01	1.38
)	1.89	0.00	0.00	POS	0.06	0.06	15.31
(	1.88	0.00	0.00	WP	0.06	0.00	0.00
VBP	1.98	4.78	41.26	JJS	0.05	0.02	6.60
TO	1.55	0.00	0.00	RBS	0.05	0.01	4.40
VBZ	1.54	0.45	5.20	“	0.03	0.19	100.00
SYM	1.07	4.87	76.43	``	0.03	0.19	100.00
PRP	0.88	1.61	30.59	PDT	0.02	0.11	100.00
VB	0.74	0.05	1.11	RBR	0.01	0.03	44.44

Table 4: Percentage Breakdown of POS Tags in MedTag corpus and Errors in MontyTagger as Percentage of POS Tags Assignation. This table tabulates the POS tagging errors made by MontyTagger on MedTag corpus and the order is according to the abundance of each tag in the MedTag corpus. For example, 'NN' is the most abundant tag accounting for 28.56% or 52093 of MedTag corpus of 182399 tokens. Of which, 3084% (16067 of 52093) of the 'NN' tokens in MedTag corpus were wrongly assigned to a different POS tag by MontyTagger which accounted for 52.27% of the total wrongly assigned POS tag of 30736 tokens.

Reference Tag	*Wrongly Assigned Tag*	*Number of Wrong Assignation*	*Cummulative Frequency for Reference Tag*	*Impact on Shallow Parsing?*
NN (16067)	NNP	10865	67.6%	No, NNP was an alternative match to NN in noun phrase recognition
	JJ	2527	83.4%	No, JJ was an alternative match to NN in noun phrase recognition
	CD	941	89.2%	No, CD was an alternative match to NN in noun phrase recognition
	VBG	812	94.3%	Yes, protected verb tag
JJ (4865)	NN	1600	32.9%	No, NN was an alternative match to JJ in noun phrase recognition
	NNP	1218	58.0%	No, NNP was an alternative match to JJ in noun phrase recognition
	VBN	1170	82.0%	Yes, protected verb tag
	VBG	528	92.8%	Yes, protected verb tag
NNS (1987)	NNP	1026	51.6%	No, NNP was an alternative match to NNS in noun phrase recognition
	NN	701	86.9%	No, NN was an alternative match to NNS in noun phrase recognition
	VBZ	128	93.4%	No, VBZ was an alternative match to NNS in noun phrase and was not a protected verb tag
SYM (1496)	Not Assigned	729	48.7%	No, tokens not tagged were non-existent and SYM was not used in shallow parsing
	NN	587	88.0%	Yes, NN was matched in noun phrase
	-	115	95.7%	No, both tags was not used in shallow parsing
VBP (1470)	VB	1249	85.0%	Yes, mandatory requirement of VB in verb phrase
	NN	178	97.1%	No, NN was an alternative match to VBP in noun phrase
VBD (745)	VBN	678	91.0%	Yes, mandatory requirement of VBN in verb phrase
	JJ	34	95.6%	Yes, protected verb tag

Table 5: Error Breakdown and Analysis on the Effects of Six Most Commonly Mis-Assigned POS Tags. Six reference tags; NN, JJ, NNS, SYM, VBP, and VBD; which accounted for 86.6% of all wrong POS assignation by MontyTagger were chosen and in each tag, the assigned tags which accounted for 90% of the errors were chosen for further analysis. For example, of 16067 tags that were tagged as 'NN' in MedTag corpus, MontyTagger wrongly tagged 10865 tokens as 'NNP' and has no effect on shallow parsing, 2527 tokens as 'JJ' and has no effect on shallow parsing, 941 tokens as 'CD' and has no effect on shallow parsing, and 812 tokens as 'VBG' with an effect on shallow parsing. These 4 wrong tagging accounted for 94.3% of all 'NN' tag errors. This also meant that 922 'NN' tag errors (5.7%) were not further analyzed. A complete confusion matrix is given in http://muscorian.sourceforge.net/data/MedPost_MontyTagger_confuse.txt

Error breakdown (in Table 5) demonstrated erroneous POS tagging by MontyTagger in 31 tags, with 6 tags having no errors. A total of 6 of the 32 tags (19.4%) accounted for 86.6% (n=26630) of the total errors and were chosen for further analysis where each wrongly assigned tag was examined to deduce whether there is an effect on the shallow parsing process. For example, the beginning of a verb phrase is determined by the regular expression `(RB |RBR |RBS |WRB)`. This suggests that a RB token that is

erroneous tagged to `RBR`, `RBS` or `WRB` will not impact on the shallow parsing process. We term this as POS tagging error nullification. Applying these rules to each of the examined erroneous tags (86.6% of the errors), it was found that 78.6% of the errors had no effect on shallow parsing.

3.4 Verifying Protein-Protein Binding Interactions

Precision of Muscorian for mining protein-protein binding interactions from published abstracts was evaluated by manual verification of a sample of assertions (n=135) yielded by the protein-protein binding finder module against the original abstracts. Each of the sampled assertions was assumed to be atomic, in the form of "X binds Y". In cases where there were more than one target, such as "X binds Y and Z", they would be reduced to atomic assertions. In this case, "X binds Y and Z" would be reduced to 2 assertions, "X bind Y" and "X bind Z". These were then checked with the original abstract, traceable by the PubMed IDs, and precision was measured as the ratio of the number of correct assertions to the number of sampled atomic assertions (which is 135). A 95% confidence interval was estimated by bootstrapping (re-sampling with replacement) (Efron and Tibshirani, 1986) of the manual verification results. Our results suggested a precision of 88.1%, with a 95% confidence interval between 82.4% to 93.7%.

An IE trial was performed using the Protein-Protein Binding Finding module to search for the binding partners of CREB and insulin receptor and a sample network diagram of the results are shown in Figure 3 and 4 respectively.

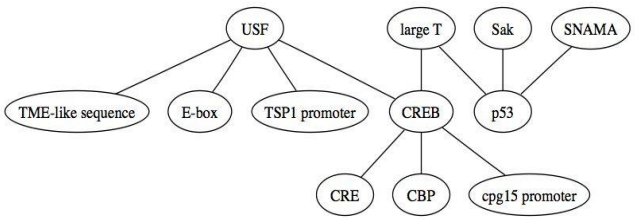

Figure 3: Preliminary Protein Binding Network of CREB

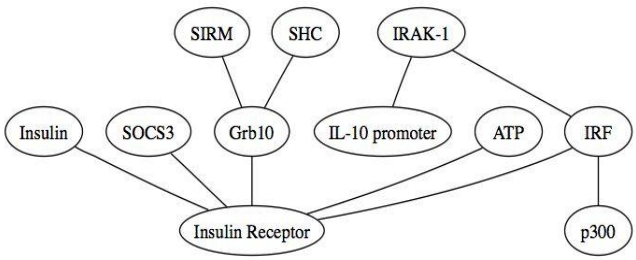

Figure 4: Preliminary Protein Binding Network of Insulin Receptor

3.5 Large Scale Mining of Protein-Protein Binding Interactions

A large scale mining of protein-protein binding interactions was carried out using all of the PubMed abstracts on mouse (about 860000 abstracts), which were obtained using "mouse" as the keyword for searches, with a predefined set of about 3500 abbreviated protein entities as the list of proteins of interest (available from http://cvs.sourceforge.net/viewcvs.py/ib-dwb/muscorian-data/protein_accession.csv?

rev=1.2&view=markup). In this experiment, the primary aim was to apply Muscorian to large data set and the secondary aim was to look for multiple occurrences of the same interactions as multiple occurrences might greatly improve precision confidence.

For example, given our lower confidence estimate that the precision of Muscorian with respect to mining protein-protein binding interactions is 82%, which means that every binding assertion has an 18% likelihood of not having a corresponding representation in the published abstracts. However, if 2 abstracts yielded the same binding assertion, the probability of both being wrong was reduced to 3.2% (0.18^2), and the corresponding probability that at least one of the 2 assertions was correctly represented was 96.8% ($1-0.18^2$). The more times the same assertion was extracted from multiple sources text (abstracts), the higher the possibility that the mined interaction was represented at least once in the set of abstracts. For example, if 5 abstracts yielded the same assertion, the possibility that at least one of the 5 assertions was correctly represented would be 99.98% ($1-0.18^5$).

Our experiment mined a total of 9803 unique protein-protein binding interactions, of which 7049 binding interactions were from one abstract (P=82%), 1297 binding interactions were from two abstracts (P=96.8%), 516 binding interactions were from three abstracts (P=99.4%), 235 binding interactions were from four abstracts (P=99.9%), 164 binding interactions were from five abstracts (P=99.98%), 105 binding interactions were from six abstracts (P=99.997%), 69 binding interactions were from seven abstracts (P=99.9993%), 398 binding interactions were from more than seven abstracts (P>99.9993%).

3.6 Pilot Study – Protein-Protein Activation Interactions

In order to demonstrate the adaptability of our proposed two-layered model, a small pilot study for mining protein-protein activation interactions was carried out. For this study, the protein-protein binding finder module, the data mining module for mining protein-protein binding interaction, was replaced with a protein-protein activation finder module.

The protein-protein activation finder was semantically similar to the original protein-protein binding finder module as described in Section 2.3.3 previously. The only difference was that raw assertion output from MontyLingua was filtered for activation-related assertions, instead of binding-related assertions, before analysis for the presence of protein names in both subject and object nouns from a pre-defined list of proteins of interest. For example, by modifying the Protein-Protein Binding Finding module to look for the verb 'activate' instead of 'bind', it can then be used for mining protein-protein activation interactions. A trial was done for insulin activation and a subgraph is illustrated in Figure 5 below.

The precision measure of Muscorian for mining protein-protein activation interactions was calculated using identical means as described for protein-protein binding interactions. Using a sample of 85 atomic assertions, the precision of Muscorian for mining protein-protein activation interactions was estimated to be 90.7%, with a 95% confidence interval of precision between 84.7% to 96.4% by bootstrapping (Efron and Tibshirani, 1986).

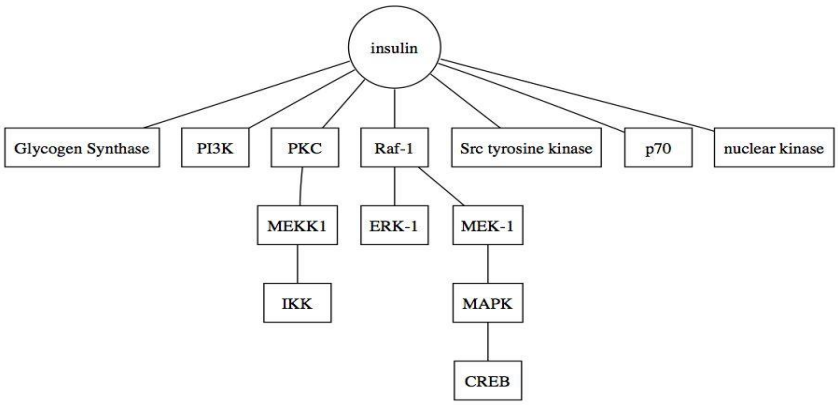

Figure 5: Preliminary Protein Activation Network of Insulin

4 Discussion

New research articles in gene expression regulation networks, protein-protein interactions and protein docking are emerging at a rate faster than what most biologists can manage to extract the data and generate working pathways. Information extraction technologies have been successfully used to process research text and automate fact extraction (Abulaish and Dey, 2007). Previous studies in biological text mining have developed specialized text processing tools and adapted generic tools to relatively good performance of more than 80% in precision (Chiang and Yu, 2004; Daraselia et al., 2004; Jensen et al., 2006; Santos et al., 2005). However, either specialized tool development or modifying existing tools often require much effort (Jensen et al., 2006). The need to modify existing tools has not been formally tested and the possibility of using an un-modified generic text processor for biological text for the purpose of extracting protein-protein interaction remains unresolved. Using a two-layered approach (Novichkova et al., 2003) of generalizing biological text into a structured intermediate form, followed by specialized data mining, we present Muscorian, which uses MontyLingua natively in the generalized layer, as a tool for extracting either protein-protein or genic interactions from about 860000 published biological abstracts.

Benchmarking Muscorian against LLL05, a tested data set, demonstrated a precision of 55.6%, which is about 5% higher than that reported in the conference and a recall of 19.7% is similar to that reported by other participants of LLL05 (Cussens, 2005). This may be due to the emphasis of LLL05 on F-measure, which is the harmonic mean of precision and recall, rather than putting more emphasis on precision. Nevertheless, this also suggested that Muscorian is able to perform text analysis for the purpose of extracting genic interactions effectively, which is comparable to specialized systems reported in LLL05. In addition, directionality of genic interactions was not a concern for protein-protein binding interactions as binding interaction is scalar rather than vector. By eliminating directionality of genic interactions, the precision and recall of Muscorian was 86.1% and 30.7% respectively. This suggested that Muscorian is a suitable tool for mining quality genic interactions from biological text compared to other tools reported in LLL05 (Cussens, 2005).

However, this contradicts the common view that "*error propagation through cascades of processors may in aggregate severely degrade performance on the final task*" as stated in the *Call for Papers for the Tenth Conference on Natural Language Processing 2006 (CoNLL-X)*. Tateisi and Tsujii (Tateisi and Tsujii,

2004) have demonstrated that generic POS taggers are only about 83% accurate when used to tag biomedical text. This suggests that MontyTagger, the generic POS tagger in MontyLingua, is unlikely to perform as well as taggers trained on biomedical text, such as MedPost (Smith et al., 2004). Therefore, it is likely that the above mentioned contradiction is resolved at the step immediately downstream to POS tagging, the shallow parsing. In MontyLingua shallow parsing, the input sentence is broken into noun phrase and verb phrase. The process of shallow parsing can be seen as a collapse of a sequence of POS tags into 2 groups; hence, we expect high level of permissible substitution of POS tags within related classes. We term this permissible substitution as "alternate POS tag use".

MontyLingua's POS tagger, MontyTagger, was substituted with a specialized biomedical POS tagger, MedPost (Smith et al., 2004). The precision and recall of MedPost-MontyLingua Muscorian evaluated using the LLL05 data set (Cussens and Nedellec, 2005) were 56.8% and 24.8% (F-score = 0.35) respectively for directional interactions, and 81.8% and 35.6% (F-score = 0.50) respectively for nondirectional interaction. Our results showed that using MedPost in place of MontyLingua's POS tagger, MontyTagger, had improved the F-score by about 5% in both directional and nondirectional interactions extraction, and recall (24.8% versus 19.7% and 35.6% versus 30.7%).

An initial evaluation of MontyTagger on MedTag Corpus (Smith et al., 2004) indicated 83.1% accuracy, which was considerably less than from MedPost's reported accuracy of 96.9% (Smith et al., 2004) and was close to the 83.0% tagging accuracy of a generic POS tagger on biomedical text (Tateisi and Tsujii, 2004). This result was expected as MontyTagger was not developed for biomedical text (Ling, 2006).

The POS tagging errors were expected to impact on performance of the entire text processing pipeline but this was not observed in our results. Instead, the precision of un-modified-MontyLingua Muscorian was comparable to that of MedPost-MontyLingua Muscorian on directional genic interactions (55.6% versus 56.8%) and un-modified-MontyLingua Muscorian outperformed MedPost-MontyLingua Muscorian on nondirectional genic interactions (86.1% versus 81.8%). Taken collectively the precision of both system and their respective POS tagging accuracies, seemed contradictory to general expectations as stated in the *Call for Papers for the Tenth Conference on Natural Language Processing 2006 (CoNLL-X)*.

An error analysis on MontyTagger was carried out in attempt to provide insight into resolving this contradiction. A likely hypothesis to explain why POS tagging errors did not derail the entire text processing pipeline was that the errors were nullified post-tagging. Text processing is used in Muscorian as a means to convert unstructured text into structured form for data mining - an extremely limited use of natural language processing compared to more complex uses, such as automated translation. As mentioned previously, POS tagging can be seen as a process of mapping potentially infinite number of words in the English language into a finite set of tags, based on their syntactic meanings. Shallow parsing, also known as chunking, can then be seen as a process which examines the sequence of tags and splits them into semantic phrases, of which verb phrase and noun phrase are of interest in this case. Given that MontyLingua's shallow parser parses the sequence of tags into 3 types of phrases (verb, noun, and adjectives), it is conceivable that a number of POS errors have no effect on shallow parsing.

Of the 182399 token in MedTag Corpus (Smith et al., 2004), 30736 were erroneously tagged by MontyTagger (16.9% error) spreading over 40 tags. The top 6 most common tag errors accounted for 86.6% of the total errors and were chosen for further evaluation. In each of the 6 most abundant error tags, the top 95% of the errors were examined.

The effects of each type of errors, such as 'NN' wrongly tagged to 'NNP', were examined by analyzing the routines for shallow parsing which uses Regular Expressions. It was found that in 26630 of the examined POS tagging errors, 20928 (78.6% of 26630) had no effect on the chunking process and the remaining 5703 errors adversely affected shallow parsing, which might account for lower recall of un-

modified-MontyLingua Muscorian as compared to MedPost-MontyLigua Muscorian.

Therefore, despite a low POS tagging accuracy of 83.1% by MontyTagger, more than three-quarters of the errors had no detrimental effect on chunking, suggesting a "functional POS tagging accuracy" of at least 94.6%, which was relatively close to MedPost's reported 97% accuracy (Smith et al., 2004). This apparent high "functional POS tagging performance" despite poor actual tagging accuracy might be the reason to explain un-modified-MontyLingua Muscorian's good performance in LLL05 test (Cussens and Nedellec, 2005) despite poor tagging accuracy compared to MedPost-MontyLigua Muscorian. This suggested that the nature of POS tagging errors might be more important than a single measure of POS tagging accuracy in a specific use of generic text processing tools where a shallow parser is involved. Therefore, it can be inferred that applications of biomedical literature analysis where a shallow parser is likely to be involved, such as extracting entity interactions and protein or molecule localization, POS tagging errors may not result in a decline in system performance.

At the same time, it is known that building domain-specific text processing tools requires much manual efforts (Jensen et al., 2006) suggesting that the cost and effort needed to train taggers specifically for biomedical text may not be needed, depending on the target application. However, it should also be cautioned that other applications or systems that do not involve shallow parser, such as Arizona Relation Parser (Daniel et al., 2004) which uses full sentence parsing, are likely to benefit from superior POS tagging accuracy of MedPost (Smith et al., 2004) and may experience degraded results from tagging errors.

Our results on protein-protein binding and activation interactions show the insulin receptor binds to IL-10 promoter through IRF and IRAK-1, which is an important insulin receptor signalling pathway. In addition, our data shows insulin activates CREB via Raf-1, MEK-1 and MAPK, which is consistent with the MAP kinase pathway. Combining these data (Figures 3 and 4) indicated that insulin activates CREB via MAP kinase pathway, and CREB binds to cpg15 promoter in the nucleus. A simple keyword search on PubMed, using the term "cpg15 AND insulin" (done on 15th of May, 2010), did not yield any results, suggesting that the effects of insulin on cpg15, also known as neuritin (Cappelletti et al., 2007), had not been studied thoroughly. This might also suggest limited knowledge shared between insulin investigators and cpg15 investigators as suggested by Don Swanson in his classical paper describing the links between fish oil and Raynaud's syndrome (Swanson, 1986). Neuritin is a relatively new research area with less than 35 papers published (as of 15th of May, 2010) and had been implicated as a lead for neural network re-establishment (Han et al., 2007), suggesting potential collaborations between endocrinologists and neurologists.

Our experiments in extracting two different forms of relations demonstrated that despite using specialized dictionaries in the generalized layer, it is still general to the extend that specific application (the type of relationships to extract) was not built into the generalized layer.

At the same time, these 2 experiments also illustrated the relative ease in re-targeting the system for extracting another form of relationship by modifying the specialized layer. The Protein-Protein Activation Finder module is a slight modification of the original Protein-Protein Binding Finder module where the original SQL statement that selects 'bind'-related SVOs from total SVOs, "*select * from SVO where verb = 'bind'*", was changed to "*select * from SVO where verb = 'activate'*" to select for 'activation'-related SVOs from total SVOs. Hence, it is plausible that similar changes may suffice for extracting other relationships, such as 'inhibition'. This relative ease of re-targeting the system for extracting other relationships also demonstrated the robustness of the generalization layer, as implied by Novichkova et. al. (Novichkova et al., 2003) – "*the adaptability of the system to related problems other than the problem the system was designed for*".

Given large numbers of published abstracts, the performance of Muscorian on precision was comparable with published values of BioRAT (58.7%) (David et al., 2004), GIS (84%) (Chiang et al., 2004), Cooper and Kershenbaum (74%) (Cooper and Kershenbaum, 2005) and CONAN (53%) (Malik et al., 2006) while Muscorian's recall was comparable with published values of Arizona Relations Parser (35%) (Daniel et al., 2004) and Daraselia et. al. (21%) (Daraselia et al., 2004). Poor precision was considered unacceptable because incorrect information is more detrimental than missing information (1 - recall) when protein-protein binding interactions were used to support other biological analyses. Muscorian's mediocre recall of 30% (from LLL05 test set evaluation) could be supplemented by the fact that the same interaction could be mentioned or described by multiple abstracts; thus, the actual recall when tested on a large corpus may be higher. For example, 30% recall essentially means a loss of 70% of the information; however, if the same information (in this case, protein interactions) were mentioned in 3 or more abstracts, there is still a reasonable chance to believe that information from at least 1 of the 3 or more abstracts will be extracted. This is supported by our results indicating that almost 30% (2754 of 9803) of binding interactions were extracted from more than one abstract.

Multiple isolation of 2754 binding interactions enabled a higher confidence that these interactions were correctly extracted with reference to the source literature. Based on this analysis, 2754 binding interactions could be assigned higher confidence based on their occurrences (Jenssen et al., 2001), in this case more than 95% chance of being correct based on literature. In addition, the number of multiple interaction occurrences varies inversely with the number of abstracts these interactions were found in is in line with expectation. Although this line of argument is based on the assumption that the appearance of protein names across abstracts were independent, it can be reasonably held as this study uses abstracts rather than full text – abstracts tends to describe what main results of the particular article while the introduction of a full text article tends to be a brief background review of the field. Hence, independence of protein names can be better assumed in abstracts than in full text articles.

An evaluation of a sample of atomic assertions (interactions) of binding and activation interactions between entities was performed by domain experts comparing the assertions with their source abstracts. Both approaches gave similar precision measures and are consistent with the evaluation using LLL05 test set. The ANOVA test demonstrated that there was no significant difference between these three precision measures. Taken together, these evaluations strongly suggested that Muscorian performed with precisions between 86-90% for genic (gene-protein and protein-protein) interactions, which was similar to that reported by studies either modifying existing tools (Santos et al., 2005) or developing specialized tools (Daraselia et al., 2004). This suggested that MontyLingua could be used natively (un-modified), with good precision, to process biological text into structured subject-verb-objects tuples which could be mined for protein interactions.

References

Abulaish, M., and Dey, L. 2007. Biological relation extraction and query answering from MEDLINE abstracts using ontology-based text mining. Data & Knowledge Engineering, 61: 228.

Cappelletti, G., Galbiati, M., Ronchi, C., Maggioni, M.G., Onesto, E., and Poletti, A. 2007. Neuritin (cpg15) enhances the differentiating effect of NGF on neuronal PC12 cells. Journal of Neuroscience Research

Chang, J. T., Schutze, H., and Altman, R. B. 2002. Creating an online dictionary of abbreviations from MEDLINE. Journal of the American Medical Informatics Association 9:612-620.

Chiang, J. H., and Yu, H. C. 2003. MeKE: discovering the functions of gene products from biomedical literature via sentence alignment. Bioinformatics 19:1417-1422.

Chiang, J. H., Yu, H. C., and Hsu, H. J. 2004. GIS: a biomedical text-mining system for gene information discovery. Bioinformatics 20(1):120.

Cooper, J. W., and Kershenbaum, A. 2005. Discovery of protein-protein interactions using a combination of linguistic, statistical and graphical information BMC Bioinformatics 6:143.

Crystal, David. 1997. The Cambridge Encyclopedia of Language, 2nd edition, Cambridge: Cambridge University Press.

Cunningham, H. 2000. Software Architecture for Language Engineering. PhD Thesis. Department of Computer Science: University of Sheffield.

Cussens, J. (ed). 2005. Proceedings of the Learning Languages in Logic Workshop 2005.

Daniel, M. M., Hsinchun, C., Hua, S., and Byron, B. M. 2004. Extracting gene pathway relations using a hybrid grammar: the Arizona Relation Parser. Bioinformatics, 20:3370.

Daraselia, D., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A., and Mazo, I. 2004. Extracting human protein interactions from MEDLINE using a full-sentence parser. Bioinformatics 20: 604-11.

David, P. A. C., Bernard, F. B., William, B. L., and David, T. J. 2004. BioRAT: extracting biological information from full-length papers. Bioinformatics 20:3206.

Efron, B. and Tibshirani, R. 1986. Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. Statistical Science 1:54-75.

Eslick, I., and Liu, H. 2005. Langutils – A natural language toolkit for Common Lisp. Proceedings of the International Conference on Lisp 2005.

Feng, C., Yamashita, F., and Hashida, M. 2007. Automated extraction of information from the literature on chemical-CYP3A4 interactions. Journal of Chemical Information and Modeling, 47**,** 2449-55.

Friedman, C., Alderson, P. O., Austin, J. H., Cimino, J. J., and Johnson, S. B. 1994. A general natural-language text processor for clinical radiology. Journal of the American Medical Informatics Association 1:161-174.

Friedman, C., Kra, P., Yu, H., Krauthammer, M., and Rzhetsky, A. 2001. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics, 17:S74-S82.

Grover, C., Klein, E., Lascarides, A., and Lapata, M. 2002. XML-based NLP Tools for Analysing and Annotating Medical Language. Proc. of the 2nd Int. Workshop on NLP and XML (NLPXML-2002), Taipei, 2002.

Han, Y., Chen, X., Shi, F., Li, S., Huang, J., Xie, M., Hu, L., Hoidal, J.R., and Xu, P. 2007. CPG15, A New Factor Upregulated after Ischemic Brain Injury, Contributes to Neuronal Network Re-Establishment after Glutamate-Induced Injury. Journal of Neurotrauma 24:722-731

Hu, Z., Narayanaswamy, M., Ravikumar, K., Vijay-Shanker, K., and Wu, C. 2005. Literature mining and database annotation of protein phosphorylation using a rule-based system. Bioinformatics 21:2759-2765.

Jang, H., Lim, J., Lim, J. H., Park, S. J., Lee, K. C., and Park, S. H. 2006. Finding the evidence for protein-protein interactions from PubMed abstracts. Bioinformatics, 22**,** e220-6.

Jensen, L. J., Saric, J., and Bork, P. 2006. Literature mining for the biologist: from information retrieval to biological discovery. Nature Review Genetics, 7:119-129.

Jenssen, T. K., Laegreid, A., Komorowski, J., and Hovig, E. 2001. A literature network of human genes for high-throughput analysis of gene expression. Nature Genetics*,* 28, 21-8.

Ling, M. H. T. 2006. An Anthological Review of Research Utilizing MontyLingua, a Python-Based End-to-End Text Processor. The Python Papers 1: 5-12.

Liu, H., and Singh, P. 2004. ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal 22:211-226.

Kim, J. D., Ohta, T., and Tsujii, J. 2008. Corpus annotation for mining biomedical events from literature. BMC Bioinformatics, 9**,** 10.

Malik, R., Franke, L., and Siebes A. 2006. Combination of text-mining algorithms increases the performance. Bioinformatics*,* 22, 2151-2157.

Marcus, M.P., Santorini, B., and Marcinkiewicz, M.A. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, 19:313-330.

Masseroli, M., Kilicoglu, H., Lang, F. M. and Rindflesch, T. 2006 Argument-predicate distance as a filter for enhancing precision in extracting predications on the genetic etiology of disease. BMC Bioinformatics 7: 291.

Nasukawa, T., and Nagono, T. 2001. Text analysis and knowledge mining system. IBM System Journal 40:967-984.

National Library of Medicine. 2003. UMLS Knowledge Sources, 14th edition.

Novichkova, S., Egorov, S., and Daraselia, N. 2003. MedScan, a natural language processing engine for MEDLINE abstracts. Bioinformatics 19:1699-1706.

Rebholz-Schuhmann, D., Kirsch, H., and Couto, F. 2005. Facts from Text - Is Text Mining Ready to Deliver? PLoS Biology,3:e65.

Santos, C., Eggle, D., and States, D. J. 2005. Wnt pathway curation using automated natural language processing: combining statistical methods with partial and full parse for knowledge extraction. Bioinformatics 21:1653-1658.

Sleator, D., and Temperley, D. 1991. Parsing English with a Link Grammar. Proceedings of the 3rd International Workshop on Parsing Technologies.

Smith, L., Rindflesch, T., and Wilbur, WJ. 2004. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics 20: 2320-1.

Swanson, D. R. 1986. Fish oil, Raynaud's syndrome, and undiscovered public knowledge. Perspectives in Biology and Medicine, 30, 7-18.

Tateisi, Y., and Tsujii, J.I. 2004. Part-of-Speech Annotation of Biological Research Abstracts. 4th International Conference on Language Resource and Evaluation (LREC2004).

van Eck, N. J. and van den Berg, J. 2005. A novel algorithm for visualizing concept associations. Proceedings of the 16th Int. Workshop on Database and Expert System Applications (DEXA'05).

Uramoto, N., Matsuzawa, H., Nagano, T., Murakami, A., Takeuchi, H., and Takeda, K. 2004. A text-mining system for knowledge discovery from biomedical documents. IBM System Journal 43:516-533