Gene Interaction Extraction from Biomedical Texts by Sentence Skeletonization

Přemysl Vítovec
vitovpre@fel.cvut.cz
Jiří Kléma
klema@labe.felk.cvut.cz

Czech Technical University in Prague, Faculty of Electrical Engineering,
Department of Cybernetics

Abstract. The presented paper 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. 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.

Keywords: gene interaction extraction, relation mining, text mining

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, text preprocessing needs to be employed. The text preprocessing method (*sentence skeletonization*) discussed in this paper builds on a priori linguistic knowledge.

2 Related Work

The methods commonly applied in the gene interaction extraction task include computational-linguistics based methods (mainly *language parsing*), *rule-based methods* and *machine learning based methods* [26].

Shallow parsing provides only partial decomposition of the sentence structure: part-of-speech tagged words are grouped into non-overlapping chunks of grammatically related words, whose relations are subsequently analyzed [10, 26]. Pustejovsky et al. [18] and Leroy et al. [13] accomplish the analysis using finite state automata. *Deep parsing*, in contrast, considers the entire sentence structure. Ahmed et al. [1] analyze the full parse by assigning predefined syntactic frames to parsed clauses, Skounakis et. al [23] automate the analysis by employing hidden Markov models (empiricist approach [26]). *Rule-based approaches* employ textual rules or patterns encoding relationships between entities [26]. Manually defined rules have been applied e.g. by Blaschke and Valencia [4] or Proux et al. [20]; systems capable of inducing rules automatically have been proposed e.g. by Huang et al. [11] or Hakenberg et al. [9] On the field of *machine learning based methods*, Kraven and Kumlien [7] employ bayesian classifier, Stapley and Benoit [24] use co-occurrence statistics, Airola et al. [2] extend the general graph kernel method introduced by Bunescu and Mooney [6] and construct a custom kernel to be passed to support vector machines.

Instead of being mutually exclusive, the three above principles rather supplement each other, as they each describe a different methodological aspect: parsing focuses on understanding the internal domain *structure*, rules on *encoding* internal dependencies and machine learning on *procedures* of revealing such dependencies. Advanced sequential approaches, like the *episode rules* proposed by Plantevit at al. [17], *encode* in fact the findings of machine learning based *procedures*. The missing *structural* view may be added by employing a reasonable text preprocessing.

The method of the text preprocessing discussed in this paper builds on the work of Miwa et al. [15], Jonnalagadda et al. [12] and Siddhartan [22], who propose various techniques transforming sentences into syntactically simpler structures.

3 Method Description

Text preprocessing discussed here converts a sentence into a set of structurally simpler word sequences called *skeletons*. Each skeleton *estimates* a subset of core semantical and syntactical features of the original sequence. Moreover, the language behind the skeleton is more reliably mirrored by the corresponding word sequence than in case of the original sentence. Thus, the *sequentiality* of skeletons is higher than the *sequentiality* of the original sentence, which makes them more suitable for applying sequential approaches. As a result of estimation, the skeleton set can not be regarded as a decomposition of the original sentence.

Skeletons are constructed following the bottom-up principle: being grounded at the *clause level*, they are further modified at the *sentence level*. The skeleton construction rely mostly on metalingual categories assigned to text by TreeTagger [21].

3.1 Clause Level

Problem Identification

(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 Example 1 demostrates that altering a simple phrase by adequate language components causes the phrase to grow both to the left and to the right. Although there are limitations of such growth given by the demand of understandability, the space of all possible forms of phrases remains infinite. Assuming any semantically relevant sentence, e.g. *g inhibits X*, all the above phrases constitute a lexical paradigm for variable nominal argument *X*. This phenomenon will be referred to as *paradigmatic phrase space complexity*.

(2) *the [G1 activates G2]*

(3) *the expression of [G1 activates G2]*

In Example 2 *G1* binds to predicate *activates* (i.e. to the right), whereas in Example 3 *G1* binds to verbal noun *expression* (i.e. to the left). Therefore, in both sentences the marked subsequence represents different syntagma. This phenomenon will be referred to as *syntagmatic phrase space complexity*.

In conclusion, due to arbitrary phrase space complexity, the *positional distance* in the word sequence does not imply the underlying *language distance*.

Building Principles To deal with the above difficulties, the following principles have been defined as building blocks for the *clause level* text preprocessing:

Phrase structure reduction. The language sentence may be considered as a projection of a multidimensional, non-sequential language structure into a sequence of lexical elements. Backward mapping (i.e. word sequence interpretation) may be extremely difficult without fully qualified language knowledge. However, playing with paradigmatic relations (Example 1) reveals, that *semantically related* structures of different structural complexity can be placed at the same position, i.e. complex structures may be replaced with simpler ones without significant information loss. Applying recursively such transformations results in

clause level *skeleton*, which is assumed to hold or at least represent the core of the original clauses.

Operation atomicity. Working with the sentence as a whole implies facing the potential complexity of a general sentence. This can be avoided by operating on the lowest syntactical level: simplifying transformations considering only the closest context rely on *what we almost certainly know about the local language*. Moreover, atomicity and linguistic relevancy allow for heuristic qualifying and quantifying the additive semantic shifts caused by these transformations. However, the semantic shifts may be negligible, as in Example 4, where only attributes and appositional adjuncts are removed.

(4) $gene_{(att)}$ *G in eukaryotic*$_{(att)}$ *cells* $\rightarrow$ *G [in cells]*$_{(adj)}$ $\rightarrow$ *G*

Gene name propagation. Simplifying a word sequence can not proceed without removing words considered irrelevant. Language relevancy of words is closely related to their position in the phrase: word in *head* position holds the core meaning of the phrase and represents the minimal member of the corresponding paradigm, words at other position are linguistically less relevant. However, the language relevancy may conflict with the relevancy rising from the gene interaction extraction task, since gene entity names may occupy also attributive or adjunct positions. Therefore, to prevent the gene entity names from being removed, they need to be *propagated* to more stable positions. However, this procedure causes non-negligible (though measurable) shift in the semantic space of the given sentence (Example 5).

(5) $G_{(att)}$ *expression* $\rightarrow$ *G; expression of* $G_{(adj)}$ $\rightarrow$ *G*

Proximity assumption. Due to declared operation atomicity, the word sequence is never seen as a whole, but always locally. As a result, especially conjunction words may be ambiguous: being given only the immediate neighborhood, it may by hard to determine, what subsequences of the sentence actually constitute the arguments of the conjunction word. However, in case that both left and right neighboring words are of the same or related class, the following principle is applied: unless there is special reason for not treating them as arguments of the conjunction word (Example 7), they are treated as such (Example 6).

(6) *G1 activates [G2 and G3]* $\rightarrow$ *G1 activates G2+G3*

(7) *... expression [of G1] and [G2 activates]...*

The clause level transformations designed according to the above principles are summarized in Table 1.

Skeleton Construction The process of finding the clause skeletons can be roughly summarized into four steps: (1) reduce *noun chunks* into *minimal chunks* using the *left removal* and *forward propagation*; resolve *appositions* and *coordinations*, which results to a *nominal skeleton*. The remaining two steps are

Table 1. Clause level transformations. Legend: NC ∼ applicable within noun chunk; VC ∼ applicable within verb chunk; NCS ∼ applicable to noun chunk sequences.

Transformation	Cost	Type	Description
Leftřemoval(LR)	∼ 0	NC	Attribute removed, head preserved: *cell gene → gene; gene G → G*
Forward propagation (FP)	> 0	NC	Attribute moved to head position: *G expression → G*
Verb reduction	∼ 0	VC	Left verb form removed: *has activated → activated; is able to activate → activate*
Apposition reduction	∼ 0	NC	Concatenation + LR and FP *gene, G, → G; G1, G2, → G1+G2*
Coordination reduction	∼ 0	NC	Coordination + LR and FP *gene and G → G; gene and protein → protein; G1 and G2 → G1+G2*
Right removal	∼ 0	NCS	Appositional adjunct removed: *gene in cells → gene; G in cells → G*
Backward propagation	> 0	NCS	Appositional adjunct moved to preceding head: *expression if G → G*

specific to *verb skeletons*: (3) resolve nominal structures, mainly using the *right removal* and *backward propagation*; (4) resolve *appositions* and *coordinations* more freely. 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 chunk sequences*; (II) reduce the *noun chunk sequences* (if possible) to such forms which can be passed as arguments to clause verb predicate.

(8) *expression of G1 gene activates G2 induced protein G3 in mouse cells*
→ *expression of G activates G2 induced G3* (nominal skeleton)
→ *G1 activates G3* (verb skeleton)

Nominal structures are resolved and passed as arguments to clause predicates, i.e. nominal structures are subordinated to verb predicates. However, subset of nouns and adjectives may be also employed as predicates, i.e. they bind arguments: nouns, *gene entity words* or other nominal predicates. Nominal structures built around nominal predicates are saved in nominal skeletons before they are dissolved to become verb arguments. However, if they appear as arguments of a nominal predicate, they need to be stored in another nominal skeleton, before they are dissolved to become arguments of the superior nominal predicates. The procedure dealing with nested nominal predicates is not covered here due to limited space.

3.2 Sentence Level

Problem Identification

(9) *... it activates G2; ... and activates G2*

Even though the sentence stubs 9 seem incomplete with respect to their subjects, none of them has actually empty subject argument: both pronoun and unstated subject are valid syntactical subjects. However, these elements do not hold their own semantics; they only point to another language elements, thus propagating the once declared content to another sentence locations. The propagation naturally implies the *binding ability*: elements one representing the *holder* of the semantics and one the *pointer* (either explicit, or implicit) are clearly related to each other. This phenomenon will be referred to as *the existence of language pointers*.

(10) $[G1\ activating_{nominal}\ G2]\ interacts_{finite}\ with\ G3$

The predicative power of verb allows it to operate as top level node which divides clause in two regions containing (mainly nominal) arguments of the given verb. However, nominal verb forms (past participles, *ing*-forms) occur also within these regions (Example 10), while still preserving the verb syntactic behaviour. Moreover, some nominal verb structures tend to constitute their own subordinated clauses. An error in determining, which verb holds the role of sentence predicate, may lead towards loss of the sentence integrity. This phenomenon will be referred to as *existence of nominal verb forms*.

Building Principles The skeletons grounded at the clause level are further modified at the sentence level according to the following principles:

Mapping language pointers to corresponding values. Pointers need to be replaced by the elements they are pointing to, in order to prevent sequential algorithm from missing relation the element is involved in through this pointer. Correct mapping requires deep knowledge of discourse and information structure of general English sentence. Currently, the mapping employs only simple heuristic rules.

Mapping nominal verb forms to potential interaction predicates. To preserve the sentence semantical integrity, nominal verb forms are mapped to potential interaction predicates: verbs, nouns or adjectives with respect to current local context. The mapping follows complex heuristic rules extracted manually from random subsets of biomedical abstracts.

Assumption of neutral thematic structure. Scientific texts are assumed to follow the neutral textual principle: an entity is referred to not until it has been introduced. Therefore, only pointers pointing to the left are taken into account.

Operation minimality. In contrast to the clause level, transformations at the sentence level can not be evaluated using a reliable language based measure, since the context which needs to be covered is too large and therefore too versatile. To minimize the probability of making errors, only a minimum number of steps

are applied. Therefore, only those mappings are carried out, which cause any predicate to get two arguments, each containing at least one gene entity name.

The sentence level transformations designed according to the above principles are summarized in Table 2.

Table 2. Sentence level transformations. Legend: N ∼ within noun chunks, C ∼ within single clause, CC ∼ in context of two coordinate clauses; CS ∼ in context of clause and its subordinated clause.

Transf.	Appl.	Description
Explicit pointer mapping	N, CC, CS	Personal and possessive pronouns are mapped to gene entity names *... G1 consists of three exons and [it → G1] activates...* *... G1 and [its → G1] activation...*
Implicit pointer mapping	CC, C(S)	Unstated subjects are mapped to gene entity names *... G1 activates G2 and [¡none¿ → G1] associates...* *... G1 activates G2 by [¡none¿ → G1] associating...*
Ing-forms mapping	N, C(S)	Mapping *ing*-forms to nouns, adjectives or verbs
Participle mapping	N, C(S)	Mapping past participles to verbs or adjectives

4 Experiments

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 (grammar is often a fundamental prerequisite for semantic integrity). 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 categories (i.e. singular, plural, proper etc.; first level abstraction) actually assigned to any English noun word by TREETAGGER [21]; i.e. any noun may be substituted for the *noun*-token.

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

4.2 Experimental Data

The resulting sequentialty was evaluated on six biomedical corpora annotated both for gene entites and gene interactions: AIMED [16], CHRISTINE BRUN CORPUS [5], HPRD50 [14], IEPA [3], LLL05 [25] and BC-PPI [8]. All six corpora were handled in the same way according to the following four principles: (I) sentences are stemmed and assigned grammar tags using TREETAGGER [21]; (II) interactions employing more than two gene entities are converted into corresponding number of binary interactions (e.g. one ternary 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 *predicate* is counted only ones in the pattern analysis regardless of how many times it actually appears in the sentence.

4.3 Results

The overall performance of the presented approach in terms of *precision*, *recall* and *F-measure* is given in Table 3.

Table 3. Precision, recall and F-measure for all testing corpora

	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

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 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.

False positives result either from (a) shortcommings of the sentence skeletonization, or (b) shortcommings of the sequential algorithm. (a) Provided that

Table 4. Analysis of false negatives: unhandled structures, confusing factors

	Category	Explanation
1	Incorrect tagging	E.g. *G1 binds@noun 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]*

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. The *atomicity principle* declared at the clause level typically prevents the corresponding transformations from exceeding the allowed level of distortion. Unfortunately, the *minimality principle* declared at the sentence level instead of the *atomicity principle* does not guarantee the same level of control. As a result, the corresponding transformations appear as error contributors more frequently. Moreover, their negative effect is often multiplied by coordinations, which distribute the error to all coordination participants. (b) Errors of the testing algorithm rise mostly from the omission of semantics: not every word holding the position of an interaction predicate does trully descibe an interaction. The overall performance on various corpora (Table 3) depends strongly upon the frequency of such confusing predicate candidates.

The atomicity allows to define a language based distance measure for estimative quantifying the semantic shift: the quantified overall semantic deviation from the original word sequence could be understood as a confidence in the obtained result (skeleton). However, the atomicity is currently declared only at the clause level. Therefore, any distance measure designed for estimating the overall semantic deviation from the original text representation will necessarily mirror exclusively the effect of clause level transformations. Experiments designed to find the optimal maximum allowed deviation by setting non-zero cost for both forward and background propagations (Table 1) proved, that such a measure is not sufficiently informative.

Pyysalo, Airola et al. [19, 2] use very similar approach to evaluate extraction performance of several approaches on five corpora, four of which are used in the presented experiments: AIMed, HPRD50, IEPA and LLL05. A comparison of some of them with the method proposed in this report is given in Table 5. Obviously, the presented approach achieves comparable results, even

though it was targeted only to evaluate the effect of sentence skeletonization and was not seriously meant as a full featured system for gene interaction extraction.

Table 5. Performance comparison. Legend: Graph kernel ∼ SVM based approach [2], RelEx ∼ approach involving deep parsing [19], Skel. + seq. ∼ the presented approach.

		AIMed	HPRD50	IEPA	LLL05
P	Graph kernel	0.529	0.643	0.696	0.725
	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
	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
	RelEx	0.44	0.69	0.67	0.77
	Skel. + seq.	0.48	0.69	0.65	0.79

5 Conclusion and Further Work

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, language phenomena breaking the sentence sequentiality have been identified. To deal with these obstacles, heuristic transformations have been designed, all of which are employed to convert a sentence into a set of skeletons, structures with improved level of sequentiality.. 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 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 pattern mining 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.

Sentence skeletonization will be further improved by applying the atomicity principle also at the *sentence level* and *text level*: this can be achieved by identifying the information flow between pairs of patternalized, i.e. further skeletonized clauses or sentences. Such method should not only solve the mapping problems, but it might also be helpful in dealing with various issues strongly related to pragmatics.

Furthermore, *episode rules*, an advanced general sequential approach proposed by Plantevit et al. [17], will be applied to sentence skeletons in the

gene interaction extraction task. Both lexical and metalingual information should be employed as features to balance the generalization potential and semantical relevancy of extracted rules.

Acknowledgment

The work of Přemysl Vítovec was funded by the Grant Agency of the Czech Technical University in Prague, grant No. SGS11/126/OHK3/TT/13. The work of Jiří Kléma was funded by the Czech Ministry of Education in the framework of the research programme Transdisciplinary Research in the Area of Biomedical Engineering II, MSM 6840770012.

References

1. Syed Toufeeq Ahmed, Deepthi Chidambaram, Hasan Davulcu, and Chitta Baral. Intex: a syntactic role driven protein-protein interaction extractor for bio-medical text. In *ISMB '05: Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*, pages 54–61, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
2. 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.
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. 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.
8. Jörg Hakenberg. *BC-PPI Corpus.* Humboldt-Universität zu Berlin - Institut für Informatik, http://www2.informatik.hu-berlin.de/ hakenber/corpora/. Accessed March 2010.
9. 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.
10. 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.
11. 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.

12. Siddhartha Jonnalagadda, Luis Tari, Jörg Hakenberg, Chitta Baral, and Graciela Gonzalez. Towards effective sentence simplification for automatic processing of biomedical text. *CoRR*, 2010.
13. 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.
14. 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.
15. Makoto Miwa, Rune Saetre, Yusuke Miyao, and Jun'ichi Tsujii. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 788–796, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
16. Raymond J. Mooney. *AiMed*. University of Texas at Austin, https://wiki.inf.ed.ac.uk/TFlex/AiMed. Accessed March 2010.
17. 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.
18. 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.
19. 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.
20. 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.
21. Helmut Schmid. *Treetagger*. Institute for Computational Linguistics of the University of Stuttgart, http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/. Accessed March 2010.
22. Advaith Siddharthan. Syntactic simplification and text cohesion. *Language and Computation*, 4:77–109, 2006.
23. Marios Skounakis, Mark Craven, and Soumya Ray. Hierarchical Hidden Markov Models for Information Extraction. In *IJCAI*, pages 427–433, 2003.
24. 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.
25. Unité Mathématique, Informatique et Génome, http://genome.jouy.inra.fr/texte/-LLLchallenge/. *LLL05 Corpus*. Accessed March 2010.
26. Deyu Zhou and Yulan He. Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*, 41:393–407, 2008.