Danger Mercaderes, RM.; Pla Santamaría, F.; Molina Marco, A.; Rosso, P. (2014). Towards a Protein-Protein Interaction information extraction system: recognizing named entities. Knowledge-Based Systems. 57:104-118. doi:10.1016/j.knosys.2013.12.010.

Towards a Protein-Protein interaction Information Extraction System: recognizing named entities

Roxana Danger[a,*], Ferran Pla[b], Antonio Molina[b], Paolo Rosso[b]

[a] *Dept. of Computing. Imperial College London. South Kensington Campus. UK.*
[b] *Natural Language Engineering and Pattern Recognition (ELiRF), Dpto. de Sistemas Informáticos y Computación, Universitat Politècnica de València. Spain*

Abstract

The majority of biological functions of any living being are related to Protein-Protein Interactions (PPI). PPI discoveries are reported in form of research publications whose volume grows day after day. Consequently, automatic PPI information extraction systems are a pressing need for biologists. In this paper we are mainly concerned with the named entity detection module of PPIES (the PPI Information extraction system we are implementing) which recognizes twelve entity types relevant in PPI context. It is composed of two sub-modules: a dictionary look-up with extensive normalization and acronym detection, and a Conditional Random Field classifier. The dictionary look-up module has been tested with Interaction Method Task (IMT), and it improves by approximately 10% the current solutions that do not use Machine Learning (ML). The second module has been used to create a classifier using the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA'04) data set. It does not use any external resources, or complex or ad-hoc post-processing, and obtains 77.25%, 75.04% and 76.13 for precision, recall, and F1-measure, respectively, improving all previous results obtained for this data set.

Keywords: Biomedical named entity recognition, Protein-Protein Interaction, Dictionary look-up, Machine Learning

1. Introduction

The study of Protein-Protein Interactions (PPI) has become crucial for many research topics in biology, since they are intrinsic to virtually every cellular process ([1]). The majority of PPI information is available in the form of research articles whose volume grows day after day. In order to provide biologists with fast access to all this information, curators from various research institutes are dedicated to extracting the most important descriptions from publications, and

*to whom correspondence should be addressed. This work was developed while the first author was working for the ELiRF Research Group at the Department of Computer Systems and Computation, Universidad Politécnica de Valencia, Spain.

to storing the extracted data on Protein Interaction Databases, such as: the Munich Information Center for Protein Sequence (MIPS) protein interaction Database [2]; the Biomolecular Interaction Network Database (BIND) [3]; the Database of Interacting Proteins (DIP) [4]; the Molecular Interaction Database (MINT) [5]; the protein Interaction Database (IntAct) [6]; the Biological General Repository for Interaction Datasets (BioGRID) [7]; and the Human Protein Reference Database (HPRD) [8].

Currently, the curation load is shared amongst all databases, and is built on the MIMIx [9] (Minimum Information about a Molecular Interaction Experiment) resources, part of the Proteomics Standards Initiative (PSI), of the Human Proteome Organization (HUPO)[1]. The MIMIx resources are composed by the MIMIx guidelines, the PSI-MI XML interchange format, and the corresponding controlled vocabularies for molecular interaction description.

The curated data are regularly interchanged using the common standard PSI-MI extensible markup language (XML). However, expert curators may need a whole day to extract all the relevant information from an article[2], and it is estimated that about 5% of Pubmed articles are referred to PPI[3]. Therefore, a semi-automatic processing of these papers is a pressing need for biologists and a challenge for bioinformatics researchers.

Automatic PPI information extraction involves many tasks: article classification (as positive/negative according to the PPI subject), biology named entity detection (especially for genes and proteins), normalization, and entity relation identification (especially interacting genes/proteins), which have been been extensively discussed, mainly during the BIOCREATIVE Challenges[4].

In this paper we introduce the general architecture of our system for automatizing the process of PPI information extraction, PPIES, as well as its module for *named entity detection*, and the results it obtains. The *named entity detection module* allows the complete set of entities described by MIMIx to be identified. It is a crucial step for the information extraction system and can also alleviate the curator's task, since all important detected entities can be highlighted, and the curator could go directly to extract the relevant information around them. It is composed of a dictionary look-up and a Conditional Random Field (CRF) classifier.

The dictionary look-up searches in a text for entities which can be associated to a relatively stable set of terms for *organisms, interaction detection and participant identification methods, interaction types, interactor types, biological roles*, and *tissue types*, using soft matching. To assess the performance of this module is a difficult task, as there are no available corpora in the PPI context tagged with all these entities. We have, however, used this module to solve the IMT task of BIOCREATIVE III [10], which consists in the recognition of the

[1] http://www.psidev.info/

[2] Based on answer to query 26 at http://biocreative.sourceforge.net/ppi_questions.html.

[3] Based on motivation for ACT-BC-III at http://www.biocreative.org/tasks/biocreative-iii/ppi/.

[4] http://www.biocreative.org

interaction detection methods used in PPI discovery.

The CRF classifier searches for entities that cannot be described through a dictionary, due to their incompleteness or inaccuracy (new molecules are discovered day after day, new synonyms and acronyms for a specific entity can be introduced and, depending on the data source, the list of names can be more or less complete and the ambiguity more or less difficult to resolve), as in the case of proteins, cell lines, cell types, DNA, and RNA molecules. In this sense the JNLPBA'04 corpus [11] is the only available resource containing biomedical texts tagged by these entity types.

In the following section a literature review related to our named entity detection module is presented. A general overview of the PPIES system as well as of the implementation details of the named entity detection module are given in Section 3. Section 4 describes and discusses the obtained results. Finally, in Section 5 conclusions are drawn and future work directions are discussed.

2. Background

The most important details related to the dictionary look-up systems are highlighted below in Section 2.1. The JNLPBA'04 corpus and the solutions described in the literature for the annotation of its entities are summarized in Section 2.2.

2.1. Dictionary look-up

Dictionary look-up, a type of string matching [12] algorithm, is useful in many Natural Language Processing applications, since it allows to retrieve terms of a given controlled vocabulary (CV) from a raw text. Normally, this vocabulary is formed by tuples of $(Id, term, entity\_type)$. The identifiers, Id, can be used to normalize the recognized *terms*, which are also linked to *entity types*. The accuracy of a dictionary look-up depends on the measure function that is used to compute the matching score level between texts and terms. Examples of soft matching measures are *n-gram similarity*, *Levenshtein distance* [13], and the *Jaro-Winkler* measure [14]. More sophisticated approaches combine different soft matching measures and/or learn the weights of their parameters from the dictionary (e.g. [15], [16], [17], and [18]).

Various techniques that optimize the time searching and the similarity measures have been proposed for dictionary look-up (e.g. [19], [20], [21], [22, 23], [24]). Currently, search engines are used to create indexes of CV and/or of texts and allow retrieve texts associated to terms entered by users. Many bibliographic databases, e.g. PubMed, PubMed Central, Science Citation Index Expanded, ACM, Google Scholar, Citebase and Embase, uses such approach, but only a few of them uses a CV for indexing texts.

PubMed and Embase are the most important examples, in the biomedical area, using CV to index texts. Indexing texts with a CV implies that each text is processed by a dictionary look-up algorithm to capture the mentioned CV terms, and to maintain the recognized terms along with the texts in the

index. Embase [23] indexes texts using their own Emtree thesaurus, formed by approximately 60,000 biomedical terms with a large coverage of chemicals and drug terminology. Part of the database is automatically indexed, but the details of the dictionary look-up algorithm are not provided.

PubMed is indexed using the NLM (National Library of Medicine[5]) Medical Text Indexer (MTI) which in turn uses MetaMap (see [21] for an overview), a dictionary look-up for UMLS Metathesaurus [25]. Other efforts for annotating texts for UMLS and MeSH are MicroMeSH [26], CHARTLINE [27] CLARIT [28], SAPHIRE [29], KnowledgeMap [30], MGREP [31].

MetaMap is the best well-known technology, in the biomedical field for dictionary look-up. It has merged in one tool all experiences for annotating biomedical texts and outperforms almost all other similar systems (an exception is KnowledgeMap in the context of biological process). Text processing in MetaMap is carried out using a series of linguistic steps for obtaining a mapping between segments of a text and concepts in UMLS: 1) tokenization, sentence boundary determination and acronym/abbreviation identification; 2) part-of-speech tagging; 3) lexical lookup of input words in the SPECIALIST lexicon; 4) a shallow parser to identify phrases and their lexical heads; 5) each phrase is analysed for obtaining different variations, and the Metathesaurus terms matching the input text, called candidates, are selected and evaluated; 6) a mapping between text phrases and a combination of the candidates is generated and evaluated. The mapping is filtered, optionally disambiguated, and given as final result. It is out of the scope of this paper to describe the whole complexity behind each of these steps. The interested reader can refer to [21] for a deeper understanding.

Using MetaMap and adjusting it according to a particular use case is difficult. One the one hand, it is open-source but uses SICStus Prolog which is not-open source software. On the other hand, many parameters (e.g. the syntactic analysis algorithms and/or models) cannot be configured at the level granularity that a developer could desire. So, our goal is to construct a highly-configurable CV lookup system with similar linguistic approach as in MetaMap for terms in the context of PPI[6], based only on open-source developments. The complete description of the system is given in Section 3.1.

As previously mentioned, the dictionary lookup module will be used to solve the IMT task of BIOCREATIVE III. IMT task consists in annotating full articles with the experimental methods that were used to detect a protein-protein interaction (PPI), where the PSI-MI ontology is used to obtain the controlled vocabulary that characterizes the experimental methods. The data given by the organizers of the BIOCREATIVE III edition are summarized in Table 1. The task was evaluated considering macro and micro observations, that is, considering only the documents for which a result was returned and considering all documents in the test set, respectively.

Eight teams participated in this task [10]. Six of them used ML approaches

[5] www.nlm.nih.gov

[6] However, we have not yet addressed the word disambiguation problem.

	articles	paragraphs	sentences	words	annotations
Training	2035	178523	2113785	15620104	4348 (in 2003 articles)
Test	305	137047	346974	2600373	528 (in 203 articles)

Table 1: Description of datasets for IMT at BIOCREATIVE III.

to perform the required task. Basically, they focused the task as a multi-label, multi-class classification problem at document or chunk level based on bag-of-words after a lexical analysis (a few teams used n-grams and named entity recognition). The probability output of the classifiers was used to rank and select the final list of experimental methods described in each article. Respect to the macro values, the system described in [32] obtained the best overall performance with 55.06 of F1-measure, in 199 documents, with a precision of 62.46% and a recall of 55.17%. Respect to the micro values, the best overall performance was 55.12 of F1-measure, 52.30% of precision and 58.25% of recall, obtained by the system described in [33]. In [32] a classification model was constructed for each interaction detection method. In [33] in addition to the multi-label, multi-class classifier, the authors converted the problem to a binary classification and in both cases a rich set of features, including contextual text and named entity recognition, was used. Multi-label, multi-class classifier was a 5% superior to the binary classification. In an experiment, after the Challenge, the authors describe an improvement by combining the results of both classifiers and using Logistic regression instead of Support Vector Machine (SVM).

Two systems did not use any ML algorithm. Both used dictionary look-up, but in different ways. The first system [34] used Lucene [35] to maintain documents in the test set and a set of searches was performed (one for each method term). The top 100 documents for each search were recovered, and a method identifier was associated to a document if the score during the search was above certain threshold. The second system [36], used an approach similar to ours: they created a dictionary look-up with the method terms, and returned the largest matching between an analysed text and a method name included in the CV. The first system obtained, as best results, 29.10%, 45.04%, 33.60 for macro precision, recall, and F1-measure, respectively, in a set of 219 documents and 28.17%, 45.92%, 34.92 for the micro-measures. The second system obtained 80.00%, 41.50%, 51.51 for macro precision, recall, and F1-measure, respectively, but returning results for only 30 documents. The results for the micro-observation are 80.65%, 4.74%, 8.96 for precision, recall, and F1-measure, respectively.

2.2. JNLPBA'04 corpus and current solutions

JNLPBA'04 Challenge [11] consisted in the annotation of biomedical texts with a set of five entity types: *protein*, *cell line*, *cell type*, *DNA* and *RNA*. Its corpus training dataset comes from the GENIA corpus, version 3.02, consisting of 2,000 abstracts from a controlled search on MEDLINE using the MeSH terms "human", "blood cells", and "transcription factors". The test data was made

	Abst.	Sent.	Words	protein	DNA	RNA	cell type	cell line
Training	2000	20546	472006	30269	9533	951	6718	3830
Test	404	4260	96780	5067	1056	118	1921	500

Table 2: Training and test set description of JNLPBA'04 challenge.

up of 404 MEDLINE abstracts, most of which were retrieved using the same set of MeSH terms as for training. A general description of the training and test data is given in Table 2.

Eight systems participated in the challenge, obtaining up to 72.55 for the F1-measure. Five of the eight systems used SVM (three of them in combination with HMM and CRF); the other systems used MEMM, HMM, or CRF in isolation. A large set of features was used by the systems, from the lexical (word) level up to syntactic tags and external resources. Table 3 shows the set of features and different approaches for the systems participating in the challenge, and those developed later (separated by a horizontal line).

Lexical predominant features are word, affixes (prefixes and suffixes up to 6 letters), word shape (replacing capital letters by "A", lowercase letters by "a" and digits by "0"), brief word shape (replacing consecutive capital letters by "A", consecutive lowercase letters by "a" and consecutive digits by "0") and orthographic features (binary codes denoting when a word holds a specific feature, i.e., is capitalized, numeric, a punctuation mark, is all in uppercase, is all in lowercase, is a single character, is a special character, includes a hyphen, includes a slash, etc. and a combination of them). Abbreviation detection, word length, and DNA sequence detection have been less frequently used, and the possible advantages of using these features have never been demonstrated.

Boundary error reduction is a problem that has been dealt with in different ways by various systems. Head nouns were used in [37] and [38]; word lists that are highly associated to classes are extracted as lexicons in [52]; keyword lexicons are statistically computed in [39]; keyword and boundary lists in [50].

Part of speech (POS) has demonstrated to be a very useful feature and has been used in the majority of the systems. Other syntactic features as chunk and syntactic tags and the governor of a sentence are used with caution since they could introduce errors obtained by the syntactic analysers into the entity classifier. However, it has been demonstrated that, in general, syntactic features improve the results of biomedical entity recognizers.

Six of the eight systems in the challenge used at least one type of external resources: 1) corpora such as the British National Corpus, the MedLine abstracts and the Penn Treebank for computing frequencies and trigger word extraction; 2) personalized gazetteers extracted from Swissport, LocusLink, Gene Ontology, etc., for keyword identification; 3) specialized taggers to increase the accuracy of certain types of entities (for example, in [52] two gene/protein taggers were used even though the accuracy of protein type extraction was not highly improved by this solution); 4) web searching of entity patterns was exploited by various systems in order to compute lexicons and/or assign weights to words associated

				Lexical features								
	P	R	F-1	W	A	WS	Orth.	Ab.	WL	ACTG	K	B
[37]	69.42	75.99	72.55		x		x	x		x		
[38]	71.62	68.56	70.06	x	x	x		x				
[39]	70.30	69.30	69.80	x	x	x	x					
[40]	67.80	64.80	66.30	x	x		x					
[41]	62.98	69.41	66.04	x								
[42]	67.40	60.10	64.00		x		x		x	x		
[43]	66.50	59.80	63.00		x	x	x			x		
[44]	50.80	47.60	49.10	x	x							
[45]	71.62	68.60	70.10	x	x	x		x				
[46]	68.30	67.50	67.90	x		x	x					
[47]	72.01	73.98	72.98	x	x	x	x					
[48]	70.16	72.27	71.20		x	x	x					
[49]	70.40	75.66	72.94	x	x	x	x		x			
[50]	72.01	76.76	74.31	x	x	x					x	x
[51]	67.90	66.40	67.20	x		x	x					

	Syntactic features			External resources					
	POS	TR	HN	GOV	ST	C	G	BT	W
[37]	x	x	x				x		
[38]	x		x	x	x	x	x		x
[39]		x					x		x
[40]	x				x				
[41]						x			
[42]						x			
[43]	x	x			x	x			
[44]	x							x	
[45]	x		x	x		x	x		x
[46]	x								
[47]		x			x				
[48]	x				x				
[49]	x						x		
[50]	x	x			x				
[51]									

	ML approach			Post-processing								
	SVM	HMM	MMEM	CRF	Abr.	CAS	PH	PRE	NA	POSE	MAO	PI
[37]	x	x			x	x		x	x			x
[38]			x		x		x	x				
[39]				x				x				
[40]	x			x				x				
[41]		x						x				
[42]	x	x						x				
[43]	x											
[44]	x											
[45]			x									
[46]	x											
[47]				x								x
[48]				x								
[49]			x									
[50]				x			x	x		x	x	
[51]	x											

Table 3: Most important results related to the JNLPBA'04 task.
W:word; A:affixes; WS:word shape; Orth: orth. features; Ab: abbreviations; WL: word length; ACTG: DNA sequences; K: keywords; B: boundary word; TR: trigger words; HN: head nouns; GOV: governor; ST: syntactic tags; C: corpus; G: gazetteers; BT: bio-tagger; Abr: abbreviation detection and exclusion of short forms in training data; CAS: cascade resolution for nested entities [37]; PH: parenthesis handling; PRE: previous detected entities; NA: Name alias resolution; POSE: boundary entity detection expansion guided by POS tags; MAO: Merge and/or (as in the JNLPBA guidelines); PI: Pattern induction.

to entities.

From the challenge, it was not clear which (set of) features, external resources, or classification models really contributed to obtaining the best performances. The systems developed in the following years did not use external resources as extensively as in the challenge.

Pre- and post-processing such as abbreviation detection, cascaded entity identification, parenthesis handling, and previously predicted entity tag were also integrated in various systems. The three best systems in the challenge as well as [50] have demonstrated the importance of such kind of processing.

Two of the three systems with the highest F1-measure, [49] and [50], have explored the cascaded classification approach for named entity detection. This consists of dividing the task into two phases: segmentation, in which each word is classified as being part or not being part of an entity; and classification, in which each entity-segment is classified in one of the classes. This solution allows to reduce the training time and also to improve the results. Its drawback is that the improvements obtained might not justify the extra time needed for new classifications.

The main insights that can be drawn from the bio-entity classification systems for the JNLPBA'04 data described in the literature are the following: 1) word shape, suffixes, and prefixes are important features; 2) the deeper and more accurate the text analysis, the more useful it is for entity recognition; 3) CRF seems to be the preferred classifier model; 4) external resources do not improve the performance by more than 2%; 5) some specific pre- and post-processing such as abbreviation detection, expansion by parenthesis pairs, or noun phrase detection are essential for increasing the accuracy of the recognizers.

Protein/gene taggers

Although JNLPBA'04 corpus contains protein/gene entities, we are interested in testing the performance of our JNLPBA'04 classifier for tagging proteins in other protein/gene specific corpora, as obtaining the highest accuracy of protein/gene detection is essential for the PPIES.

Over the last years one of the bio-entities that more attention has received from the Bioinformatics Natural Language Processing community are the proteins, and various corpora contain protein/gene annotations. In addition to the JNLPBA'04 corpus, the GM_II [53], Penn-BioIE corpus [54], and Fsuprge-6 [55] corpora are now publicly available, in IEXML format [56], a uniform format for annotating biomedical corpora. BM_II corpus was released during BioCreAtIve-II for the development of the gene mention (GM) task. It consists on a collection of 4171 sentences in which human genes and proteins are annotated. Penn-BioIE corpus is a selection of 1414 abstracts selected from a corpus describing oncology diseases linked to oncology. Fsuprge corpus, contains 3236 abstracts covering immunogenetics and gene regulation events.

The task of tagging protein/gene has been addressed using two different methodologies: by dictionary look-up strategies for searching protein names described in protein databases, or by constructing a ML-model trained with a protein/gene corpus. A detailed overview of the proposed solutions using each

Test Corpus	System (train corpus)	R	P	F1
GM_II	BANNER(GM_II)	71.2	72.9	72.1
Fsuprge	BANNER(GM II)	51.5	60.6	55.7
Penn-Bio IE	BANNER(GM II)	48.2	56.4	52.0
JNLBPA'04	Abner (JNLBPA'04)	74.7	66.5	70.4

Table 4: Best overall results per protein/gene corpora, considering the available protein/gene taggers. From: Figure 3 in [57].

of the above methods, as well as a comparison of all available taggers against the same corpora set we are using here, can be found in [57]. Some of their findings most relevant for this work are:

- ML approaches outperform dictionary approaches, when tested against the same corpus for which the model was trained.
- BANNER system [58] obtained the best results across all corpora, except for JNLPBA'04, in which the best performance was obtained when the Abner system was trained with the JNLPBA'04 corpus. The summary of their results is reproduced in Tabletab:reproducedResults.
- Using ML techniques for filtering false positives obtained by dictionary approaches does improve their results (improving the precision and not diminishing significantly the recall).

The reference tool in protein/gene taggers is the BANNER system. BANNER is a CRF classifier, trained on a set of lexical and morphological syntactic features that includes word shape, suffixes, and prefixes, lemma, word POS, bigrams and trigrams and combinations of these features. BANNER has achieved the highest performance for the GM task of BIOCREATIVE II, with a recall 71.2%, precision of 72.9% and F1-measure of 72.1. The classifier is publicly available and this gives the advantage of testing new ideas/features on the already consolidated set of features, as well as testing its behaviour with different corpora. With this respect, the interested reader can find details in sections 3.2 and 4.2.3.

3. Named entity detection

The general architecture of PPIES, the PPI information extraction system we are implementing is depicted in Figure 1. At the base of the whole system are the two modules for Natural Language Processing (NLP) and Machine Learning (ML). LingPipe, Standford NLP, python NLTK, Lucene, Weka, libsvm, and Mallet libraries have all been integrated in our framework. Above the base modules are a set of modules placed horizontally on top of each other, and two modules, *text classification* and *domain knowledge integration* located vertically, as they can be used by any other component of the system to improve, asses, or optimize intermediate results.

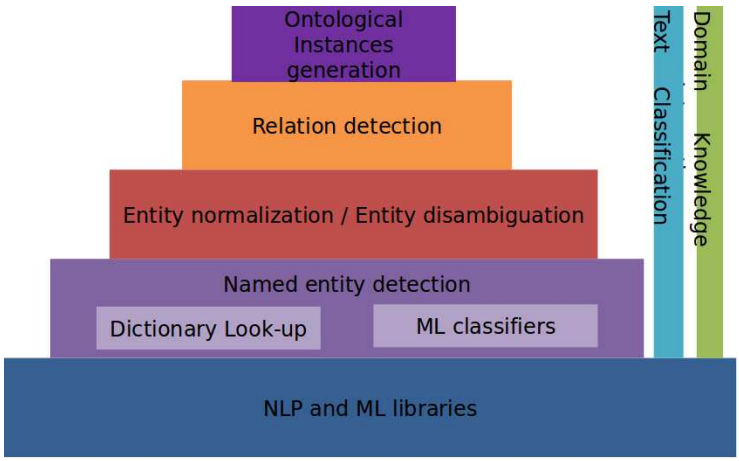


Figure 1: Architecture for PPIES.

Text classification can be performed without considering any named entity detection process to make a coarse classification of biomedical articles. It can also be used at paragraph/sentence level and considering disambiguated entities to classify paragraphs and sentences as a particular description of a PPI detection sub-process.

Domain knowledge, expressed as an OWL knowledge base, is used to: 1) reduce the complexity of some problems based on standardized rules, 2) review collected information to avoid contradictions with well established knowledge, 3) express all extracted knowledge using the same format, which is useful for interchange purposes.

Horizontal modules at higher levels use the information recognized by inferior levels. *Named entity detection* module is in charge of detecting where named entities (such as proteins, cells, organisms, interaction detection method, etc.) are mentioned in texts. The identification of the exact term and its association with a particular identifier in a biomedical resource is the goal of the *entity normalization or disambiguation* module. Relations between such entities can be solved with the *relation detection module*. Finally, at the highest level of the architecture, the *ontological instance generation module* produces an ontological representation, as much complete as possible, of the all mentioned concepts and entities in a text. For doing this, we will follow the technique described in [59], which produced satisfactory results in the archeology domain.

In this paper, we are mainly concerned with the *named entity detection* module, since it is crucial for the information extraction system. Moreover, highlighting of detected entities can simplify curators' job, as they only need to assess the accuracy of the shown detections by reading the text around them, and then complete and link the missing information. A detailed description of each sub-module that composes our named entity recognizer can be found in sections 3.1 and 3.2, respectively. Finally, a description of a greedy and prelim-

Entity type	# of ent. names	Head examples	Source
Organism	548838	virus, *sp.*	NCBI taxonomy
Interact. detection meth.	326	assay, study	psi-MI.obo
Participant ident. meth.	75	assay	psi-MI.obo
Interaction type	97	reaction	psi-MI.obo
Interactor type	62	complex, acid	psi-MI.obo
Biological role	15	donor, aceptor	psi-MI.obo
Cell	1178	cell, neuron, lymphocyte	cell.obo[8]
Tissue	1985	cell, gland, carsinoma	tisslist.txt[9]
Protein	672744	synthetase, protein, precursor	Uniprot database

Table 5: Controlled Vocabulary description.

inary approach to merge the results of both sub-modules is given in Section 3.3.

3.1. Dictionary look-up module for bio-entities associated to protein interactions

Although the MIMIx guidelines are based on general molecular interactions, in this work we limit our study to protein and gene molecules, leaving out the recognition of chemical entity names (which could be obtained from the PubChem or ChEBI databases) and nucleotide sequences (DDBJ, EMBL, or GeneBank). However, we included cells, cell lines, and tissue types since they could be useful for providing a complete experiment description concerning the host system [9].

In Table 5, each entity type that is included in our dictionary look-up is described according to the number of terms it contains, the head noun examples that are commonly used in an entity name, and the source from which the entity names have been obtained. The head nouns of entities are important for identifying the meaning with which the names are expected to be used. For example, the phrase "binding studies" can be associated to *detection methods* instead of recognizing "binding" as an *interaction type*. The majority of the terms of our dictionary look-up come from the CV of PSI-MI: the *psi-MI.obo* ontology[7].

Figure 2 shows a graphic description of the dictionary look-up module, whose functioning is divided in two stages: indexing and searching. During the *indexing stage* the terms of the CV are firstly inspected to discover acronyms and expand the vocabulary with them; secondly, analysed to normalize them and extract head nouns; and finally, indexed using Lucene. During the *searching stage*, when a text is examined, a syntactic analysis is performed and fuzzy queries[10] are constructed from the verbal and noun chunks. The terms in the CV that are closest to the chunks, according to a similarity function, are returned as the list of terms of the CV mentioned in the analysed text.

3.1.1. Indexing stage

Some issues are considered at indexing stage: a) *normalization*, to reduce the variability due to differences in writing styles; b) *acronym discovery*, to

[7] http://psidev.sourceforge.net/mi/psi-mi.obo
[6] http://www.obofoundry.org/cgi-bin/detail.cgi?id=cell
[7] http://expasy.org/txt/tisslist.txt
[10] Lucene fuzzy queries allow non-exact matching phrases to be recovered.

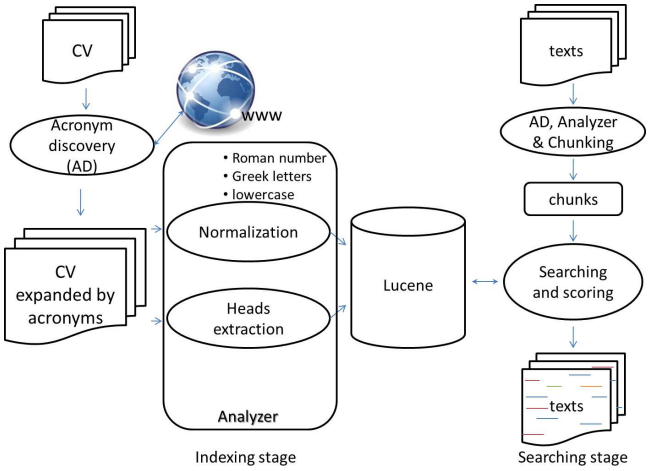


Figure 2: Controlled Vocabulary look-up module.

capture entity names described by acronyms that are not explicit in the CV; c) *head noun extraction*, to reduce the ambiguity of the entity names found; d) CV expansion considering *morphological variations*.

Normalization considers the transformation of Greek letters to a single format, Roman numbers to Latin numbers, uppercase letters to lower case letters in non acronyms. Stopwords are removed, and diacritic marks are also removed from letters, e.g., *naïve* is converted to *naive*.

During indexing phase, acronyms are discovered by aligning the synonyms of a term in the CV (e.g. BRET and bioluminescence resonance energy transfer) and corroborating the acronym (short form of a term) for a long form a term by searching in the web for expressions that associate large and short forms. To this end, long names in the CV are used as input to the Google[11] and Yahoo[12] searching APIs and the retrieved texts are examined to identify analogue expressions as those shown in Figure 3. Finally, all identified acronyms are maintained in the index and are used to expand the CV with all possible combinations.

"<*long form*>,"
"<*long form*>, also called"
"<*long form*> ("
", <*short form*>, "
"("<*short form*>")"

Figure 3: Expressions used for acronym discovery over the web.

A set of head nouns which permit to infer the meaning of an extracted phrase

[11] `https://developers.google.com/custom-search/v1/overview`.
[12] `http://developer.yahoo.com/boss/search/boss_api_guide/index.html`.

is also recovered from the CV. For this purpose, we define the head noun of a term as the last word in the term, if it does not contain any preposition, or as the last word before the preposition, otherwise. This approach is suitable to deal with compound terms in our CV, such as: *cytoplasmic complementation assay* whose head noun is *assay*; and *colocalization by fluorescent probes cloning*, whose head noun is *colocalization*, and is similar to that previously used by [60, 61, 62]. Head nouns appearing in more than five different concepts of an entity type are associated to it, and used during searching as explained below.

Finally, two morphological variations are considered: a) the stems of the words are stored instead of the whole words; b) prefixes with length three or more are used to expand the vocabulary with semantically equivalent variations, e.g.: *acetylglutamic acid* is expanded into variations: *acetyl-glutamic acid* and *acety-L-glutamic acid*.

3.1.2. Searching stage

A first step during the searching stage is acronym discovery. Using again the expressions in Figure 3, acronyms are associated to their long forms, which are then used to replace acronyms in the analysed texts, and to enrich the CV as in the index stage.

The texts are then analysed using the same process as for indexing, and a chunker[13] is used to split the texts into phrases. Each phrase (chunk) is searched for in the Lucene index, and the closest entity names are returned. The Lucene query associated to each phase is computed as described in the algorithm in Figure 4. The query states that numbers and words morphologically similar to acronyms, should occur in the retrieved text with exact matching; the other words should occur in the retrieved text with a minimum similarity of 0.8. This guarantees that little variations, e.g. due to misspelling, do not prevent term recovery.

Lucene uses the Levenshtein distance to solve fuzzy queries. We have modified the default Lucene similarity measure as described in Section 3.1.3, taking into account the importance of each word in the different domains, and using the Levenshtein distance as a parameter: $d(w', w)$.

3.1.3. Similarity function

The similarity function is used to assess the answers of the searcher. It estimates the importance of the matching words in relation to the complete term, considering relevant parameters that are usually disregarded, thus improving the effectiveness of our dictionary look-up:

$$Sim(phrase, t) = \frac{\sum_{w' \in cw'} d(w', w) log(\frac{freq(w)}{1+freq_{NL}(w)})}{\sum_{w \in t \cup cw'} log(\frac{freq(w)}{1+freq_{NL}(w)})} \quad (1)$$

[13] We have experimented with LingPipe and TreeTagger chunkers.

Require: *chunk*: chunk retrieved from an analysed text.
Ensure: *query*: Lucene query associated to the chunk for recovering the closest terms.

{Following the semantics of Lucene API for BooleanQuery, FuzzyQuery and Occur.SHOULD.}
$query = BooleanQuery()$
for word, $w \in chunk$ **do**
  if $matches(w, (([A-Z][.]?)\{,5\}|[0-9]+(.[0-9]+)?))$ **then**
    {the word is a number or an acronym}
    query.add(new BooleanQuery(w), Occur.SHOULD)
  else
    query.add(new FuzzyQuery(w, 0.8), Occur.SHOULD)

return *query*

Figure 4: Algorithm for Lucene query construction.

where cw' is the set of similar words between a chunk, $phrase$, of a text and the term t in the CV, $w' \in phrase$ and $w \in t$. Similar words are recognized by a Levenshtein distance $d(w', w) < 0.2$. $freq(w) = n/N$, with n being the number of terms in the CV containing the word w and N being the total number of terms in the CV. $freq_{NL}(w)$ is an estimation of word frequency in common language, which is based on the Brown Corpus [63]. The cw' set contains also the head of the phrase if it belongs to the set of heads of the entity type of t.

The importance of a word is based on its frequency in the CV and in the common language, and spelling errors are also taken into account, since the distance matching factor has been introduced in the equation and in the fuzzy part of the constructed query.

For the search processing, a *threshold* value, δ, should be specified, and the output of the dictionary look-up system is the set of pairs $(phrase, t)$ that has $Sim(phrase, t) > \delta$.

3.1.4. Using the dictionary look-up for solving the IMT task

The dictionary look-up module was configured to detect the following entities: *interaction detection method*, *participant identification method*, *organism*, *interaction type*, *interactor type* and *biological role*. However, the assessment of the module was obtained only for the entity *interaction detection method*, as no corpus in the context of PPI is available for the rest of the entities. In the following, the treatment of the input datasets and the steps to obtain the results for IMT are described.

IMT task consists in annotating full articles with the experimental methods that were used to detect a protein-protein interaction (PPI), where the PSI-MI ontology is used to obtain the controlled vocabulary that characterizes the experimental methods. The data given by the organizers of the BIOCREATIVE III edition are summarized in Table 1.

We first preprocess the *pdf* articles to obtain the article texts, preserving their original structure (title sections, paragraphs). This is done for two main reasons. On the one hand, some authors have recognized a few clues that link section names and words with specific detection methods (various examples can be found in [10]). On the other, hand our final system should be able to provide a friendly interface for obtaining feedback from users, in which the article structure is used to make visualization and user interaction easier. The *pdf* articles are first processed with pdf2htlm[14] tool and, using the text visual features (font type and size, uppercases, etc.) we recognize paragraphs, notes, and titles, and identify their levels. Even though the conversion might be not perfectly correct, it allows us to obtain better results and contains less mistakes than the converted texts provided by the organizers of the Challenge.

For each converted article in the test dataset, we use our dictionary look-up containing the experimental detection interaction methods of the *psi-mi.obo* ontology and different configurations. The tested configurations were: two different shallow parsers (TreeTagger [64] and LingPipe [65], both using models generated from Genia Corpus), acronym detection module in indexing and searching phases, different threshold values for filtering the matchings.

We also introduced the following post-processing phases in order to generate the final list:

- filtering by parent concepts (FPC): this filters a term from the mention list if it is a parent of other mentioned terms;
- filtering by negation context (FNC): this filters a term from the mention list if it is preceded by a negative pronoun in a window of less than 5 words;
- filtering by reference mention (FRM): this filters a term from the mention list if it is mentioned only in paragraphs containing references;
- adding relevant mentions (ARM): we notice that almost half of the false positive corresponds to 'MI:0019' (coimmunoprecipitation), and half of the false negatives were associated to any of 'MI:0006' (anti bait coimmunoprecipitation) or 'MI:0007' (anti tag coimmunoprecipitation). Therefore, we substitute any 'MI:0019' for both 'MI:0006' and 'MI:0007'.

The obtained results for the IMT task using our dictionary look-up module are detailed in Section 4.1.

3.2. Classifier for proteins, cell lines, cell types, DNA and RNA entities

We use CRF for training, considering only the previous word to classify the current one (order 1). Each sentence is considered as a sequence of words (tokens) and is transformed into the IOB2 format [66]. Therefore, each word is tagged as: $B - ent$ if it is the first word of an entity name of type ent; $I - ent$

[14] http://pdftohtml.sourceforge.net/

if it is not the first word of a entity name of type *ent*; *O* if it is not part of any entity name.

Taking into account the previously well-tested set of features for the JNLPBA'04 corpus, we associated the following features to each token: a) word; b) word shape as explained in the previous section; c) brief word shape; d) prefixes and suffixes of length 3 and 4; e) POS; f) chunk tag.

In addition, we use word numeric normalization, substituting all integer numbers by '0', since it was previously proposed and used satisfactorily in [50]. This allows some common name types to be normalized, e.g. $IL-\backslash d+$ by $IL-0$, thereby increasing the generalization capability of a classifier. Words, POS, and chunks are combined in bigrams and trigrams of $(Feat_{n-1}, Feat_n)$, and $(Feat_{n-1}, Feat_n, Feat_{n+1})$, with $Feat$ being the word, POS, or chunk tag that is associated to an instance word.

Extended set of classifier features. We also defined a set of experiments to find other features to join satisfactorily with the previous ones (the basic configuration). The following features were selected:

- *cell line lexicon*, since cell line is the entity that is more difficult to recognize, we tried to improve its recognition using a lexicon that was extracted from the Cell Line database[15]. The lexicon is formed by the words in the *name*, *description*, and *morphology* fields of the database that appear more than 20 times. The cell line lexicon feature tells if a word appears in the created lexicon;
- *DNA sequence*, a boolean feature which tells if a word represents a DNA sequence;
- *head noun*, a boolean feature which tells if a word is a head of a phrase;
- *distance*, an integer measuring the distance to the head noun;
- *greek*, a boolean feature which tells if a word represents a Greek letter;
- *roman*, a boolean feature which tells if a word represents a Roman number;
- *GWC*, this feature substitutes the word shape feature, and is computed as word shape by replacing any Greek letter by "G";
- *preferred class*, this feature indicates the preferred class (entity type) of a word, if it appears in the training data set and its preferred class exists. The preferred class of a word in the training data is the entity type that is more often associated to the word, and presenting a significant difference of at least 95% with respect to the rest of associated entity types.

[15] http://bioinformatics.istge.it/cldb/cldb.php

- *previous tags*, for each entity tag, a boolean feature is added representing whether a word has been previously tagged in the same abstract as the entity type.

- *frequency tags*, for each entity tag, a double is added representing the frequency with which a word appears associated to the given entity type.

The experiments were designed in order to address the following hypotheses:

- An improvement in the results of the basic configuration can be obtained by complementing it with the extended features, or their combinations. To this end, we first train a CRF classifier with the basic set of features (the basic configuration), and then the extended configurations are constructed, and used to train new CRF classifiers, by adding extended features to the basic set. We want to check if improvements in the overall performance are obtained as a result of these additions.

- The CRF model of the best obtained classifier outperforms or obtains comparable results as an analogue system trained with SVM algorithms. The same set of features that achieve the best results were used to train two SVM algorithms. The first one algorithm, *multi-class SVM*, uses the classical SVM solution for multi-class classification problem[16]; the second algorithm, *multi-class SVM HMM*, uses the results described in [68][17] which considers the pattern structure of the training examples. In our case, it is the information about the sequence of words that constitute the sentences of the input texts. The results of both computations are compared with those obtained using the CRF algorithm.

- The best obtained classifier can be used for tagging proteins with similar results as obtained by the BANNER system. We have used the best obtained classifier for tagging the proteins in corpora GM_II, Penn-BioIE, Fsuprge-6, and compared its results with the best ones available in the literature.

- The set of features included in the best obtained classifier could be combined in the BANNER system to improve its current performance. We have adapted BANNER for allowing the construction of a model with any combination of the extended feature set, and tested its results when using the set of features in the best classifier for the corpora GM_II, Penn-BioIE, Fsuprge-6.

Answers to all the above questions can be found in Section 4.2.

[16] We use the libsvm (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`) library.

[17] We use the SVM-hmm (`http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html`) software.

3.3. Merging dictionary look-up and ML classifier results

Results obtained from the dictionary look-up and the ML classifier are merged and returned to the user as a unique answer of the entity recognition module. One the one hand, the dictionary look-up module returns the terms in the CV with maximum similarity and longest matching for each noun phase of an analysed text. On the other hand, the ML classifier returns the JNLPBA'04 entity type associated to text segments. In cases of ambiguity for a same text segment, we have proceeded in the following way:

- entity types recognized by the ML classifier have priority over the entities recognized by the CV, since the ML approach obtained a better performance than the look-up approach;
- if two or more terms from the CV are assigned to the same text segment, all the terms with three or more occurrences in the whole text are returned to the user, assuming that the tool is being used for the recognition of molecular interaction entity types in full research articles.

4. Results

The dictionary look-up module was configured to detect the following entities: *interaction detection method*, *participant identification method*, *organism*, *interaction type*, *interactor type* and *biological role*. However, the assessment of the module was obtained only for the entity *interaction detection method*, as no corpus in the context of PPI is available for the rest of the entities. In Section 4.1 below, the treatment of the input datasets as well as the obtained results for the IMT are described. The results of the CRF classifier, for the entities *protein*, *cell line*, *cell type*, *DNA* and *RNA*, is given in subsection 4.2. The overall results of the whole named entity detection module are given in subsection 4.3.

4.1. Results on the IMT task

Maintaining the positions where evidence for term annotation are found is indispensable for verification purposes. We called this the "hold term evidence position" principle and our solution is based on it. ML-approach can manage the problem of discovering implicit mentions (textual, usually complex patterns associated to terms) and therefore obtain better results than non-ML solutions. However, they are unable to satisfy the "hold term evidence position" principle and therefore cannot be used to highlight in the text the fragments associated to a particular interaction method. This is a fundamental drawback, as it makes the result validation more difficult. On the other hand, for obtaining competitive results, a very rich set of features is required (e.g. named entity detection, precomputed score per words and n-grams, and MeSH terms identification are all required in [33]).

We investigate here the results of using various different configurations for our dictionary lookup module and few simple post-processing steps to discard or

Configuration	P	R	F1
Dict. look-up, no filter by δ	5.76	100.00	10.89
$+\delta = 0.7$	28.30	30.08	29.26
+Acronyms (searching)	32.30	39.00	35.33
+filterParentTerms (FPT)	35.07	37.76	36.36
+NegContext (FNC) + RefMention (FRM)	40.71	33.19	36.57
+AddRelevMentions (ARM)	36.8	58.11	45.06

Table 6: Results for IMT-task using non-ML approach, micro-observations. Dictionary look-up module uses Lingpipe for shallow parser.

include specific detection methods, as were described in Section 3.1.4. We compare our results with those obtained by non-ML approaches which can return the term evidence position.

Using LingPipe parser the dictionary look-up recovered about 3% more entities that using TreeTagger, in all configurations. The use of acronym discovery at indexing stage decreased the precision and was switched off. The filtering threshold, δ, was empirically obtained by testing different values in the training and development datasets. The threshold for which the best results were obtained, $\delta = 0.7$, was used for the test data, and the initial interaction method mentioning list per article was obtained. Finally the post-processing steps were executed, generating, in this way, the final list.

Table 6 summarizes the results of using our approach for micro-observations, that is the global performance for the whole test set. As it could be expected, without using the threshold value, our dictionary look-up find all mentioned detection methods, but with a very low precision, as in the system described in [67]. The optimum threshold found at 0.7 improved in more than 20.00 points the precision but removed 70% of the previous findings. Discovering acronyms defined in the text improved both precision (by 4%) and recall (by 9%), which highlights the importance of acronym detection. The post-processing steps of FPT, FNC and FRM all decrease recall but seems to improve precision and overall F1 measure. Finally, the post-processing step ARM allows us to achieve a 58.11% of recall and a 45.06 of F1. ARM is a basic heuristic that shows the importance of considering semantic relations amongst recognized terms and detection method popularity in order to select the correct one.

The above described post-processing steps allowed us to obtain a final precision, recall, and F1-measure of 36.8%, 58.11%, and 45.06, respectively. This increased by 10% the best results of the two systems that did not use any ML algorithms, obtaining the same recall as the best solutions among those using complex classifiers. The relatively low precision is a consequence of the large number of significant words in the controlled vocabulary of interaction detection methods that can be used in a different context (i.e. immunoprecipitate, phosphorylate). This could be, in part, because we searched for terms in the dictionary in all of the paragraphs and titles of the full article, without making any analysis of the article parts (some researchers have described the impor-

tance of using only the title, abstract, and methods sections). Recall could be higher with a more complete CV. Synonyms discovery could be useful for CV expansion: we observed that 14 of 46 interaction detection methods in the test dataset are discovered with a precision greater than 85%.

Our approach for the IMT task addresses the problem of acronym discovery, which had never been considered before in this context, and satisfies the"hold term evidence position" principle. It is also very fast (5 seconds per article, in a normal PC), and easily extensible to new terms and domains. Combining the learned-lessons from the ML classifiers described during and after the Challenge and the insights drawn here should allow to further improve our current results in the future.

4.1.1. Failures and successful examples

As can be noticed from the figures of precision, recall and F1-measure, the most important problem is the high number of false positives. The most important source of false positives is introduced by the ARM post-processing step, which replaces any MI:0019 finding with both MI:0006 and MI:0007, and in many cases (96, in case of test dataset) only one of both interaction methods is used (see for example the articles with pubmed identifiers *19008223*, *18337465* and *19864460*).

Following this, MI:0096 (pull-down), MI:00248 (imaging technique) and MI:0051 (fluorescence technology) were the second source of false positives (80, 52 and 31 respectively). A possible explanation of such failures is the use of these terms in contexts different from the interaction detection method description. For example, in the article with PubMed identifier *19088068*, a figure caption describes a *Western blotting analysis* as: *"The presence of H2AZ was detected using anti-H2AZ. A, H2AZ binding of SWR1(1681) and SWR1(ΔN2) complexes. SDS-PAGE (14% gel) and Western blotting analysis of H2AZ pull-down by SWR1(1 681) or SWR1(ΔN2) complexes at the 0.2 or 0.3 M KCl condition..."*, not actually describing a *pull-down assay* for discovering a new interaction.

In the article with PubMed identifier *19933576*, the *green fluorescent protein* is used for the experiments, and the system has recognized "fluorescent" as a highly important word for the CV domain (see the equation 1 in Section 3.1.3), and tagged all mentions of "fluorescence" as a fluorescence technique.

Other false positives, a total of 39, were counted in less that 10 articles.

With respect to false negatives, MI:0416 (fluorescence microscopy) was undetected in 61 articles and MI:0019 (coimmunoprecipitation) in 51. Various articles use *fluorescence microscopy* but either employ the prefix *immuno*, as in article *18480411*; or the matching between a nominal phrase in the text and detection method name was not enough to exceed the prefixed threshold, as in article *20467437*.

In 22 articles, neither MI:0006 nor MI:0007 were found, as *coimmunoprecipitation* does not appear in the text, neither the variations included in the CV. Other mistakes were introduced after using the post-processing step: 18 in the case of MI:0096, 7 in MI:0405 (competition binding) and 5 for MI:0114 (x-ray crystallography). In spite that these strategies improve the overall performance

of the system, more sophisticated methods need to be used in order to reduce the number of removed "good" findings. Other false negatives, a total of 57, were counted in less that 5 articles.

The following three segments of paragraphs from article *20133654* show examples of successful findings and the tagging made by the system:

- *Using [x-ray crystallography – MI:0114], we show the structural basis for titin-M10 interaction with obsl1 in a novel antiparallel Ig-Ig architecture and unravel the molecular basis of titin-M10 linked myopathies.*
- *We investigated the importance of this particular side chain in both OL1 and O1 by [isothermal titration calorimetry (ITC) – MI:0065] and [pull-down – MI:0096] assays.*
- *This single-chain M10-O(L)1 complex was then sandwiched between three concatenated ubiquitin domains (scheme in Fig. 4A), which serve as handles for attachment to the cantilever and the surface of the [AFM – MI:0872].* Notice that in this particular case, the acronym recognition sub-module was used first to match the acronym *AFM* with its long form: *Atomic Force Microscopy*, and then, recognize the term when mentioned using its acronym.

4.2. Results on the JNLPBA'04 challenge dataset

The results we obtained on the JNLPBA'04 dataset with the basic configuration of our system are: 72.52%, 70.10% and 71.29, for recall, precision, and F1-measure respectively. This is the best result amongst the systems which do not use any external resources, neither any post-processing steps, and it is three points less with respect to the best F1-measure result in the literature [50].

As usual for the JNLPBA'04 task evaluation, the results of the classification are expressed in terms of recall, precision and F1-measure, and consider three matching types between a recognized entity and its corresponding entity in the test dataset. A right (left) matching is achieved when the end (start) of a recognized entity coincides with the end (start) of the corresponding entity in the test dataset. A complete matching is achieved when both right and left matching are observed. Details of the results per entity and for complete, right, and left matching are given in Table 7. *Protein* was the entity for which the classifier retrieved the highest percentage of names, and with the best balance between recall and precision. *Cell type* was the best recognized entity. This is not surprising, since *protein* and *cell type* entities are best represented in the training set, providing more examples to detect them correctly.

Seven (cell line lexicon, DNA sequence, distance, roman, GWC, preferred class, frequency tags) out of ten configurations do not improve the results of the basic configuration. Only one configuration (previous tags, with 76.13 of F1-measure) produces a significant improvement of up to five points with respect to the basic configuration, and approximately two points above the best current result in the literature. Tests performed by combining the three features (previous tags, head noun and greek) that improve the basic configuration

	Complete matching			Right matching			Left matching		
	R	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**
protein	*78.07*	69.57	73.58	83.68	74.57	78.86	83.05	74.01	78.27
DNA	65.62	68.41	66.99	72.35	75.42	73.85	67.80	70.68	69.21
RNA	*67.80*	66.67	67.23	77.12	75.83	76.47	69.49	68.33	68.91
cell_type	65.28	78.52	71.29	73.09	87.91	79.82	66.22	79.65	72.31
cell_line	*59.80*	54.86	57.22	69.60	63.85	66.60	63.00	57.80	60.29
[-ALL-]	**72.52**	**70.10**	**71.29**	**79.05**	**76.41**	**77.71**	**76.11**	**73.57**	**74.82**

Table 7: Results of the basic configuration

result did not lead to further improvements. The set of features that produced the best performance is summarized in Tabletab:featuresBestClassifier. So, the final JNLPBA'04 classifier uses a model trained with a CRF and this set of features.

For all measures, entities, and configurations, right matching shows better results than left matching. Right matching is easier because in a large amount of noun phrases the head noun is the last word of the phrase, summarizing its meaning (in our context, its type). Left matching is especially difficult for deciding whether an adjective should be included as part of a name. For example, the adjective "human" appears in at the beginning of 657 *cell types*, 213 *proteins* and 354 *DNA* entities, but was missed in other 96 equivalent cases: 31 for *cell types*, 29 for *proteins* and 25 for *DNA*s. A description of other inconsistences and annotation problems detected in the training set can be found in [47] and [45].

Considering both recall and precision, the worst recognized entity is *cell line*. *Protein* obtains the highest recall while *cell type* obtains the highest precision. However, the F1-measures for these two entity types are comparable (less than 2% of difference for complete and right matching; and 6% for left matching). Entities *DNA* and *RNA* obtain similar values of recall and precision of about 65% for left and complete matching, and as much a 6% more for right matching. The difficulty of classifying a biomedical phrase into one of the goal classes and the trade-off between precision and recall cause the improvement of one measure in one entity type to be related to the decrease of other measures and/or entity types. This makes it difficult to find features that help to improve the overall classification results.

As in other works, i.e. [39], the use of lexicons does not have a positive effect. In our case, the three measures, all entities and configurations were negatively affected by using this feature. Description field in the Cell Line Database details the cell line growth and maintenance. References to cell types, organisms and other biochemical entity types are frequent. Therefore, using all these terms as a lexicon for cell line is inappropriate. In fact, without this lexicon the CRF is able to capture cell line entities based, for example, on highly correlated words appearing near the cell line entities (such as cultured).

Although we have observed certain trends that could suggest the opportunity to use features such as *DNA sequence*, *distance*, *roman*, and *preferred class*, we

Name	**Description**
W	word in lowercase
Lemma	Lemma of the word, according to stemming algorithm by M.F. Porter, available at: `http://snowball.tartarus.org/`
POS	POS tag of the word according to Lingpipe parser trained with the GENIA corpus
Chunk	Chunk tag of the word according to Lingpipe parser trained with the GENIA corpus
WC	*WordClass* of the word
BriefWC	Brief *wordClass* of the word
3Prefix; 4Prefix	3 and 4 prefixes
3Sufix; 4Sufix	3 and 4 prefixes
LemmaComb	unigrams of the lemmas in positions -2, -1, 0, 1 and 2; bigrams of the lemmas in positions [-2, 1], [-1, 0], [0, 1] and [1, 2] and trigrams of the lemmas in positions [-2,-1, 0], [-1, 0, 1] and [0, 1, 2]
POSComb	unigrams of the POS in positions -2, -1, 0, 1 and 2; bigrams of the POS in positions [-2, 1], [-1, 0], [0, 1] and [1, 2] and trigrams of the POS in positions [-2,-1, 0], [-1, 0, 1] and [0, 1, 2]
ChunkComb	unigrams of the Chunk in positions -2, -1, 0, 1 and 2; bigrams of the Chunk in positions [-2, 1], [-1, 0], [0, 1] and [1, 2] and trigrams of the Chunk in positions [-2,-1, 0], [-1, 0, 1] and [0, 1, 2]
Window_*feature*	a combination of the features W, WC, BriefWC, prefixes and suffixes, in a window of [-1, 2] respect to the word.
previous tags_*tag*	boolean representing whether a word has been previously tagged in the same abstract as the entity type.

Table 8: Set of features with the best performance.

	Complete matching			Right matching			Left matching		
	R	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**
protein	82.16	74.10	77.92	88.47	79.80	83.91	86.20	77.75	81.76
DNA	70.27	74.65	72.39	76.33	81.09	78.63	72.92	77.46	75.12
RNA	64.41	69.72	66.96	73.73	79.82	76.65	66.10	71.56	68.72
cell_type	71.94	83.00	77.08	79.59	91.83	85.28	73.30	84.56	78.53
cell_line	65.60	61.89	63.69	73.80	69.62	71.65	69.20	65.28	67.18
[-ALL-]	**77.25**	**75.04**	**76.13**	**83.98**	**81.58**	**82.76**	**80.47**	**78.17**	**79.30**

Table 9: Performance of the best result for JNLPBA'04 task.

have not obtained good results using them.

Greek and *head noun* improve slightly the results obtained with our basic configuration, with 72.56%, 70.16% and 71.34 and 72.27%, 71.34% and 71.80 of recall, precision and F1-measure, respectively. While *greek* feature does not show any particular pattern in the results obtained for entity types and measures, *head noun* feature improves the precision for all matching and entity types. The DNA entity type obtains approximately five points of precision improvement, and only the F1-measure for the RNA class does not increase using the *head noun* feature.

Table 9 shows the results obtained by using *previous tags*, with an overall performance of 77.25%, 75.04% and 76.13 of recall, precision and F1-measure. It is not only our best configuration, but also the current best result amongst the described systems solving the JNLPBA'04 task. The previous best result [50] with 76.76%, 72.01% and 74.31 of precision, recall and F1-measure, is also more complex than ours, since it uses two classification models (one for biomedical entity boundary detection and the other for classifying a biomedical term in a specific class) and a post-processing step that is made up of 4 algorithms.

4.2.1. Failures and successful examples

The system outputs for the abstracts *21184079* and *21066742* are shown in Figure 5. The annotations of the system are highlighted in the text, and the correct annotations are underlined. The first example shows a large coincidence between the correct annotations and those obtained by the system, both in type and matching. The exception is the first appearance of *AP1*, which was tagged as a protein, but corresponds instead to cell type *AP1 site*. Our classifier is less successful in the second example, especially when lists of entities appear. It can be noticed that the identification of the correct left ending of the entities is especially difficult.

In both examples, there are mistakes related with discrepancies in the annotations when parentheses and conjunctions list of entities appear. The IeXML format for entity corpora annotation can solve this type of mistakes, and increased stability could be achieved by systems that consider entities with non-adjacent words.

For a better understanding of the performance of our classifier, we have com-

Human T-cell leukemia virus type 1 tax protein activates transcription through AP-1 site by inducing DNA binding activity in T cells. Human T-cell leukemia virus type 1 (HTLV-1) Tax protein induces the expression of various family members of the transcription factor AP-1, such as c-Jun, JunD, c-Fos, and Fra-1, at the level of RNA expression in T cells. We examined the activity of Tax in transcription through AP-1-binding sites (AP-1 site) in T cells. Transient transfection studies showed that Tax activated the expression of a luciferase gene regulated by two copies of an AP-1 site in the human Jurkat T-cell line. Tax activates the expression of viral and cellular genes through two different enhancers : a cAMP-responsive (CRE) -like element and a kappaB element. Two Tax mutants differentially activated expression of these two elements. Tax703 preferentially activated the kappaB element but not the CRE-like one, whereas TaxM22 showed the reverse. In addition, Tax703 and Tax, but not TaxM22, converted cell growth of a mouse T-cell line from being interleukin (IL) -2-dependent to being IL-2-independent. Unlike the wild-type Tax, Tax703 and TaxM22 only weakly activated the AP-1 site in the T-cell line. Thus, Tax seems to activate the AP-1 site via mechanisms distinct from those of kappaB or CRE-like elements, and the activation of the AP-1 site is dispensable for IL-2-independent growth of CTLL-2 . Electrophoretic mobility shift assays showed that Tax induced strong binding activity to an AP-1 site in CTLL-2, whereas Tax703 did not, indicating that the induction of binding activity to the AP-1 site is essential for the transcriptional activation by Tax. The binding complex induced by Tax in CTLL-2 contained JunD and Fra-2. Other AP-1 proteins were undetectable. Activation of transcription through the AP-1 site in Jurkat cells by JunD and/or Fra-2 was weak. c-Jun, JunB, and c-Fos activation was greater, although the level was still less than that with Tax. Thus, the induction of AP-1 mRNA by Tax may not be sufficient for a complete activation of AP-1 site by Tax. Our results suggest that Tax activates the transcription of cellular genes with AP-1 sites by inducing the DNA-binding activity of AP-1 proteins in T cells, a mechanism distinct from those of CRE-like and kappaB elements. Copyright 2001 Academic Press.

(a) Example 1 (article: 21184079)

The latency pattern of Epstein-Barr virus infection and viral IL-10 expression in cutaneous natural killer/T-cell lymphomas. The nasal type, extranodal natural killer or T (NK/T) -cell lymphoma is usually associated with latent Epstein-Barr virus (EBV) infection. In order to elucidate the EBV gene expression patterns in vivo, we examined eight patients with cutaneous EBV-related NK/T-cell lymphomas, including six patients with a NK-cell phenotype and two patients with a T-cell phenotype. The implication of EBV in the skin lesions was determined by the presence of EBV-DNA, EBV-encoded nuclear RNA (EBER) and a clonality of EBV-DNA fragments containing the terminal repeats. Transcripts of EBV-encoded genes were screened by reverse transcription- polymerase chain reaction (RT-PCR), and confirmed by Southern blot hybridization. The expression of EBV-related antigens was examined by immunostaining using paraffin-embedded tissue sections and cell pellets of EBV-positive cell lines. Our study demonstrated that all samples from the patients contained EBV nuclear antigen (EBNA) -1 mRNA which was transcribed using the Q promoter, whereas both the Q promoter and another upstream promoter (Cp/Wp) were used in EBV-positive cell lines, B95.8, Raji and Jiyoye. Latent membrane protein-1 (LMP-1) mRNA was detected in seven of eight patients and all cell lines, whereas EBNA-2 transcripts were found only in the cell lines. Immunostaining showed no LMP-1, EBNA-2 or ZEBRA antigens in the paraffin-embedded tissue sections, although they were positive in the cell line cells. Latent BHRF1 transcripts encoding bcl-2 homologue and BCRF1 transcripts encoding viral interleukin (vIL) -10 were detected in one and two of eight patients, respectively. A patient with NK-cell lymphoma expressing both transcripts died of rapid progression of the illness. Our results indicate that the restricted expression of the latency-associated EBV genes and the production of vIL-10 and bcl-2 homologue may favour tumour growth, evading the host immune surveillance. Copyright 2001 Cancer Research Campaign.

(b) Example 2 (article: 21066742)

Protein DNA RNA cell type cell line

(c) Color legend

Figure 5: JNLPBA'04 classifier output examples

		Real classification				
		protein	**DNA**	**RNA**	**cell_type**	**cell_line**
Results	**protein**	**562(68.4)**	162(19.61)	27(3.27)	51(6.17)	24 (2.91)
	DNA	64(37.43)	**100(58.48)**	4(2.34)	2(1.17)	1(0.58)
	RNA	7(35.00)	0	**13(65.00)**	0	0
	cell_type	13(5.39)	0	0	**181(75.10)**	47(19.50)
	cell_line	11(5.67)	0	0	117(60.31)	**66(34.02)**

Table 10: Disagreement matrix. Between parenthesis the percentage in relation with the total of false positive cases for the corresponding row.

puted a *disagreement matrix* (Table 10) which shows how many false positives of an entity type correspond to true cases of each entity type.

The first consideration that can be drawn by observing Table 10 is that a great part of the disagreement is due to incomplete matchings (one or more words of the entity are not detected), especially for *cell type*, *protein*, and *RNA* entity types. Incomplete matchings are shown on the diagonal of the matrix, where the entity type for false positives and real classification coincides, and it can be observed that all diagonal values are the highest of their columns. This issue elicits a research question: are the current matching criteria reflecting the relevance of the named entity set found? During the curation task, for example, it is important to mark as many entities as possible, even if their complete names are not well detected. In fact, by using incomplete matching, the overall results of our classifier improve up to 84.96%, 85.77% and 85.36 for recall, precision and F1-measure. As future work, we plan to study more flexible matching criteria and their inclusion in quality measures.

A second issue is related to the errors made by confusing pairs of different entity types: all false positives of *RNA* type are classified as *proteins*; the 60.31% of *cell lines* as *cell types* (19.50% of the *cell types* as *cell lines*); and 37.43% of *DNA* as *proteins* (19.61% of the *proteins* as *DNA*). The reduced capabilities of the current ML algorithms for dealing with imbalanced datasets is, in part, responsible for these errors. However, we also think that the background knowledge of curators has not been reflected in the features tested until now. Another research question arises: How should background knowledge be integrated into ML approaches? This is another future direction of our research.

4.2.2. Results using SVM algorithms

As described in Section 3.2, we use the set of features in Table 8 for training the multi-class SVM and multiclass SVM-hmm algorithms, and the classification processes were updated in order to consider the previously tagged entities in an abstract. The resulting models do not improve the performance obtained by using the CRF algorithm. Tables 11 and 12 show the performance of these classifiers in terms of recall, precision and F1-measure for left, right and complete matchings.

On the one hand, as expected, multi-class SVM is outperformed by both multi-class SVM-hmm and CRF. This justifies the importance of using the word

	Complete matching			Right matching			Left matching		
	R	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**
protein	75.65	68.83	72.08	88.79	80.79	84.60	83.42	75.90	79.48
DNA	58.71	60.19	59.44	77.27	79.22	78.24	67.33	69.03	68.17
RNA	51.69	65.59	57.82	77.97	98.92	87.20	55.93	70.97	62.56
cell_type	60.59	72.66	66.08	79.85	95.76	87.08	69.65	83.52	75.96
cell_line	52.00	52.53	52.26	74.20	74.95	74.57	61.40	62.02	61.71
-ALL-	**68.55**	**67.56**	**68.05**	**84.41**	**83.19**	**83.80**	**76.76**	**75.65**	**76.20**

Table 11: Results of the JNLPBA'04 classifier using the features in Table 8 and multi-class SVM algorithm for training.

	Complete matching			Right matching			Left matching		
	R	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**
protein	73.95	70.35	72.11	82.83	78.80	80.77	79.99	76.10	77.99
DNA	61.17	75.91	67.75	72.82	90.36	80.65	63.35	78.61	70.16
RNA	50.85	64.52	56.87	70.34	89.25	78.67	51.69	65.59	57.82
cell_type	65.90	81.84	73.01	79.07	98.19	87.60	69.08	85.78	76.53
cell_line	53.20	68.91	60.05	67.40	87.31	76.07	58.40	75.65	65.91
-ALL-	**69.09**	**72.96**	**70.98**	**79.72**	**84.18**	**81.89**	**73.91**	**78.04**	**75.92**

Table 12: Results of the JNLPBA'04 classifier using the features in Table 8 and multi-class SVM-hmm algorithm for training.

sequence structure in NER problems, as has been previously described (see [69] for example). Both SVM-hmm and CRF algorithms consider the dependences between the states in the HMM machine and between the states and the features of the training examples. This brings advantage over the independent words vision in SVM algorithm. SVM, however, slightly outperforms SVM-hmm in both right and left matching. Once again, the inconsistencies in the training samples in the endings of the recognized entities could justify that an independent-structure provides some advantages when non-complete matching is observed.

On the other hand, CRF outperforms the solution provided by SVM-hmm in all matching types. Given that both algorithms have been executed with identical set of features, and use the finite machine state model of HMM, the difference on their performance could be expected to be smaller. However, a subtitle implementation issue, associated with the use of different feature functions to compare the HMM states, causes this problem : while the CRF algorithm implemented in Mallet uses a second order forward functions, the SVM-hmm implementation uses first order and token independent first and second order functions, as was noticed and proved in [70].

4.2.3. Protein tagging with the CRF classifier

Considering our best solution for the JNLPBA'04 task (the CFR classifier trained with the features in Table 8), we proceed to verify if its results are

Corpus	Recall	Precision	F1
GM_II	72.66	56.64	61.72
JNLPBA'04_proteins	81.67	75.00	78.19
Penn-BioIE	43.05	49.13	45.89
Fsuprge	54.43	58.54	56.41

Table 13: Results of our JNLPBA'04 classifier for the test dataset in protein corpus.

comparable to those obtained by classifiers trained with specialized protein corpora. We fed our classifier with the test dataset of corpora: GM_II, Penn-BioIE, Fsuprge and JNLPBA'04 tagged only for protein entities. In Table 13 the results obtained by our classifier for complete matching are shown.

It can be observed that our classifier obtains significant better results than the BANNER solution for the JNLPBA'04 corpus. This support the hypothesis of a highly dependence between the tagging guidelines of the training sets and the results obtained when testing the models with other corpus (same behaviour as observed in [57]). Notice that the training set used in our classifier differentiates between proteins, DNA and RNA molecules, a differently from the approach used in BC_II, Penn-BioIE, Fsuprge corpora in which DNA and RNA molecules, are also considered as sometimes proteins, depending on the context. We obtain also a slight improvement compared with the BANNER results for the Fsuprge.

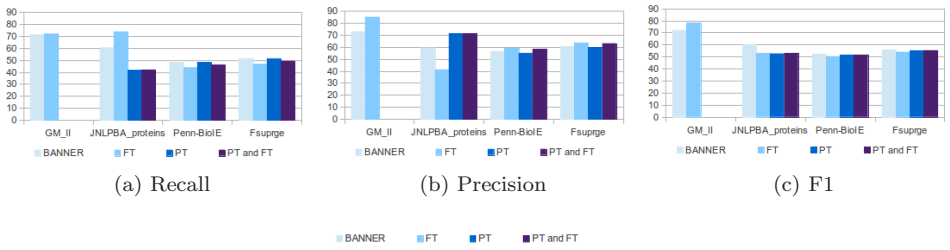
As the most beneficial feature for our system was *previous tags*, we wanted to verify if the BANNER system could also benefit from its inclusion. We have also tested *frequency tags* feature, as GM_II corpus is formed by sentences instead of abstracts, and therefore *previous tags* feature is not applicable. *frequency tags* has a similar aim as *previous tags*, but is applicable independently from the length of the texts in the training dataset.

The details of the performance obtained by our version of BANNER, with the four combination of using these two features are shown in Table 14, and represented in Figure 6. As it can be noticed, none of the models seem to improve their F1-measure as a consequence of using using *previous tags* and *frequency tags* features, while precision increase and recall decrease by similar quantities. For the JNLPBA'04 corpus, however, the opposite behaviour is observed (precision of 41.31%, and recall of 73.77%), which is compatible with our results for the CRF classifier respect to protein entities, and with the results in [57] for the BANNER system tested with the JNLPBA'04 corpus.

	BANNER			**using FT**		
	R	**P**	**F1**	**R**	**P**	**F1**
GM_II	71.20	72.90	72.04	71.99	84.85	77.89
JNLPBA_proteins	60.30	59.50	59.90	73.77	41.31	52.96
Penn-BioIE	48.20	56.40	51.98	43.86	58.96	50.30
Fsuprge	51.50	60.60	55.68	46.88	63.55	53.96

	using PT			using FT and PT		
	R	**P**	**F1**	**R**	**P**	**F1**
JNLPBA_proteins	41.65	71.34	52.6	41.99	71.38	52.88
Penn-BioIE	48.39	54.96	51.47	46.23	58.39	51.60
Fsuprge	51.36	59.61	55.18	49.32	62.97	55.32

Table 14: Results of our version of BANNER using any combination of *frequency tags* and *previous tags* features. FT: frequency tags; PT: previous tags.



(a) Recall (b) Precision (c) F1

(d) Legend

Figure 6: Results of our version of BANNER using any combination of *frequency tags* and *previous tags* features. FT: frequency tags; PT: previous tags.

4.3. Merging dictionary look-up and ML classifier results

The results obtained from the dictionary look-up and the ML classifier are merged (giving priority to those obtained from the JNLPBA'04 classifier and repeated at least three times in the text) and returned to the user as a unique answer. In our experiments, only a 5% of the noun phrases presented the ambiguity problem, and just the 2% of them needed the second strategy.

Given the way of merging the results, the performance of the system for entity types *proteins, DNA, RNA, cell type and cell line* (from the JNLPBA'04 classifier) remained unchanged, as described in Table 9. The performance for entity type *interaction method detection* after merging showed a 1.7 points decrease of recall, obtaining 35.10%; a slight improvement (0.31 points) in precision, achieving 58.32%; and an F1 measure of 43.82.

A demo of the system is available at: `http://www.doc.ic.ac.uk/~rdanger/cgi-bin/biochemicalER/biochem_demo/pcpal.cgi`.

5. Conclusions

In this paper, we have described the architecture of PPIES, our PPI information extraction system, and detailed the named entity detection module, formed by a dictionary look-up for standardized vocabulary and a ML classifier, which allows the complete set of entities described by MIMIx to be identified.

Various techniques for normalization and for acronym detection were explored in the dictionary look-up system. The best results we obtained improves

by about 10% the current solutions for the IMT task that do not use ML, highlighting the advantage of using these two strategies. Automatic synonym and term discovery will be addressed in the future to mitigate the effects of vocabulary dynamism.

We developed a CRF classifier for *protein, cell line, cell type, DNA* and *RNA* entities, based on the JNLPBA'04 data set, which contains a useful set of well-tested features: *word, word shape, POS* and *chunk*, and we tested a set of other features which have revealed to be not useful for this task. Our best solution was obtained by adding a contextual feature at abstract level, improving by approximately 5 points our basic configuration performance. The obtained final results of 77.25%, 75.04% and 76.13 of recall, precision, and F1-measure, respectively, outperform the results of all current available solutions in the literature for the JNLPBA'04.

Two interesting conclusions have come from these experiments: 1) the need to define new quality measures considering more flexible matching criteria; and 2) the difficulty of obtaining better results without integrating background domain knowledge into text processing. Combining natural language processing with knowledge domain modeling, as in the PPIES architecture, could be a way to obtain better results. In the case of dictionary look-up, a second phase could use conceptual density, for example, to select a term that is described but not mentioned. In the case of named entity recognition using machine learning algorithms, constraining the statistical computation with some background knowledge could help to guide algorithms in selecting the appropriate feature modeling. We plan to study these issues in future developments.

The most remarkable achievement of this work is the availability of a system that harmoniously integrates a dictionary look-up and a CRF classifier modules, highly configurable, for the most important PPI entity types, which obtains better or comparable performances that the current available state of the art. Our tool can be applied and was tested on different corpora and configurations. We plan to employ the insights drawn from this work to perform new experiments, in which the outputs of each module will be contextually taken into account, for the mutual improvement and better integration of their results.

6. Acknowledgement

This work has been funded by MICINN, Spain, as part of the "Juan de la Cierva" Program and the project Text-Enterprise 2.0 project (TIN2009-13391-C04-03), as well as the by the European Commission as part of the WIQ-EI IRSES project (grant no. 269180) within the FP 7 Marie Curie People Framework.

References

[1] E. Phizicky, S. Fields, Protein-protein interactions: Methods for detection and analysis, Microbiological reviews 59 (1) (1995) 94–123.

[2] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stempflen, H. Mewes, A. Ruepp, D. Frishman, The mips mammalian protein-protein interaction database., Bioinformatics 21 (2005) 832–834.

[3] G. Bader, D. Betel, C. Hogue, Bind: the biomolecular interaction, Network Database. Nucleic Acids Res 31 (2003) 248–250.

[4] L. Salwinski, C. Miller, A. Smith, F. Pettit, J. Bowie, D. Eisenberg, The database of interacting proteins: 2004 update, Nucleic Acids Res. 32 (2004) D449–D451.

[5] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, G. Cesareni, Mint: the molecular interaction database, Nucleic Acids Res 35 (2007) D572–D574.

[6] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, R. Apweiler, H. Hermjakob, Intact - open source resource for molecular interaction data, Nucleic Acids Res 35 (2007) D561–D565.

[7] T. Reguly, A. Breitkreutz, L. Boucher, B.-J. Breitkreutz, G. Hon, C. Myers, A. Parsons, H. Friesen, R. Oughtred, A. Tong, C. Stark, Y. Ho, D. Botstein, B. Andrews, C. Boone, O. Troyanskya, T. Ideker, K. Dolinski, N. Batada, M. Tyers, Comprehensive curation and analysis of global interaction networks in saccharomyces cerevisiae, J. Biol 5 (11).

[8] G. R. Mishra, M. Suresh, K. Kumaran, N. Kannabiran, S. Suresh, P. Bala, K. Shivakumar, N. Anuradha, R. Reddy, T. M. Raghavan, S. Menon, G. Hanumanthu, M. Gupta, S. Upendran, S. Gupta, M. Mahesh, B. Jacob, P. Mathew, P. Chatterjee, K. S. Arun, S. Sharma, K. N. Ch, A. Deshp, K. Palvankar, R. Raghavnath, R. Krishnakanth, H. Karathia, B. Rekha, R. Nayak, K. S. Deshp, M. Sarker, T. S. K. Prasad, A. P., Human protein reference database: 2006 update, Nucleic Acids Res 34 (2006) D411–D414.

[9] S. Orchard, L. Salwinski, S. Kerrien, L. Montecchi-Palazzi, M. Oesterheld, V. Stempflen, A. Ceol, A. Chatr-aryamontri, J. Armstrong, P. Woollard, J. J. Salama, S. Moore, J. Wojcik, G. D. Bader, M. Vidal, M. E. Cusick, M. Gerstein, A.-C. Gavin, G. Superti-Furga, J. Greenblatt, J. Bader, P. Uetz, M. Tyers, P. Legrain, S. Fields, N. Mulder, M. Gilson, M. Niepmann, L. Burgoon, J. D. L. Rivas, C. Prieto, V. M. Perreau, C. Hogue, H.-W. Mewes, R. Apweiler, I. Xenarios, D. Eisenberg, G. C. . H. Hermjakob, The minimum information required for reporting a molecular interaction experiment (mimix), Nature Biotechnology 25 (2007) 894 – 898.

[10] M. Krallinger, M. Vazquez, F. Leitner, D. Salgado, A. Chatr-aryamontri, A. Winter, L. Perfetto, L. Briganti, L. Licata, M. Iannuccelli, L. Castagnoli,

G. Cesareni, M. Tyers, G. Schneider, F. Rinaldi, R. Leaman, G. Gonzalez, S. Matos, S. Kim, W. Wilbur, L. Rocha, H. Shatkay, A. Tendulkar, S. Agarwal, F. Liu, X. Wang, R. Rak, K. Noto, C. Elkan, Z. Lu, The protein-protein interaction tasks of biocreative iii: classification/ranking of articles and linking bio-ontology concepts to full text, BMC Bioinformatics 12.

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

[12] C. E. L. Thomas H. Cormen, R. L. Rivest, , C. Stein, Introduction to Algorithms, MIT Press and McGraw-Hill, 2001, Ch. Chapter 32: String Matching, pp. 906–932.

[13] V. Levenshtein, Binary codes capable of correcting spurious insertions and deletions of ones, Prob. Inf. Transm 1 (1965) 8?17.

[14] W. Winkler, The state of record linkage and current research problems, Tech. rep., Statistical Research Division, U.S. Bureau of the Census (1999).

[15] E. Ristad, P. Yianilos, Learning string-edit distance, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 522..532.

[16] W. Cohen, J. Richman, Learning to match and cluster large high-dimensional data sets for data integration, in: Proceedings of KDD, 2002, pp. 475–480.

[17] M. Bilenko, R. Mooney, Adaptive duplicate detection using learnable string similarity measures, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD- 2003), 2003, pp. 39–48.

[18] Y. Tsuruoka, J. McNaught, J. Tsujii, S. Ananiadou, Learning string similarity measures for gene/protein name dictionary look-up using logistic regression, BIOINFORMATICS 23 (20) (2007) 2768–2774. doi:doi:10.1093/bioinformatics/btm393.

[19] M. J. C. Alfred V. Aho, Efficient string matching: An aid to bibliographic search, Commun. ACM 18 (6) (1975) 333–340.

[20] R. A. Baeza-yates, C. H. Perleberg, Fast and practical approximate string matching, in: In Combinatorial Pattern Matching, Third Annual Symposium, Springer-Verlag, 1992, pp. 185–192.

[21] A. R. Aronson, F.-M. Lang, An overview of metamap: historical perspective and recent advances, J Am Med Inform Assoc 17 (2010) 229–236.

[22] Embase guide to emtree and indexing systems, Support Publications from Excerpta Medica/EMBASE 2.

[23] Embase indexing - guide 2012: A comprehensive guide to embase indexing policy (2012).
URL http://www.embase.com/info/UserFiles/Files/Embase\%20indexing\%20guide\%202012.pdf

[24] J. Giles, Science in the web age: Start your engines, Nature 438 (7068) (2005) 554–555.

[25] D. Lindberg, B. Humphreys, A. McCray, The unified medical language system, Methods Inf Med 32 (4) (1993) 281–91.

[26] B. G. Lowe HJ, Micromesh: a microcomputer system for searching and exploring the national library medicine's medical subject headings (mesh) vocabulary, in: Proc Annu Symp Comput Appl Med Care, 1987, pp. 717–720.

[27] R. A. Miller, F. M. Gieszczykiewicz, J. K. Vries, G. F. Cooper, CHARTLINE: providing bibliographic references relevant to patient charts using the UMLS Metathesaurus Knowledge Sources, Proceedings of the Annual Symposium on Computer Application in Medical Care. (1992) 86–90.
URL http://view.ncbi.nlm.nih.gov/pubmed/1483014

[28] D. A. Evans, K. Ginther-Webster, M. Hart, R. Lefferts, I. Monarch, Automatic indexing using selective nlp and first-order thesauri, in: A. Lichnerowicz (Ed.), Intelligent Text and Image Handling. Proceedings of a Conference, RIAO '91. Amsterdam, NL, 1991, pp. 624–644.

[29] H. W. R., G. R.A., Saphire: an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships, Comput Biomed Res 23 (1990) 410–425.

[30] J. C. Denny, J. D. Smithers, R. A. Miller, A. S. III, Research paper: "understanding" medical school curriculum content using knowledgemap, JAMIA 10 (4) (2003) 351–362.

[31] P. Nadkarni, R. Chen, C. Brandt, Umls concept indexing for production databases: a feasibility study., Am Med Inform Assoc 8 (2001) 80–91.

[32] R. Leaman, R. Sullivan, G. Gonzalez, A top-down approach for finding interaction detection methods, in: Proceedings of BioCreative III, 2010, pp. 92–96.

[33] X. Wang, R. Rak, A. Restificar, C. R. Chikashi Nobata, R. T. B. Batista-Navarro, R. Nawaz, S. Ananiadou, Detecting experimental techniques and selecting relevant documents for protein-protein interactions from biomedical literature, BMC Bioinformatics 12 (S11).

[34] D. Salgado, M. Krallinger, E. Drula, A. Tendulkar, A. Valencia, C. Marcelle, Myminer system description, in: Proceedings of BioCreative III, 2010, pp. 148–151.

[35] M. McCandless, E. Hatcher, O. Gospodneti, Lucene in Action, Second Edition, Manning publications co., 2010.

[36] S. Matos, D. Campos, J. Oliveira, Vector-space models and terminologies in gene normalization and document classification, in: Proceedings of BioCreative III, 2010, pp. 110–115.

[37] Z. GuoDong, S. Jian, Exploring deep knowledge resources in biomedical name recognition, in: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Association for Computational Linguistics, 2004, pp. 96–99.

[38] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. Manning, G. Sinclair, Exploiting context for biomedical entity recognition: From syntax to the web, in: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Association for Computational Linguistics, 2004, pp. 88–91.

[39] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, in: In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA, 2004, pp. 104–107. doi:http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.112.7693.

[40] Y. Song, E. Kim, G. G. Lee, B.-K. Yi, Posbiotm-ner: a trainable biomedical named-entity recognition system., Bioinformatics 21 (11) (2005) 2794–2796. doi:10.1093/bioinformatics/bti414.
URL http://dx.doi.org/10.1093/bioinformatics/bti414

[41] S. Zhao, Named entity recognition in biomedical texts using an hmm model, in: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, JNLPBA' 04, Association for Computational Linguistics, Stroudsburg, PA, USA, 2004, pp. 84–87.
URL http://portal.acm.org/citation.cfm?id=1567594.1567613

[42] M. Rössler, Adapting an ner-system for german to the biomedical domain, in: JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Association for Computational Linguistics, Morristown, NJ, USA, 2004, pp. 92–95.

[43] K. M. Park, S. H. Kim, D. G. Lee, H. C. Rim, Boosting lexical knowledge for biomedical named entity recognition, in: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004), Geneva, Switzerland., 2004.

[44] C. Lee, W.-J. Hou, H.-H. Chen, Annotating multiple types of biomedical entities: A single word classificication approach, in: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)., 2004.

[45] S. Dingare, M. Nissim, J. Finkel, C. Manning, C. Grover, A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations., Comp Funct Genomics 6 (1-2) (2005) 77–85. doi:10.1002/cfg.457.
URL http://dx.doi.org/10.1002/cfg.457

[46] C. Giuliano, A. Lavelli, L. Romano, Simple Information Extraction (SIE) (2005).
URL http://tcc.itc.it/research/textec/tools-resources/sie/giulianosie.pdf

[47] R. T.-H. Tsai, C.-L. Sung, H.-J. Dai, H.-C. Hung, T.-Y. Sung, W.-L. Hsu, Nerbio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition., BMC Bioinformatics 7 Suppl 5 (2006) S11. doi:10.1186/1471-2105-7-S5-S11.
URL http://dx.doi.org/10.1186/1471-2105-7-S5-S11

[48] C. Sun, Y. Guan, X. Wang, L. Lin, Rich features based conditional random fields for biological named entities recognition., Comput Biol Med 37 (9) (2007) 1327–1333. doi:10.1016/j.compbiomed.2006.12.002.
URL http://dx.doi.org/10.1016/j.compbiomed.2006.12.002

[49] S.-K. Chan, W. Lam, X. Yu, A cascaded approach to biomedical named entity recognition using a unified model, in: Proc. Seventh IEEE Int. Conf. Data Mining ICDM 2007, 2007, pp. 93–102. doi:10.1109/ICDM.2007.20.

[50] L. Li, R. Zhou, D. Huang, Two-phase biomedical named entity recognition using CRFs, Comput Biol Chem 33 (4) (2009) 334–338. doi:10.1016/j.compbiolchem.2009.07.004.
URL http://dx.doi.org/10.1016/j.compbiolchem.2009.07.004

[51] M. S. Habib, J. Kalita, Scalable biomedical named entity recognition: investigation of a database-supported svm approach., Int. J. Bioinform. Res. Appl. 6 (2) (2010) 191–208.

[52] K.-J. Lee, Y.-S. Hwang, S. Kim, H.-C. Rim, Biomedical named entity recognition using two-phase model based on SVMs, J. Biomed. Inform. 37 (6) (2004) 436–447. doi:10.1016/j.jbi.2004.08.012.
URL http://dx.doi.org/10.1016/j.jbi.2004.08.012

[53] L. Smith, L. Tanabe, R. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. Struble, R. Povinelli, A. Vlachos, W. Baumgartner, L. Hunter, B. Carpenter, R. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans,

C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Mana-Lopez, J. Mata, W. J. Wilbur, Overview of biocreative ii gene mention recognition, Genome Biology 9 (Suppl 2) (2008) S2. doi:10.1186/gb-2008-9-s2-s2.
URL http://genomebiology.com/2008/9/S2/S2

[54] K. Seth, A. Bies, M. Liberman, M. Mandel, R. Mcdonald, M. Palmer, A. Schein, Integrated annotation for biomedical information extraction, in: Proceedings of the BioLINK 2004, 2004.
URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.7405

[55] U. Hahn, E. Beisswanger, E. Buyko, M. Poprat, K. Tomanek, J. Wermter, Semantic annotations for biology: a corpus development initiative at the jena university language & information engineering (julie) lab., in: LREC, European Language Resources Association, 2008.
URL http://dblp.uni-trier.de/db/conf/lrec/lrec2008.html#HahnBBPTW08

[56] D. Rebholz-Schuhmann, H. Kirsch, G. Nenadic, Iexml: towards a framework for interoperability of text processing modules to improve annotation of semantic types in biomedical text., in: BioLINK, ISMB 2006, Fortaleza, Brazil., 2006.

[57] D. Rebholz-Schuhmann, S. Kafkas, J.-H. Kim, C. Li, A. Jimeno Yepes, R. Hoehndorf, R. Backofen, I. Lewin, Evaluating gold standard corpora against gene/protein tagging solutions and lexical resources, Journal of Biomedical Semantics 4 (1) (2013) 28. doi:10.1186/2041-1480-4-28.
URL http://www.jbiomedsem.com/content/4/1/28

[58] R. Leaman, G. Gonzalez, Banner: an executable survey of advances in biomedical named entity recognition., Pac Symp Biocomput (2008) 652–663.

[59] R. Danger, R. Berlanga, Generating complex ontology instances from documents, Algorithms (2009) 16–30.

[60] W. B. C. Donald Metzler, Analysis of statistical question classification for fact-based questions, Journal of Information Retrieval.

[61] F. Li, X. Zhang, J. Yuan, X. Zhu, Classifying what-type questions by head noun tagging, in: COLING, 2008, pp. 481–488.

[62] M.-C. de Marneffe, C. D. Manning., Stanford typed dependencies manual. (2008).
URL http://nlp.stanford.edu/software/dependencies_manual.pdf.

[63] W. N. Francis, H. Kucera, A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Brown), Tech. rep., Brown University (1964, 1971, 1979).

[64] H. Schmid, Probabilistic part-of-speech tagging using decision trees, in: Proceedings of International Conference on New Methods in Language Processing, 1994.

[65] Alias-i. 2008. lingpipe 4.1.0, http://alias-i.com/lingpipe (accessed April, 2013).

[66] T. K. Sang, J. Veenstra, Representing text chunks, in: EACL, 1999, pp. 173–179.

[67] G. Schneider, S. Clematide, F. Rinaldi, Detection of interaction articles and experimental methods in biomedical literature, BMC Bioinformatics 12 (S13).

[68] Y. Altun, I. Tsochantaridis, T. Hofmann. Hidden Markov Support Vector Machines. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), 2003.

[69] Li D., G. Savova, K. Kipper-Schuler. Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts. Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, Association for Computational Linguistics, 2008, 94-95.

[70] S. Keerthi, S. Sundararajan. CRF versus SVM-Struct for Sequence Labeling, Technical Report. Yahoo Research, 2007.