

Automating the gathering of relevant information from biomedical text

Catherine Canevet



Doctor of Philosophy

Institute for Communicating and Collaborative Systems

School of Informatics

University of Edinburgh

2009

Abstract

More and more, database curators rely on literature-mining techniques to help them gather and make use of the knowledge encoded in text documents. This thesis investigates how an assisted annotation process can help and explores the hypothesis that it is only with respect to full-text publications that a system can tell relevant and irrelevant facts apart by studying their frequency.

A semi-automatic annotation process was developed for a particular database - the Nuclear Protein Database (NPD), based on a set of full-text articles newly annotated with regards to subnuclear protein localisation, along with eight lexicons. The annotation process is carried out online, retrieving relevant documents (abstracts and full-text papers) and highlighting sentences of interest in them. The process also offers a summary Table of the facts found clustered by type of information.

Each method involved in each step of the tool is evaluated using cross-validation results on the training data as well as test set results. The performance of the final tool, called the “NPD Curator System Interface”, is estimated empirically in an experiment where the NPD curator updates the database with pieces of information found relevant in 31 publications using the interface. A final experiment complements our main methodology by showing its extensibility to retrieving information on protein function rather than localisation.

I argue that the general methods, the results they produced and the discussions they engendered are useful for any subsequent attempt to generate semi-automatic database annotation processes. The annotated corpora, gazetteers, methods and tool are fully available on request of the author (catherine.canevet@bbsrc.ac.uk).

Acknowledgements

I would like to express my gratitude to my supervisors - Professor Bonnie Webber and Professor Wendy Bickmore - for their time, help and support. I appreciate Prof. Webber's vast knowledge and skill in many areas, and her assistance in writing reports. I would like to thank Prof. Bickmore in particular for taking time to annotate corpora and test the system developed for this thesis. Above all, I would like to thank them both for always believing in me as it really helped my confidence.

I would also like to thank Tamara Polajnar for working with me to produce the baseline results given in Section 3.4.3 (Table 3.8), as well as Dr Anna Divoli for altering the 200 KB limit on BioIE for a weekend so that I could compute BioIE's results on my training corpus (Section 3.5, Table 3.15).

A very special thanks goes out to my parents for the support and encouragement they provided me through my entire life. Finally, I would like to thank Doctor Perdita Stevens and Professor Teresa Attwood for taking time out from their busy schedules to serve as my internal and external readers. In conclusion, I recognize that this research would not have been possible without the financial assistance of the Medical Research Council and the Informatics Graduate School at the University of Edinburgh, and express my gratitude to both those agencies.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Catherine Canevet)

Table of Contents

1	Introduction	1
1.1	The need for text mining in biomedicine	2
1.2	Protein subcellular localisation	6
1.2.1	A brief overview of cell structure	7
1.2.2	Motivation behind protein localisation	10
1.2.3	Biological methods used to ascertain protein localisation	11
1.2.4	Examples of protein subcellular localisation databases	13
1.3	The Nuclear Protein Database (NPD)	15
1.4	The NPD Curator System Interface	21
1.5	Claims of the thesis	21
1.5.1	Claims	22
1.5.2	Evidence the claims are based on	22
1.5.3	Contributions made in supporting claims	22
1.6	Structure of the thesis	23
2	Background	25
2.1	Overview of text-mining	25
2.1.1	IR and TC	26
2.1.2	IE and NER	27
2.1.3	Evaluation measures	28
2.2	Background on general methods and tools	32
2.2.1	Part-Of-Speech (POS) tagging	32
2.2.2	Stem	33
2.2.3	Stop words	33
2.2.4	TF.IDF	33
2.2.5	Machine Learning (ML)	33
2.2.6	WEKA	33

2.2.7	Naive Bayes (NB)	34
2.2.8	Decision Tree (DT)	35
2.2.9	Maximum Entropy (MaxEnt)	38
2.2.10	Support Vector Machine (SVM)	38
2.3	Related work on tools detecting relevant information in abstracts and full-text articles	38
2.3.1	iHOP	39
2.3.2	BioRAT	39
2.3.3	BioIE	40
2.3.4	METIS	41
2.3.5	EBIMed	42
2.3.6	TXM project	42
2.3.7	PolySearch	44
2.3.8	FACTA	46
2.4	Previous work reported in open evaluation contests	46
2.4.1	KDD Challenge Cup 2003	47
2.4.2	TREC Genomics tracks	48
2.4.3	TREC 2006 QA track	50
2.4.4	TREC Novelty tracks	51
2.5	Related work on machine-assisted database maintenance	51
2.5.1	BRENDA	52
2.5.2	FlyBase	52
2.5.3	Protein fingerprint database (PRINTS)	53
2.6	Summary	54
3	Retrieval of relevant sentences in full text biomedical papers	55
3.1	Text pre-processing	55
3.2	Set of gazetteers	57
3.3	Corpora	57
3.3.1	Training set	58
3.3.2	Test set	61
3.4	Supervised method	62
3.4.1	Set of features	62
3.4.2	From sentences to feature vectors	65
3.4.3	Baseline	70

3.4.4	Experiments and Results	70
3.4.5	Analysis of the results	72
3.5	Rule-based method with BioIE	74
3.6	Unsupervised method: Vector Space Models with Infomap	76
3.7	Discussion	79
3.8	Summary	83
4	Elements of automated annotation assistance	85
4.1	Document retrieval	85
4.1.1	Text categorisation task	85
4.1.2	Format issues	88
4.1.3	Abstracts vs. Full Text	88
4.2	Relevance detection	92
4.3	Redundancy detection	93
4.3.1	Introduction to Ixtransduce	94
4.3.2	Grouping sentences manifesting the same localisation relations	95
4.3.3	Results	95
4.3.4	Discussion	96
4.4	Novelty detection with regards to the NPD	97
4.4.1	Getting the latest version of the NPD	97
4.4.2	Checking NPD flat file	98
4.4.3	Checking aliases	98
4.4.4	Results and assessment	98
4.5	Highlighting sentences related to a localisation relation using colour codes	101
4.6	Summary	102
5	The NPD Curator System Interface for annotation assistance	103
5.1	The Curator-System Interface	103
5.1.1	Homepage	103
5.1.2	Documents retrieval	103
5.1.3	Retrieval results	106
5.1.4	Highlighted full text	106
5.1.5	Summary results	106
5.1.6	Highlight info type	106
5.2	Evaluation	111

5.2.1	Comparing amount of information extracted	112
5.2.2	Comparing amount of time spent annotating	115
5.2.3	Comparing amount of time wasted on FPs	115
5.2.4	Conclusion and future work	115
5.3	Summary	116
6	Extensibility and maintainability	117
6.1	Extensibility	117
6.1.1	From the nucleolus to the nucleus and other compartments	117
6.1.2	From the nucleus to the cell	119
6.1.3	From inside the cell to outside the cell	119
6.1.4	From localisation to function	121
6.1.5	Unsupervised learning for extensibility	124
6.1.6	Generic re-usable elements	124
6.2	Maintainability	125
6.2.1	MRes tool	125
6.2.2	Retrieving full text	126
6.2.3	Updating the protein names lexicon	126
6.3	Summary	127
7	Epilogue	129
7.1	Conclusions	129
7.1.1	Claims of the thesis revisited	129
7.1.2	Contribution to the field	129
7.2	Future work	130
A	Abbreviations	135
B	The Cell Component Ontology (CCO)	139
C	The compartment name lexicon	143
D	The protein keyword lexicon	147
E	The list of stop words	149
F	Examples of sentence classification	153
F.1	Third sentence of the abstract of article [RRB ⁺ 03]	153

F.2	Title of article [CSK98]	153
F.3	Fourth sentence of the abstract of article [KZCJ02]	153
	Bibliography	157

List of Figures

1.1	Growth of MEDLINE publications	2
1.2	Growth of MEDLINE searches	3
1.3	From experimental data to publications to databases	4
1.4	The cell and its nucleus	8
1.5	Protein synthesis: transcription and translation	9
1.6	The nucleolus and its three distinct zones	10
1.7	The nuclear compartments	11
1.8	The NPD entry for protein BIG1	17
1.9	The NPD nuclear compartment browser	19
1.10	The NPD's old annotation process	20
2.1	From implicit to explicit knowledge	26
2.2	Example of a text categorisation tool	27
2.3	Example of IE in the biomedical domain. Information can be extracted from relevant free text and put into a structured format.	28
2.4	Confusion matrix	29
2.5	EBIMed screenshot	43
3.1	Learning curve on the size of the training corpus showing there are enough labelled documents in this collection.	69
4.1	Categorisation of PMIDs	86
4.2	Example of email sent to curator	87
4.3	PubMed entry for [PPRMV04]	90
4.4	PubMed entry for [SSE ⁺ 98]	91
4.5	From raw text to domain-specific annotation	94
5.1	Screenshot of the NPD Curator System Interface Homepage	104
5.2	Screenshots showing the documents retrieval tool	105

5.3	Screenshot of the “Highlighted full text” page (abstract only)	107
5.4	Screenshot of the “Highlighted full text” page (full text)	108
5.5	Screenshot of the “Summary results” page	109
5.6	Screenshot of the “Highlight info type” page	110
5.7	Graphs showing time logs of the tool	113

List of Tables

1.1	Content comparison of four subcellular localisation databases	15
1.2	Localisations included in four subcellular localisation databases	16
2.1	Sentence classification results obtained by METIS	41
3.1	Categories of NE, their abbreviation and number of instances in each gazetteer	58
3.2	Number of sentences in the training corpus	59
3.3	Detailed description of the training corpus	61
3.4	Number of sentences for each paper in the test set	61
3.5	C&C tagger’s output	66
3.6	Feature vector’s possible values	67
3.7	10-fold cross-validation on the training corpus	68
3.8	Baseline results	70
3.9	Results on the test set	71
3.10	Ensemble results on the test set	71
3.11	Results on the test set @n	72
3.12	‘DIY’ cross-validation results on the training corpus	73
3.13	Results on the test set A@n	74
3.14	BioIE templates for the localisation category	75
3.15	Confusion Matrix for BioIE on training set	76
3.16	Results produced by BioIE on the same test set	76
3.17	Results produced by BioIE on the test set A@n	76
3.18	Normalised term-sentence matrix	77
3.19	Results produced by Infomap on the same test set	78
3.20	Results produced by Infomap on the test set A@n	78
3.21	Infomap’s results on the training set	79
3.22	A@n results obtained by 3 different methods on test set	81

4.1	Number of sentences covering the most important facts of two articles	89
4.2	Grouping precision on training and testing corpora	96
4.3	Precision of sentences highlighting	101
5.1	Scores indicating the usefulness of the tool	111
5.2	Average time spent on publications	114
F.1	C&C tagger's output for 3rd sentence of [RRB ⁺ 03]	154
F.2	C&C tagger's output for title of [CSK98]	155
F.3	C&C tagger's output for 4th sentence of [KZCJ02]	156

Chapter 1

Introduction

Biologists used to study particular proteins and their interactions until entire genomes and proteomes were discovered. Since then, scientists have been able to take a more global approach to their research work by, for example, analysing all the factors involved in a pathway at the same time. Leading their research in such a manner is only possible thanks to large amounts of data being made available. Furthermore, this also relies on relevant data being easily accessible.

The biological literature is a major and rapidly expanding repository of knowledge. In order to make this source of data accessible, information can be extracted from it and stored in databases where information is easier to find. This process can be achieved by domain experts. They can read articles, select what is of interest to them and annotate accordingly their particular database. However, it is now virtually impossible for curators to read or even skip through all the articles that are being published. This issue means experts in the biomedical domain are craving solutions that can help them save time without missing important information.

The first section of this chapter explains this need for text mining in biomedicine. It also stresses the need for text mining on full text rather than on abstracts, and discusses some of the additional problems that working on full-text papers introduces. The second section presents existing protein subcellular localisation databases, and stresses why working on protein localisation is important. The third section then introduces the Nuclear Protein Database (NPD), which is the subcellular localisation database that I have worked with. The fourth section presents the NPD Curator System Interface, the final tool I developed for this database. Finally, the claims of the thesis are given in the fifth section while the last section gives an overview of the structure of the thesis.

1.1 The need for text mining in biomedicine

Recent technological advances have driven the development of increasing high throughput experimental methods in biomedicine, with a resulting explosion in data. In most research areas - and biomedicine is no exception - scientists publish their results in conference articles and journals. This large number of data coming out has triggered an exponential growth of publications. Whereas Figure 1.1 [RSKC05] shows the growth in MEDLINE publications, Figure 1.2 (from http://www.nlm.nih.gov/bsd/medline_growth.html) presents the growth in MEDLINE searches. It is interesting to see that from 2002 to 2004 - in the space of only two years - the number of publications in MEDLINE rose by about half a million (augmenting in a regular manner - see exponential curve on Figure 1.1) while MEDLINE searches increased by 20 million. Although this definitely shows scientists use MEDLINE more and more, these graphs do not tell us how many articles scientists read a year and, more to the point, how many full-text papers they actually take time to entirely study. The truth is not that many, as they simply cannot afford the time to do so.

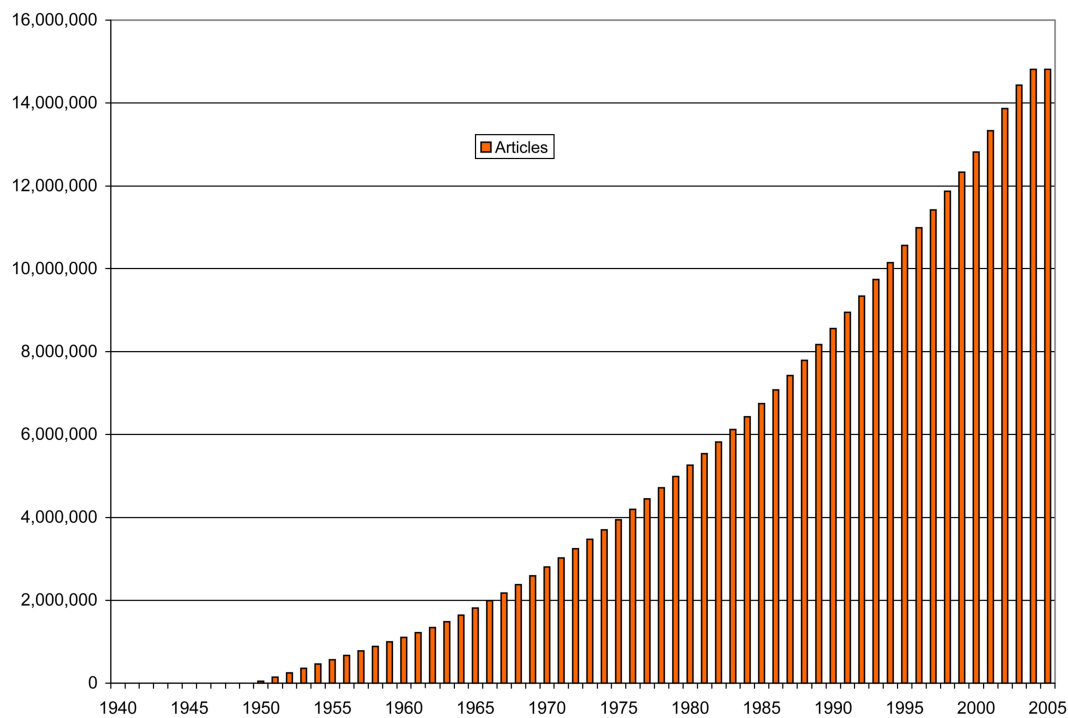


Figure 1.1: Growth of MEDLINE publications [RSKC05].

It has become extremely difficult for researchers to find specific evidence in the literature they need in developing hypotheses, designing experiments or interpreting their

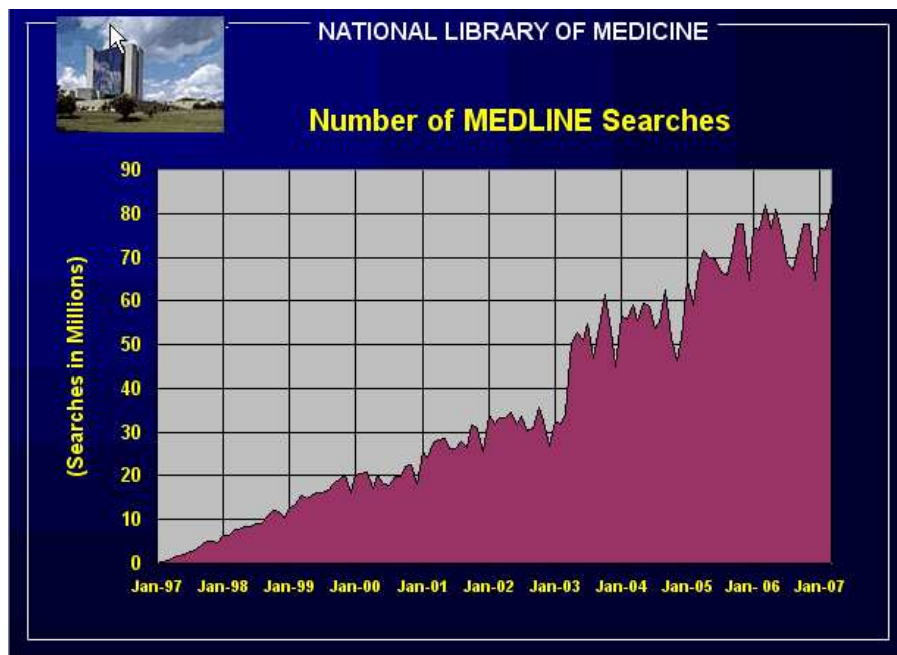


Figure 1.2: Growth of MEDLINE searches, http://www.nlm.nih.gov/bsd/medline_growth.html

results, as it is a very time-consuming task. There is a real need for computer scientists to help biologists with this particular issue. Although biomedical text mining cannot deliver results with 100% accuracy, it can provide biologists with a semi-automatic approach, which can save them a lot of time and hassle, and let them make the final and all important decisions that, in some cases, only human experts can take. This relatively new research field has evolved to cover this need, taking advantage of computational linguistics techniques (see Section 2.1) as well as domain-specific resources. Figure 1.3 illustrates this issue.

Natural Language Processing (NLP) is a very complex field of research in itself. BioNLP, which represents NLP applied to the biomedical domain, is an even more difficult task as the nature of its named entities (NEs) is much more variable. For example, according to Acromine¹ [OA06], “CAT” stands for 64 acronyms ranging from “chloramphenicol acetyltransferase” to “computer assisted tomography”, not to forget that cat, in a biomedical article, could also refer to an animal. Indeed the key issue is disambiguation of NEs to avoid extracting information erroneously.

The incentives for research in bioNLP are numerous. It can help database curators save time and/or extract more/richer information. It can help bench scientists improve

¹Acromine is available from <http://www.nactem.ac.uk/software/acromine/>

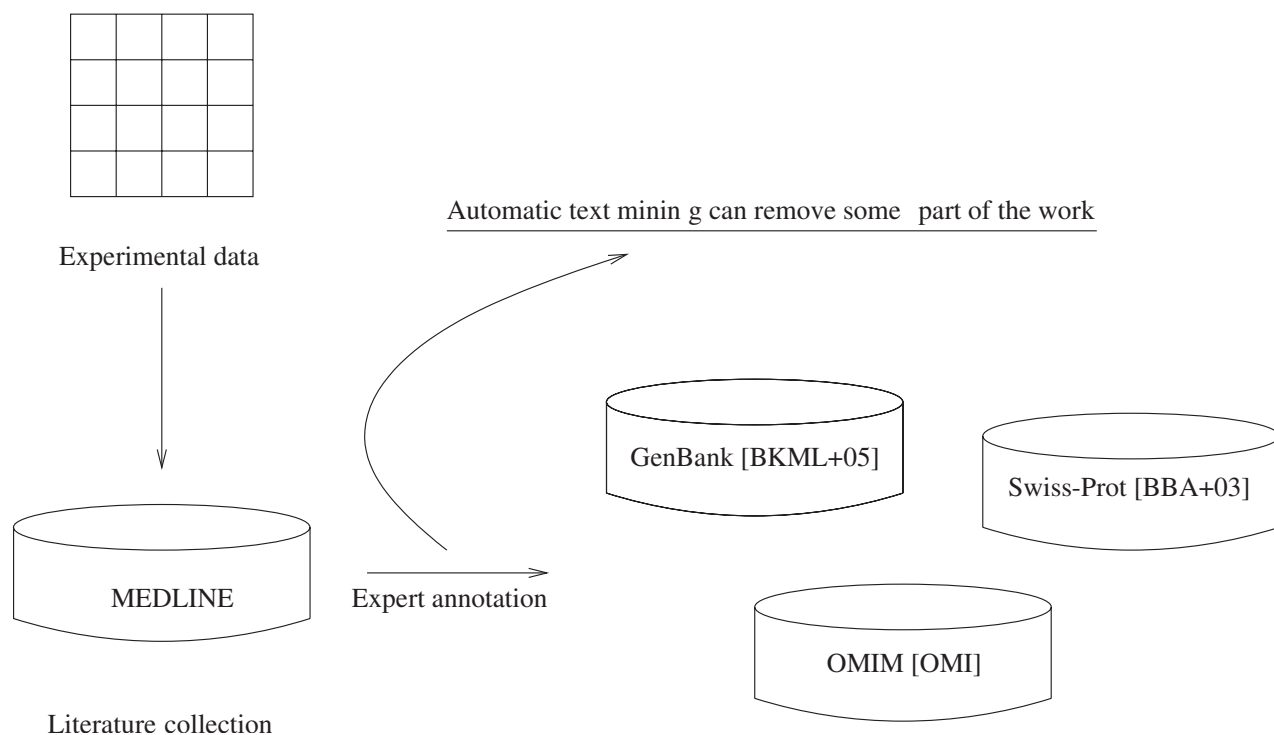


Figure 1.3: From experimental data to publications to databases. Researchers publish their experimental results in articles. In turn, information can be extracted from these publications and stored into databases. Expert annotation can be assisted and annotation time speeded up using text-mining techniques.

searches of sequence databases. Moreover, most bioscientists cannot keep up with the literature getting published in their field and bioNLP can support them in doing so and save them precious time.

The main achievements of bioNLP so far have come from three kinds of approaches: co-occurrence-based methods (see examples in Sections 2.3.5 and 2.3.8), rule-based approaches (see examples in Sections 2.3.2 and 2.3.3) and finally statistical approaches (see examples in Section 2.3.6 and Chapter 3). The first type of approach is the least complex of the three as it mainly relies on Named Entity Recognition (NER, see Section 2.1.2) results. The second one can require a considerable amount of time to create the rules while the last one is based on Machine Learning (ML, see Section 2.2.5). Even the most sophisticated methods can still fail against complicated task such as the extraction of protein function (see Section 6.1.4).

BioNLP exploits the fact that the biomedical literature is accessible on the Internet. MEDLINE [med] is a database of bibliographic references to all biomedical papers. For each such paper, it includes its title, authors, journal location, abstract and numerous forms of valuable meta-data. PubMed [Pub] is a computer system used to access MEDLINE, as well as other databases. Full text is available in a variety of formats, under a variety of licensing agreements. Section 4.1.2 discusses this further.

Beyond format issues and availability, why is working on full-text publications more important, valuable and useful than working on abstracts only? As one might expect, there are more facts of interest in the full text of an article than in its abstract. In Chapter 4, I will review two articles that discuss this issue. As noted in Chapter 5, when evaluating the system developed here, Professor Wendy Bickmore extracted protein localisation relations from full-text articles that were not present in their abstract. Even though she used to manually extract information from abstracts only, she is now happy to curate extra information from full-text papers.

Moreover, important protein localisation relations in full-text papers can be identified by looking at how frequently they are mentioned (see Section 4.3). Whereas localisation relations that are important to a paper are generally present in the abstract, working on abstracts only would not allow this analysis of frequency. Section 4.1.3 discusses this further.

Working on full-text articles does introduce problems not encountered when working on titles and abstracts alone. First of all, simply locating full-text papers and downloading them is a more complicated process than getting abstracts, which are readily available through PubMed [Pub] in a consistent format.

Then, once the full-text document is at hand, the format it has been downloaded in can introduce further difficulties. As discussed in Section 4.1.3, HTML is the chosen format in this thesis work. HTML has its own challenges. For example, publications are presented online using several separate files. Indeed, the main text of the article and its Figures cannot be obtained in a single download.

Bickmore has mentioned many of the experimental results are shown in Figures and their captions. In most cases, captions are in fact part of the main text file. Traditionally, images were not handled by text-processing systems. Besides, Figure legends and Table captions tend to describe the material of interest they embody.

However, a few tools do work with images and their captions notably FigSearch [LJN⁺04] and more recently BioText [HDG⁺07]. BioText is a Web-based search engine that enables users to perform a text-mining search on abstracts or Figure captions. The results display the Figures along with the retrieved text. Image analysis is not an area of research I considered for my thesis work but it seems it is now becoming more prevalent. For example, a dedicated session will take place at the ninth annual meeting of BioLINK (Linking Literature, Information and Knowledge for Biology) Special Interest Group at the conference on Intelligent Systems for Molecular Biology for the first time in June 2009.

Moreover, full text in HTML format contains a lot of text that is not actually part of the article. In order to get a pure version of the text (free of any menus, links, *etc.*), one needs to pre-process the text before analysing it in any way. In Section 3.1, I present the text pre-processor I developed for this thesis work.

The system I have developed is applied to full-length documents rather than abstracts only and offers a complete annotation tool for a particular database. It is because the tool is targeted and domain-specific that it can generate relevant results. Moreover, the extensibility of the approach is discussed in Chapter 6 and a final experiment shows how the tool could be adapted in order to extract different kinds of information.

1.2 Protein subcellular localisation

A cell is the smallest unit of an organism that can function on its own (see Figure 1.4). Molecular biology studies all the interactions that take place in a cell, as well as the way these interactions are regulated for the cell to operate as it should in healthy organisms or faultily in diseased ones. In order to understand where these interactions may occur, the next subsection presents the main organelles contained in a cell.

1.2.1 A brief overview of cell structure

The **nucleus** is the core centre of the cell. It contains the cell's chromosomes and is highly important for the activities that occur therein. Indeed, most of the DNA replication and RNA synthesis take place in this organelle. Figure 1.5 shows the processes of transcription and translation, which result in the creation of new proteins. Nuclear sub-compartments do not have physical boundaries (unlike cytoplasmic ones). They represent places where proteins involved in similar biological processes are concentrated together.

The **nucleolus** (<http://npd.hgu.mrc.ac.uk/compartments/nucleolus.html>, [LTML05], [BvKNL07], see Figures 1.6 and 1.7) is the core of the nucleus and contains various components called the fibrillar centers (FC), the dense fibrillar components (DFC), the granular components (GC) and ribosomal DNA (rDNA). Its main function is to produce ribosomes, which will, in turn, themselves produce proteins.

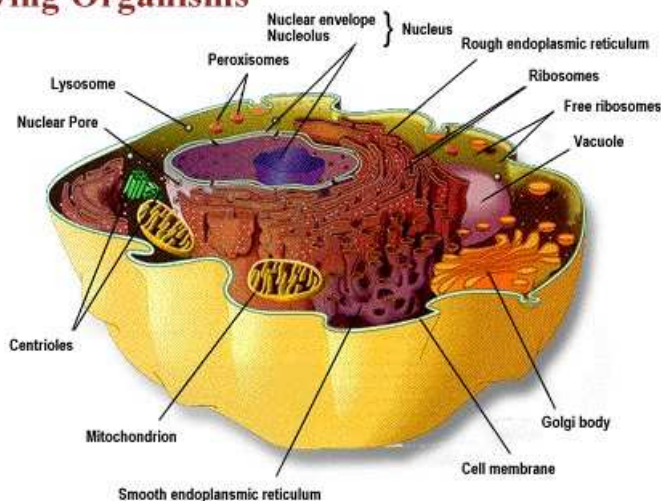
In recent years, other roles have been attributed to this sub-organelle, such as control of the cell cycle and production of other types of functional ribonucleoproteins (RNPs). The authors of [BvKNL07] expect that more functions are yet to be discovered and associated to the nucleolus. It is indeed a subnuclear compartment of paramount importance as its dysfunction has been linked to causing diseases such as genetic disorders and predisposition to cancer.

The nucleus also comprises Cajal bodies (CBs) and Gemini of coiled bodies or gems (<http://npd.hgu.mrc.ac.uk/compartments/gem-cajal.html>), promyelocytic leukaemia (PML) bodies (<http://npd.hgu.mrc.ac.uk/compartments/pml.html>), paraspeckles (<http://npd.hgu.mrc.ac.uk/compartments/paraspeckles.html>) and splicing speckles (<http://npd.hgu.mrc.ac.uk/compartments/speckles.html>). In [KID08], the authors show that CBs form by self-organisation after a certain concentration of snRNP is attained. They also suggest *de novo* formation is possible for other subnuclear bodies.

The nucleus is bounded by a double lipid membrane, perforated by the nuclear pores (see Figure 1.4). The nuclear lamina then underlies the inner nuclear membrane and is composed of intermediate filaments of lamins. It maintains the shape of the nucleus. It is also involved in various functions such as regulating transcription and DNA replication.

The paraspeckles play a role in transcription and splicing. RNA contain non-coding

Living Organisms



Animal Cell (eukaryote)

Figure 1.4: The cell and its nucleus (<http://bioweb.usc.edu/courses/2002-fall/documents/bisc150-2.html>). The nucleus is the core of the cell. It is bounded by the nuclear envelope. The nuclear pores allow materials to transit between the nucleus and the cytoplasm. The ribosome is responsible for the translation step of protein synthesis. The endoplasmic reticulum (ER) is divided into two categories: the rough ER (RER) and the smooth ER (SER). The ER helps with the transport of proteins as well as their folding. Only proteins that have been correctly folded can be transported from the RER to the Golgi body. The Golgi body then modifies proteins in order to enable them to reach their final cellular localisation. The mitochondrion is involved in aerobic respiration and oxidative metabolism (formation of ATP). The centrioles are constituted of microtubules and play a role in cell division. The peroxisome is a membrane-bound vesicle that contains over fifty enzymes carrying out diverse metabolic reactions. The lysosome is the digestive organelle of the cell. The vacuole is a temporary storehouse. Finally, the cell membrane is a barrier that provides a separation between the cell and its environment only allowing the exchange of specific substances.

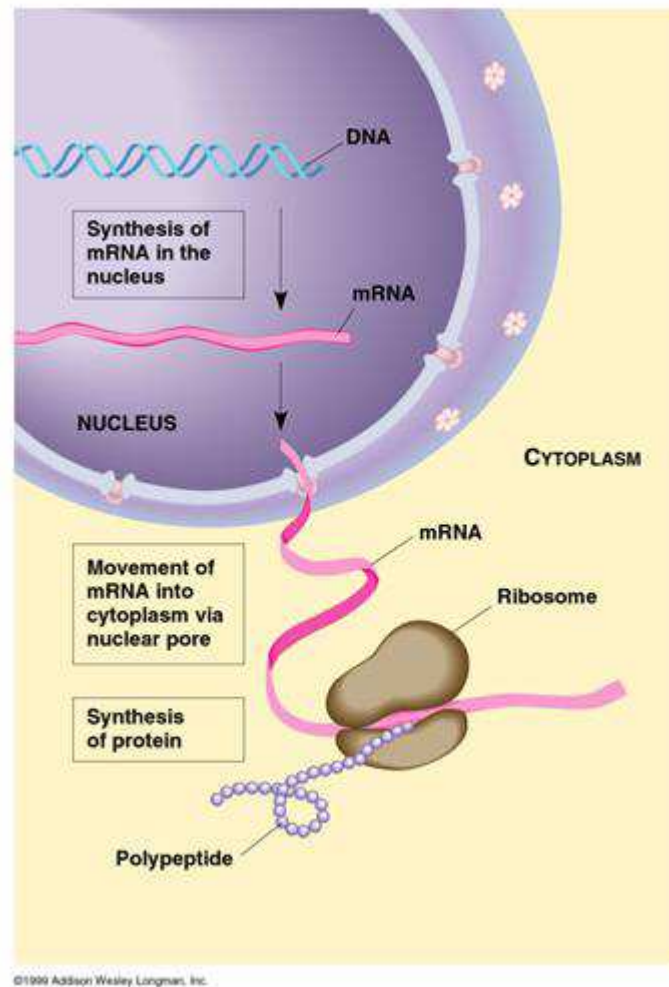


Figure 1.5: Protein synthesis is composed of two steps: transcription and translation. Transcription converts double-stranded DNA into single-stranded messenger RNA (mRNA). mRNA can leave the nucleus through a nuclear pore and, once in the cytoplasm, the translation step can take place. The ribosome can combine mRNA and transfer RNA (tRNA) to create a sequence of amino acids forming a polypeptide or protein. (<http://fajerpc.magnet.fsu.edu/Education/2010/Lectures/>)

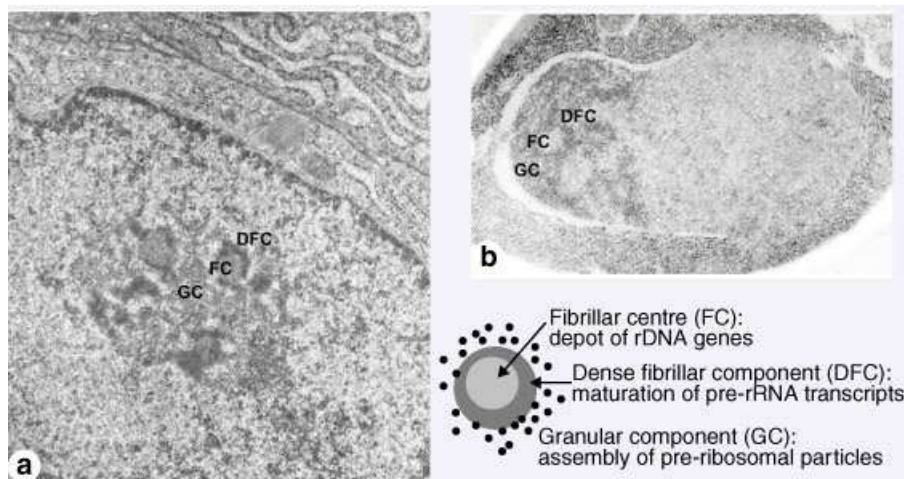


Figure 1.6: The nucleolus (<https://npd.hgu.mrc.ac.uk/compartments/nucleolus.html>) and its three distinct zones involved in ribosome biogenesis. DFC and GC are organised as surrounding layers of FC.

sequences, called introns, that need to be taken out before mRNA can participate in translation and create proteins. The process of splicing involves a spliceosome that binds to the mRNA and removes introns. The splicing speckles are used to store mRNA splicing factors.

A spliceosome is composed of several protein-RNA complexes, including small nuclear ribonucleoproteins (snRNPs). The CBs and gems play a role in snRNPs biogenesis. The PML bodies have been suggested to be involved in various activities such as transcription, DNA repair, cell cycle regulation, proteolysis and apoptosis.

1.2.2 Motivation behind protein localisation

A protein needs to be located near specific components that engage in a specific process in order for it to be capable of executing its function within that process. Hence the motivation to understand protein localisation as it gives important clues as to what processes a protein may be involved with. In [CC03a], the authors even explain how localisation is often used to elucidate protein function.

I will now review, in subsection 1.2.3, the methods used in localising proteins because what can be recorded about them in a database relies on the methods that have been used to capture the data. Subsection 1.2.4 gives examples of protein subcellular localisation databases and motivates the need for different types of subcellular localisation databases. The next section (Section 1.3) presents the database I have used in

developing a curator's assistant for my PhD thesis. This database focuses on one compartment of the cell, the nucleus (see Figure 1.7). As discussed in Section 1.2.1, this particular component's malfunction has been shown to lead to illnesses. Determining the subnuclear localisation of proteins allows us to understand genome regulation and to set us on the trail of discovering the function behind proteins.

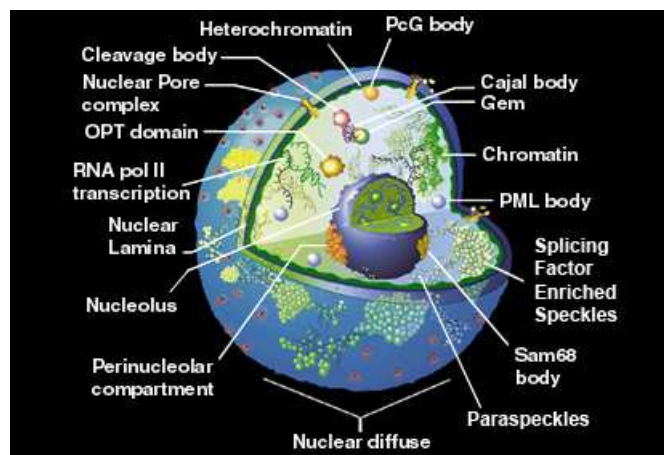


Figure 1.7: The nuclear compartments (<http://npd.hgu.mrc.ac.uk/compartments.html>). The nucleolus is the core of the nucleus. Other nuclear compartments include: PcG, PML, Sam68 bodies, perinucleolar compartment, cleavage body, nuclear pore complex, chromatin and heterochromatin, Cajal body and gem, paraspeckles and splicing speckles.

1.2.3 Biological methods used to ascertain protein localisation

This subsection introduces the different methods biologists use to resolve protein localisation. The nature of the information entered in databases about protein localisation indeed relies on the kind of techniques used to localise proteins in the first place. The final paragraph of this subsection explains what terms can be used to describe the localisation of a protein depending on what technique was used.

Because the subjects of study in cell and molecular biology are very small in size, **microscopy** uses instruments called microscopes to provide scientists with an enlarged image of their subjects. There are two principal types of microscopy used in biology - light and electron. The former uses visible light to illuminate the subject whilst the latter directs an electron stream at the subject.

The advantages of light microscopy over electron microscopy are: multiple colour

detection (*i.e.* one can look at more than one thing at a time), ability to use fluorescence (*i.e.* sensitivity), and the fact that it can be used on living cells (*i.e.* it is possible to perform dynamic studies). Electron microscopy can only be done on fixed (*i.e.* dead) material. It has high resolution (showing details about topography, morphology, composition of the subject as well as crystallographic information) but low sensitivity.

Biologists use fluorescence microscopy to ascertain the localisation of a protein [GAET06]. There are two techniques they can use, the first one is called **GFP-tagging**. It uses a gene isolated from jellyfish that encodes a protein called Green Fluorescent Protein (GFP). As its name suggests, this small protein emits a green fluorescent light when it is excited by short wavelength light. The technique involves fusing the DNA encoding GFP to the DNA encoding the protein of interest. The fusion is assumed to have no side effect to the movement or the function of either proteins. The resulting composite DNA is put into a cell, which can be analysed using a microscope. There are now variants of GFP available in different colours. Its advantage is that it can be used in living cells.

The second technique of fluorescence microscopy is called **immunofluorescence**. In immunofluorescence, antibodies that are coupled to fluorochromes are used, but this is for fixed (*i.e.* dead) cells. Humoral immunity is carried out by antibodies which are blood-borne proteins. They are created by B lymphocytes in order to prevent foreign materials from entering a host cell. For this technique, antibodies need to be specifically prepared against the protein of interest. The prepared antibody molecules are then linked to a substance that makes them visible under the microscope, but that has no other side effect. When an antibody's binding site recognises a specific target antigen, it binds to it. Immunofluorescence offers excellent clarity because whilst the proteins bound by the antibody are revealed, the rest of the materials in the cell remain invisible.

Biochemistry uses techniques such as homogenisation to break up cells and isolate organelles. Cell fractionation provides an insight of the molecular composition of a structure. A protein's location can be determined by observing the co-purification of proteins in a biochemical fraction. **Mass spectrometry (MS)** is generally used to identify the proteins in a purified biochemical fraction. It involves measuring the mass-to-charge ratio of ions in order to determine the composition of nuclear compartments [TML07]. Another technique recently developed was **LOPIT**, which stands for Localisation Of Proteins by Isotope Tagging. It is a high-throughput method to locate proteins by MS where pure organelles are not required. In [DHS⁺06], the authors

combine this technique with 2D liquid chromatography of peptides and tandem MS in order to map the *Arabidopsis* organelle proteome. **Immunoprecipitation** enables proteins that interact with each other to be identified. During immunoprecipitation, a protein antigen is precipitated out of solution by using an antibody that targets this protein. Once isolated, proteins that interact with it can be identified by Western blotting or MS. Western blot separates proteins using gel electrophoresis. **Yeast two-hybrid analysis** also checks for protein-protein interactions, or “Bait”-“Prey” interactions. In two-hybrid screening, a transcription factor is fragmented into two parts: a binding domain (BD) and an activating domain (AD). Two fusion proteins are then prepared. They correspond to “BD + Bait” and “AD + Prey”. When Bait and Prey interact, the transcription factor is indirectly connected and activates transcription. Therefore if transcription occurs, this means the two proteins interact.

While microscopy gives better resolution than biochemistry, the latter allow biologists to look at molecular composition and thereby study things at a larger scale. Once they have identified what particular organelle they need to look at, microscopy can provide them with more precise images. Different techniques can be combined to obtain final results.

Some of these approaches can be associated with certain kinds of terms used in the biomedical literature to describe the subcellular localisation of proteins. For example, MS often describes proteins as “co-purifying with”. Other keywords linked to this technique are “associate”, “contain”, “include” and “interact”, while terms describing localisation established utilising microscopy include “concentrate”, “detect”, “found”, “move”, “observe”, “present”, “seen”, “shuttle”, “stain”, “traffick”, “transit”, “visible”.

1.2.4 Examples of protein subcellular localisation databases

Databases in cell biology vary depending on what information they contain. Some only comprise data about plants, some about one or more particular model organisms, some ignore temporal aspects of when a protein is in a particular location, some contain information about a protein’s normal location, while others may talk about mutated alleles of the genes producing those proteins. It is because of this variety that there are so many different databases available.

There are three domains of life: the archaea (archaeobacteria), the bacteria (eubacteria) and the eukaria (eukaryotes). The archaea and the bacteria are both prokaryotes.

Eukaryotes have a nucleus whereas prokaryotes do not. Even though there is a lot of diversity within the eukaryotes, a handful are considered representative of them and are therefore called *model organisms*.

DBSubLoc [GHJS04] (<http://www.bioinfo.tsinghua.edu.cn/dbsubloc.html>) is a database that houses information about protein subcellular localisation in general that has been collected from model organism genome projects (covering bacteria, eukaryotes, fungi, plants, animals, viruses and archaea), other databases as well as information found in the literature.

PSORTdb [RAG⁺05] (<http://db.psort.org/>) is a database that houses information on subcellular localisation for bacteria. It actually divides into two separate sets: ePSORTdb, whose data come from experiments, and cPSORTdb, which stores results from computational predictions. As none of the other databases presented in this thesis are based on computational predictions, ePSORTdb is chosen as the dataset of interest between the two, which will be compared to other databases in this section.

Arabidopsis thaliana is the model organism of choice for plants and, for this reason, was extensively studied, which resulted in it being the first plant to have its whole genome sequenced. The SUBcellular location database for *Arabidopsis* (**SUBA** [HVTF⁺07], <http://www.suba.bcs.uwa.edu.au>) contains information on subcellular localisations of proteins in *Arabidopsis*, which ranges from results of experiments to data from other databases and literature references.

LOCATE [FAD⁺06] (<http://locate.imb.uq.edu.au/>) is a database that provides information on the membrane organisation and subcellular localisation of proteins from mouse and human. In [FAD⁺06], the authors explain that the membrane organisation is the result of computations prediction, and subcellular localisations data comes from experiments as well as information extracted from the literature.

Table 1.1 compares the data contained in the four databases introduced above. Two out of the four are specific to one organism only. The four databases contain between 2165 and 30357 protein entries each. While DBSubLoc contains 1367 non-redundant proteins for the organism “Plant”, its dataset is not specific to *Arabidopsis* and does not include as many proteins as SUBA (6743 non-redundant proteins), which is a database dedicated to *Arabidopsis*, the model organism for plant biology.

The number of localisations these proteins are associated with varies from 6 to 31. Table 1.2 compares the different localisations included in these four databases. They all contain “extracellular” as a localisation. Four localisations out of 31 are in 3 of the 4 databases: “cellular component unknown”, “cytoplasm”, “mitochondria”

	Organisms	Localisations	Non-redundant proteins
DBSubLoc	7 (list in text)	8	30357
ePSORTdb	1 (bacteria)	6	2165
SUBA	1 (<i>arabidopsis</i>)	13	6743
LOCATE	2 (mouse and human)	31	8076 (mouse) / 9108 (human)

Table 1.1: Content comparison of four subcellular localisation databases

and “nucleus”. The number of different entries associated with the Golgi body in the Table shows where some of the text-mining challenges lie. It is indeed very difficult to recognise entities of interest when one concept can be represented by an undeterminate list of terms. Section 2.1.2 will introduce NER in the next chapter.

Although LOCATE offers information on subcellular localisation of proteins from mouse and human, it is not as specialised as, for example, a database that would provide a high level of detailed information on subnuclear localisation of proteins from mouse and human, which would not be available anywhere else. The NPD was created to cover this empty gap of knowledge representation in subcellular localisation databases. The next section (Section 1.3) introduces the NPD, which I have been working with for my PhD thesis.

Given how important protein subcellular localisation information is, it is no wonder that the number of such databases is growing. All these databases share the same kinds of information, *i.e.* protein entries with details of their localisations under different conditions. Section 6.1 explains how extensible my work on the NPD Curator System Interface is to other databases.

1.3 The Nuclear Protein Database (NPD)

The NPD [DFB03] is a database that houses data on over 2200 mouse and human proteins that have been reported to be localised within 40 locations of the cell nucleus. Comparing these facts to results in Table 1.1, it seems the NPD is closest to LOCATE as they both contain data on proteins from the two same organisms (mouse and human). However, the NPD contains a lot less protein entries and also identifies more and different types of localisations. LOCATE and the NPD share six location fields (nucleus, cytoplasm, Golgi, centrosome, ribosome and ER) and differ on the rest of them. LOCATE contains subcellular localisation terms whereas the NPD focusses on

Localisations	DBSubLoc	ePSORTdb	SUBA	LOCATE
apical plasma membrane				X
basolateral plasma membrane				X
centrosome				X
cell plate			X	
cell wall		X		
chloroplast	X			
cytoplasm	X	X		X
cytoplasmic vesicles				X
cytoplasmic membrane		X		
cytoskeleton			X	X
cytosol			X	
endoplasmic reticulum			X	X
endosomes				X
early endosomes				X
late endosomes				X
ERGIC				X
extracellular	X	X	X	X
golgi apparatus			X	X
medial-golgi				X
golgi cis cisterna				X
golgi trans cisterna				X
golgi trans face				X
lipid particles				X
lysosomes				X
melanosome				X
membrane	X			
outer membrane		X		
mitochondria	X		X	X
inner mitochondrial membrane				X
outer mitochondrial membrane				X
nucleus	X		X	X
periplasmic		X		
peroxisome			X	X
plasma membrane			X	X
ribosome	X			
secretory granule				X
synaptic vesicles				X
tight junction				X
transport vesicle				X
vacuole			X	
cellular component unknown	X		X	X

Table 1.2: Localisations included in four subcellular localisation databases

Nuclear Protein Database

Home About Links Statistics Help Credits

Names

Main	Species
Big1	Homo sapiens

Other

Other	Species
Brefeldin A-inhibited GEP 1	
p200 ARF-GEP1	

Keywords

Keywords: nuclear periphery, nucleolus

Subnuclear Localization

Stage	Description	Detail	Links
Interphase	Golgi	in growing cells	PubMed:10393931
Interphase	nuclear periphery	after serum starvation	PubMed:14973189
Interphase	nucleolus	after serum starvation	PubMed:14973189
Interphase	cytoplasmic	in growing cells	PubMed:14973189

Location

Location	Species
8q13	Homo sapiens

Sequences - Protein

Detail	Domains	MwPredict (kDa)	PIPredict	MwActual	PIActual	AANo	Accession	Links
Details	NLS, sec7	208.7	5.62			1849	Entrez Protein Q9Y6D6	Entrez Protein Q9Y6D6 OMIM 504131 Uniprot Q9Y6D6

Function - Molecular

ID	Term	Links
GO:0005086	ARF guanyl-nucleotide exchange factor activity	IDA PubMed:10393931

History

Created At	Changed At
2004-03-03 16:10:24	2004-11-08 16:16:09

Done

Figure 1.8: The NPD entry for protein BIG1, <http://npd.hgu.mrc.ac.uk/search.php?action=builddetails&geneid=1NP01781>. The first section “Names” gives the main name of the protein, as well as other aliases, and the species the protein is found in. The section “Keywords” gives keywords that can be associated with the protein. The section “Subnuclear Localization” lists the different cell compartments the protein can be located in, specifying at what stage of the cell cycle that would occur. Extra details are given as well as link(s) to a relevant publication where experimental evidence can be found. The section “Location” gives the cytogenetic position of the protein, expressed in coordinates based upon the staining of chromosomes. As noted in http://npd.hgu.mrc.ac.uk/About_NPD.html, the section “Sequences - Protein” provides information on the amino acid sequence, predicted protein size and isoelectric point, as well as any repeats, motifs or domains within the protein sequence, along with links to other databases (*e.g.*, Entrez, Swiss-Prot, OMIM, PubMed and PubMed Central). The section “Function - Molecular” gives a GO (Gene Ontology) term along with its ID and link(s) to a relevant publication where experimental evidence can be found. The section “History” shows when the entry was created and modified.

subnuclear terms, as shown in Appendix C. Therefore, the NPD covers a gap in the knowledge representation of subcellular localisation databases that no other database provides.

The original motivation behind the NPD was a project which was going to result in the identification of a hundred novel nuclear proteins. The idea was to combine these discoveries with publicly available data on nuclear proteins to create a very specific database. Undeniably, central databases containing data on entire genomes are very much needed. However, smaller databases that contain specific information can be extremely useful too. In making this point, Tom Mistelli used the NPD (<http://jcs.biologists.org/cgi/content/full/115/14/2805>) as his example and wrote:

“A good example of a relatively small, but focused and highly practical database is the recently launched NPD - nuclear protein database.”

In [DFB03], the authors present the database as giving information on protein sub-nuclear localisations at different stages of the cell cycle, as well as the amino acid sequence, predicted protein size, isoelectric point, repeats, motifs or domains within the protein sequence.

GO [ABB⁺00] terms are attributed to the proteins' biological and molecular functions. The database is also very well cross-referenced to other databases, such as Entrez [Ent], Swiss-Prot [BBA⁺03], Online Mendelian Inheritance in Man (OMIM [OMI]), PubMed [Pub] and PubMed Central [pmc]. Figure 1.8 presents a screenshot of what the database displays for the entry regarding protein Big1. The entire database is easily searchable using most fields mentioned above. The Website also offers the possibility to browse by nuclear compartments (see Figure 1.9) or by domain.

Until 2004, the database was maintained by only one human curator: Professor Wendy Bickmore. Bickmore used to get a daily email from PubCrawler [HW04] giving her the new PubMed articles that had been retrieved from the daily PubMed update by the search she had chosen to set, using keywords such as “chromosomes”² and “nucleus”³. This original annotation process is shown in Figure 1.10.

Generally, the process of manual maintenance of databases involves domain experts reading articles and extracting relevant facts for the database at hand and - although annotation processes differ from one database to the other - this usually turns out to be rather time-consuming. As explained earlier in this chapter, text-mining tools

²The “chromosomes” search uses the following keywords search string: “chromosome structure” or “chromatin” or “heterochromatin” or “silencing” or “histone”

³The “nucleus” search uses: “nuclear structure” or “lamina” or “nucleolus” or “nuclear bodies” or “splicing speckle”.

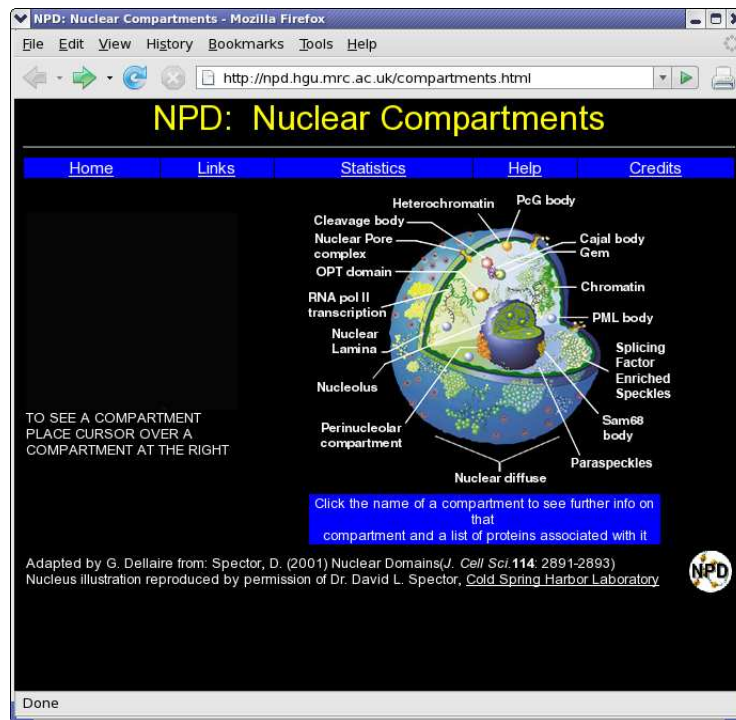


Figure 1.9: The NPD nuclear compartment browser, <http://npd.hgu.mrc.ac.uk/compartments.html>. By selecting a nuclear compartment, users are given its description as well as a list of all the proteins associated with that specific compartment contained in the database.

can help specific annotation tasks. Section 2.5 presents examples of databases whose maintenance is assisted in this way.

In conclusion, the NPD is a small database where the quality of precise information matters a lot more than quantity. The content of the database is extremely trustworthy as each entry has been carefully annotated by an expert. The text-mining work achieved in this thesis can assist such an expert in their thorough annotation work, thereby reducing the amount of time needed as well as increasing the amount of information added to the database. On the one hand, this is atypical in the sense that it is a much smaller database compared to the major ones (*e.g.*, Swiss-Prot [BBA⁺03]). On the other hand, it is typical as most curator assistant tools only provide semi-automatic approaches to their users and usually leave the final decisions to the domain experts (see Section 2.5).

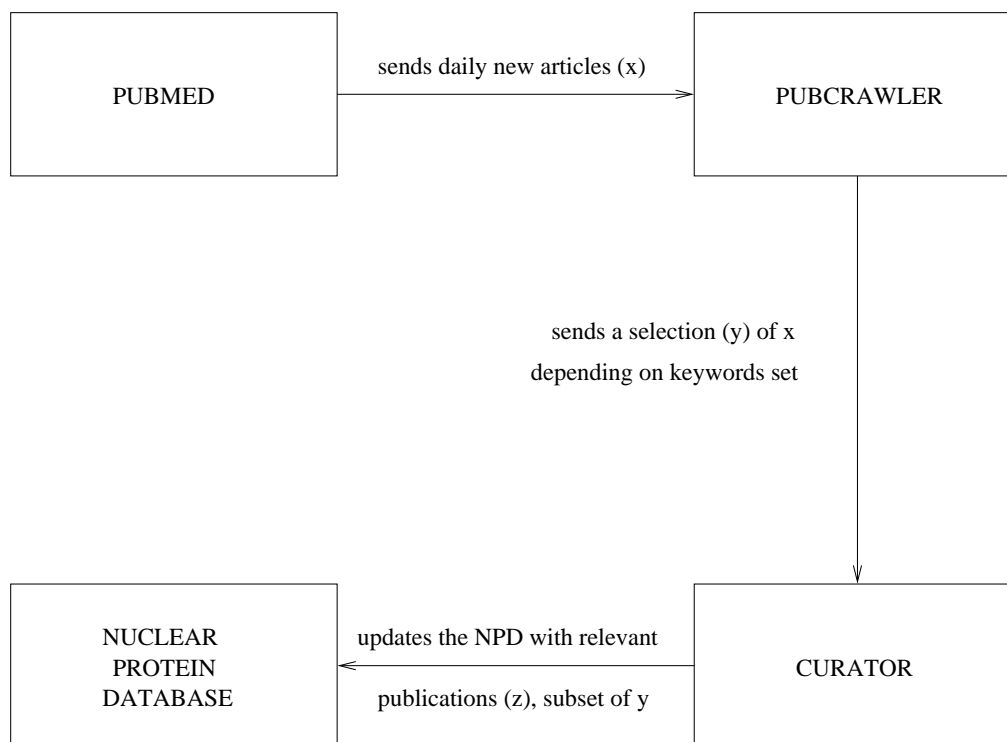


Figure 1.10: The NPD's old annotation process. The curator had set two keyword searches in PubCrawler, a free alerting service which gets updates from PubMed every day. PubCrawler would email the curator daily with a list of articles of interest. Based on the title and the abstract of each publication, the curator would decide which papers were relevant to the NPD and which were not. Out of the relevant ones, the curator would study the abstract and annotate an NPD entry with more information and a link to the particular article, or create a new database entry if required and update it accordingly.

1.4 The NPD Curator System Interface

This section gives a brief summary of what tools the interface for the curator system developed in this thesis work provides a curator with in order to introduce Chapter 2. A detailed overview of all the different features the interface supports will be given in Chapter 4. The interface allows users to either upload a file containing PubCrawler's results (in their normal email format) or simply type a PubMed identifier (PMID) of interest. If PubCrawler's results were uploaded, the curator is provided with a rank-ordered list of PMIDs. Clicking on a PMID will launch the same results as typing a PMID from scratch in the first place. If full text is found in HTML format, then the curator is presented with a full-text analysis of the paper. The interface works with the abstract otherwise.

In this thesis, I refer to "relevant sentences" or "sentences of interest" as sentences that carry information about the subnuclear localisation of one or several protein(s), unless stated otherwise (*e.g.*, in Chapter 6). Therefore, "irrelevant sentences" are sentences that do not contain any information about protein subnuclear localisation.

For the analysis of a paper, a first page highlights in yellow sentences classified as relevant. A second page displays a summary Table of all the different types of protein localisations found in the paper, rank-ordered by types that contain the highest number of sentences, with links to these sentences (which takes the user to the following page), and extra columns giving details as to whether this represents novel information to the NPD. The final page displays the text using a colour-coded highlighting of relevant sentences to a particular type of protein localisation information (selected in the previous page).

1.5 Claims of the thesis

Manual database annotation and maintenance is hard and time-consuming. Firstly, it is difficult to find articles that actually contain new facts supported by evidence. Moreover, finding new evidence-supported facts in an article is very time-consuming because it involves checking that there are experimental results backing up those facts somewhere in the full-text paper which - without any assistance - requires spending time reading it more or less completely.

The aims of this research work are to automatically find relevant information that has not yet been extracted into a biomedical database devoted to such information,

as one step in minimising the amount of time a human expert will need to spend to keep this database up to date. Detecting redundancy is a research topic that has been addressed in the newswire NLP domain (see Section 2.4) but not in the BioNLP area. In the newswire domain, news overlap and the structure of the articles is different from articles in the biomedical literature. It seems both types of article contain repetition of the information but in different places of the paper (see Section 4.3).

1.5.1 Claims

Not all facts of relevance for annotation can be found in an article's abstract. An abstract can also contain facts that are irrelevant for annotation. My thesis supports the claim that it is only with respect to full articles that a system can tell relevant and irrelevant facts apart. It also implements an annotation system that enables a curator to analyse full-text papers effectively.

1.5.2 Evidence the claims are based on

The system I built was developed in order to support those claims. Even though the system only has a superficial understanding of natural language, it can nevertheless help a curator to find relevant facts quickly by providing the curator with highlighted sentences of interest in full text and a summary Table of all the different facts found ranked in order of importance. Using the analysis of repetition in full-text biomedical articles, the system can distinguish relevant and irrelevant facts as well as present the relevant ones from the most important to the least important.

1.5.3 Contributions made in supporting claims

I have also contributed to the field by developing annotated corpora (see Section 3.3), as well as gazetteers (see Section 3.2). Indeed, specialised resources are better at recognising NEs in specialised text rather than every day text. I have created such resources for several different kinds of entities (see Sections 3.4.1 and 6.1.4). Availability is discussed in Section 7.1.2.

1.6 Structure of the thesis

This section outlines what work has been carried out in order to support the claims made in the previous Section (1.5) and gives an overview of the structure of the thesis.

Chapter 2 introduces the background for this research work. Chapter 3 focuses on finding relevant information about protein locations in full-text papers. Chapter 4 presents all the different technical features supported by the NPD Curator System Interface. Chapter 5 introduces the system interface itself - built to test the claims - and evaluates it. Chapter 6 discusses extensibility and maintainability of my approach. Finally, chapter 7 gives a conclusion.

Chapter 2

Background

The expansion of biological databases is increasingly relying on semi-automatic text-mining of the biological literature. Indeed, there has been a growing volume of work in text-mining for biological literature, and biomedical text-mining is a rapidly growing field. Experts in biology have moved from small focussed studies to high-throughput assays. Thus, there are more data in any one paper in the form of new evidence for known facts, new facts, variations on and exceptions to known facts, and so on. To clarify my previous definition of “relevant sentences” in this thesis (see Section 1.4), “relevant sentences” refer to sentences that carry information about the subnuclear localisation of one or several protein(s) - whether that piece of information constitutes a known or new fact.

There are a few main streams in text-mining. The first section of this chapter introduces them and, by doing so, gives an overview of what text-mining covers. The second section provides background information on the various methods and tools used in this thesis. The next two sections compare previous research work to the work I have achieved and I am presenting in this thesis. Section 2.3 compares my system to other systems that detect relevance in free biomedical text. Section 2.4 compares my system to other systems that detect redundancy in free biomedical text. Section 2.5 presents how other biological databases are annotated. Finally, the last section offers a conclusion for this background chapter.

2.1 Overview of text-mining

This first section is a background section that defines text-mining, its different domains and what metrics it uses to evaluate results. Text-mining is a discipline that involves

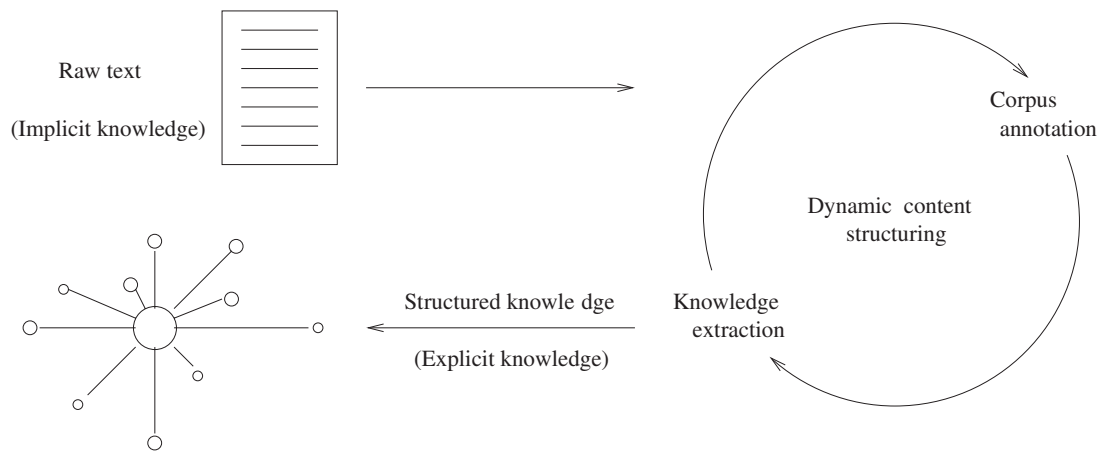


Figure 2.1: From implicit to explicit knowledge. Raw and implicit knowledge contained in free text can be extracted using text-mining methods, and formatted into structured and explicit knowledge.

looking for information of interest in free text and converting it into structured data where the information will be easily searchable. The goal behind this task is for the data to become readily available and gathered together with connected information, such as links to other databases for example. Figure 2.1 illustrates this process.

The text-mining tasks undertaken in this thesis are Information Retrieval (IR) both at the document level (this is illustrated in Figure 2.2, which essentially fits within Figure 2.1) and at the sentence level, NER and Information Extraction (IE, Figure 2.3). While a lot of research in the biomedical text-mining is focussed on extracting protein interactions [CBLJ04, AGH⁺08], for my thesis - as described in Section 1.4, I worked on protein-compartment relations and, more specifically, on the localisation of a nuclear protein in a subnuclear compartment at a given time of the cell cycle.

2.1.1 IR and TC

Information Retrieval (IR) involves trying to find all the items relevant to a specific query. The items can vary in nature. They can be text documents (document retrieval), sentences (sentence retrieval) or groups of subsequent sentences (passage retrieval). These items can even be photographs or films.

There are two standard approaches to IR. The first one consists of the user specifying a combination of keywords, this is called a Boolean query. For the second type, instead of providing one or more keywords, the user gives a “query document” that

will be used to retrieve documents considered similar.

Text categorisation (TC) is a task that automatically classifies text documents into categories by looking at the content they hold. Figure 2.2 shows the TC task that was achieved for my MRes ([Can04], see Section 4.1). TC divides up a given set into ones belonging to different categories whereas IR returns a subset of a very large given set that belong to a specified category.

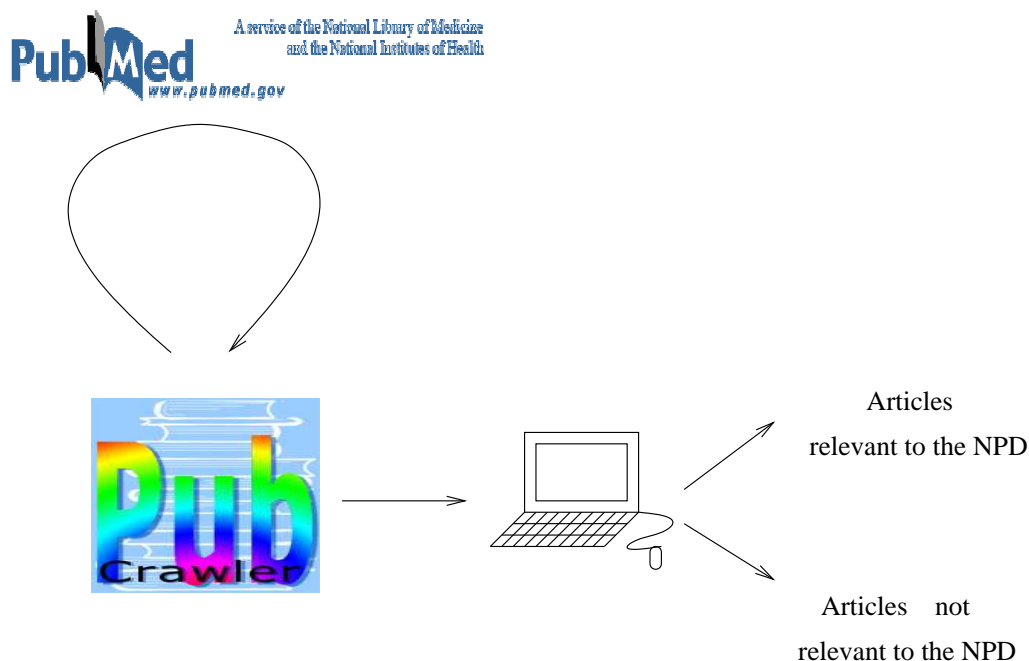


Figure 2.2: Text categorisation tool taking PubCrawler's alerts as an input and categorising PubMed abstracts in 2 categories - relevant or not relevant to the NPD.

2.1.2 IE and NER

Information Extraction (IE) involves extracting specific information of interest from free text. It makes use of NER, which aims to identify items in the text comprising one or more subsequent words and classifying them into chosen NEs. This can be achieved using a dictionary approach, a rule-based approach or, should there be an annotated corpus available, ML can be used. In the newswire domain, names of organisations, cities and people are of interest, while in the biomedical domain, the names of proteins, cell compartments, phases of the cell cycle, take the focus of NER (see Section 3.4.1). For example, as Figure 2.3 shows, it is possible to extract from a collection of documents or some Webpage content, some entities (*e.g.*, a protein name) and some

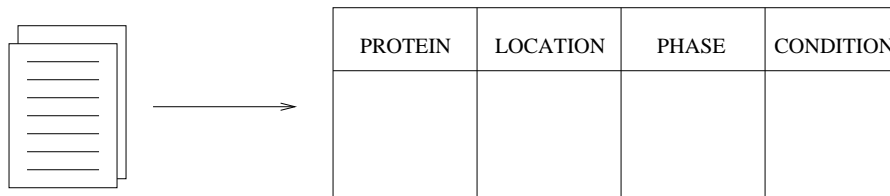


Figure 2.3: Example of IE in the biomedical domain. Information can be extracted from relevant free text and put into a structured format.

relationships (*e.g.*, the subnuclear localisation of this protein).

The names of genes and proteins can take several forms. Each of them can usually be referred to as full names (*e.g.*, nucleoporin 153) or symbols (*e.g.*, NUP153). For each possible symbol, topographical variants also exist (*e.g.*, NUP-153, NUP 153). Some genes and proteins have synonyms (*e.g.*, Nuclear pore complex protein), others have ambiguous names that could have a different meaning in a different context (*e.g.*, FRAP is a protein name in the NPD, however it is also the name of a technique and stands for “Fluorescence Recovery After Photobleaching”).

IE does not only consist of extracting NEs, it is also about extracting relations. This can be achieved using co-occurrence, patterns or fuller parsing. Co-occurrence statistics (see [RSKA⁺07] in Section 2.3.5) tend to provide high recall but low precision. Pattern-based methods (see [DA05a] in Section 2.3.3) retrieve results that offer a higher precision as words should fit in a precise template. These last two methods can also be combined. Fuller parsing methods pay attention to syntax and work with parse trees, dependency trees *etc.*

2.1.3 Evaluation measures

In order to evaluate and compare results obtained by different techniques, text-mining utilises common measures. Because it is not feasible to test systems against absolutely every single possibility, it is standard practice to perform evaluations using:

- an annotated corpus called a Gold Standard (GS),
- a measure that shows how well a system performs against such a GS.

The most reliable kind of GS is a corpus manually annotated by a domain expert. However, acquiring this type of collection is very expensive, which is the reason why it can sometimes be substituted by a surrogate GS. In a surrogate GS, the right answer(s)

are assumed to be the most frequent one(s), or the set of answers returned by participants for a contest. For example, the latter approach is used in TREC (see Section 2.4) and is referred to as “TREC pooling”. A surrogate GS, although not ideal, can nevertheless aid researchers to evaluate their experiments and advance the field. It can sometimes fail to help, in which case expert annotation is sought.

In the case of unassisted annotation, there can be multiple annotators [AGH⁺08], in which case interannotator agreement experiments are in order to check the consistency of the annotation within the corpus. When only one annotator is involved, one can look at their reliability across multiple articles or across the sentences within a single article (see Section 3.3.1). My GS is the result of one human annotator who has been the sole basis for deciding what information (articles and sentences) to extract in the NPD all along. A small amount of surrogate GS data was also generated, as explained in Section 3.3.1.

There are several standard measures to analyse the performance of systems. Two that are commonly used in IR, IE and NER are called *precision* and *recall*. They are calculated based on four counters (all usually displayed in a confusion matrix, see Figure 2.4):

- True Positives (*TP*): items *correctly* labelled as positive;
- False Positives (*FP*): items *incorrectly* labelled as positive;
- True Negatives (*TN*): items *correctly* labelled as negative;
- False Negatives (*FN*): items *incorrectly* labelled as negative;

where the total number of items is $N = TP + FP + TN + FN$.

		+ Guessed	-
+	TP	FN	
True			
-	FP	TN	

Figure 2.4: The confusion matrix is a Table showing the number of *TP*, *FP*, *TN* and *FN*.

Precision (P) represents the number of true positives with respect to the number of items the system identified as positive, whereas *Recall* (R) represents the number of true positives with respect to the total number of items that should have been identified as positive:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.2)$$

In order to produce a single measure of system capability, precision (P) and recall (R) can be combined together in a measure called F-measure or balanced *F-score*:

$$\text{F-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.3)$$

In this equation, recall and precision are given equal weights, hence the name balanced *F-score*. When one of precision or recall is more important than the other, one can vary the weighting of the two.

Another way to look at the data is to consider the True Positive Rate (TPR) or *sensitivity* and the True Negative Rate (TNR) or *specificity*, defined as follows:

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP + FN} = \text{TPR} \quad (2.4)$$

$$\text{Specificity} = \frac{TN}{TN + FP} = \text{TNR} \quad (2.5)$$

A common way to visualise the combination of the two is to create a Receiver Operating Characteristic (ROC) curve. The TPR or sensitivity is represented on the y axis while the False Positive Rate (FPR) or (1 - specificity) is represented on the x axis of the curve. In order to obtain a summary of performance based on this curve, the Area under the ROC curve (AROC, [Bra97]) is usually considered. The AROC corresponds to the probability that a classifier will position a positive instance higher than a negative one in rank-ordered list. An area of 0.90 to 1 represents an excellent result. 0.80 to 0.90 is good, 0.70 to 0.80 is fair while 0.60 to 0.70 is poor and anything below 0.60 would be a worthless result. The AROC is used as a measure in Section 2.4.1.

Recall and precision assume that all answers (hits) are equally good. If that is not the case, then one might want evaluation metrics that reflect the position of an answer in a rank-ordered list. Presenting users with a rank-ordered list makes it easier for them to check the critical cases, *i.e.* at the bottom of the list.

Precision@n and recall@n show precision and recall calculated at position n in the list of results. By computing these measures for different positions, it is possible to see whether relevant items have correctly been placed before non relevant ones. These metrics are therefore useful on showing the quality of the ranking [Can04].

$$\text{Precision@n} = \frac{TP@n}{n} \quad (2.6)$$

$$\text{Recall@n} = \frac{TP@n}{N} \quad (2.7)$$

where N is the total number of items retrieved.

The average precision can be calculated by computing an average value for all the precision@n for all ranks in the hitlist:

$$\text{Average Precision} = \frac{\sum_{r=1}^N (P@r * rel(r))}{\text{number of relevant documents}} \quad (2.8)$$

where r stands for rank and rel(r) is a Boolean function on the relevance of the given rank. This measure indicates whether the technique analysed returns more relevant documents early in the list. Average precision describes the performance of a single strategy with respect to a single query. Mean Average Precision (MAP) can be computed when there is more than one query. This measure corresponds to the mean value of the average precisions calculated for each of the queries. While the last two measures are not used to analyse my results, they are used by the TREC Genomics tracks that I cover in my literature review in Section 2.4.2.

Often, it is convenient to neglect the exact precision and recall scores and simply measure whether a system returns a relevant document. The metric a@n can be defined as there being at least one instance of the answer within the first n elements in the list. It is used in the field to calculate results when there is a single answer.

For my work in Section 3.4.4, I have introduced a new but related evaluation metric, which I call A@n. It can be used to calculate results when there are several distinct answers. It can show at what n (how far down the hitlist) all the different answers have been retrieved (no matter how many instances of each were retrieved). It is customary for a capital letter to stand for a set, and the small letter for an arbitrary member. My ‘‘A@n’’ then stands for at least one instance of each answer, while ‘‘a@n’’ stood for at least one instance of the answer. I define it as follows:

$$A@n = \frac{\text{number of answers retrieved @n}}{\text{number of answers}} \quad (2.9)$$

For example, if we wish to retrieve all the colours of the rainbow, we are actually looking for seven distinct answers (violet, indigo, blue, green, yellow, orange and red). If the first seven items retrieved all contain a different colour each then $A@1 = 1/7$, $A@2 = 2/7$, $A@3 = 3/7$, $A@4 = 4/7$, $A@5 = 5/7$, $A@6 = 6/7$, $A@7 = 7/7 = 1$ and any $A@n$ where $n > 7$ is equal to 1. However if each colour is explained over two consecutive items rather than a single one then $A@1 = 1/7$, $A@2 = 1/7$, $A@3 = 2/7$, $A@4 = 2/7$, $A@5 = 3/7$, $A@6 = 3/7$, $A@7 = 4/7$, $A@8 = 4/7$, $A@9 = 5/7$, $A@10 = 5/7$, $A@11 = 6/7$, $A@12 = 6/7$, $A@13$ and above are equal to 1.

It is interesting to see how far down the hitlist one needs to go to find all the distinct answers one is looking for. In our first example, we only need to go down to the 7th item in the list whereas, in our second example, we need to go down to the 13th. This metric allows us to focus on the number of distinct answers retrieved rather than the number of instances of answers retrieved. This is interesting when dealing with documents containing repetition within the items retrieved.

2.2 Background on general methods and tools

In this section, I describe general methods and tools used in the rest of the thesis. Some NLP parts will refer to Part-of-speech tagging and stems. WEKA is a tool I used to run ML algorithms such as Naive Bayes and Decision Tree. Maximum Entropy and Support Vector Machines are other ML methods I used through other tools.

2.2.1 Part-Of-Speech (POS) tagging

There are eight main parts of speech: noun, verb, pronoun, preposition, adverb, conjunction, adjective and article. Other parts of speech include numerals, determiners, particles (“up”, “down”). POS tagging involves assigning a single POS tag to each word (and even punctuation). Part of speech taggers are used to predict the behaviour of unseen words. Words are simply divided into categories of words that behave similarly. POS tagging then comes down to classifying words into their correct category. A POS tagger can study the neighbourhood of an unseen word. If it is surrounded by known words, it might be possible to draw conclusions from that. For example, we know it is likely for a noun to appear after an adjective or a possessive pronoun, and for a verb to appear after a pronoun.

2.2.2 Stem

A word stem is the part of a word that all variants have in common. Most of the time, stems are roots and these variants differ in the prefixes and suffixes they attach to the root. In this thesis work, words that contain a loca stem are of interest - *e.g.*, “location”, “located”, “localization”, “colocalized”, *etc.*

2.2.3 Stop words

Stop words are words that represent noise rather than meaningful data. It is common to filter them out before undertaking analyses in NLP. Appendix E gives a list of the stop words used in this thesis work.

2.2.4 TF.IDF

TF.IDF stands for “term frequency - inverse document frequency”. This measure corresponds to the importance of a term in a document with respect to a document collection. It is often used as a weight. The more times the term appears in the document, the higher the TF.IDF score is. However the score decreases the more the term occurs in the given corpus.

2.2.5 Machine Learning (ML)

Machine Learning (ML) is a field of Artificial Intelligence. ML techniques aim at making a computer “learn” and then predict answers based on data. The two main trends include supervised learning and unsupervised learning. With the first one, an algorithm benefits from training data and takes decisions based on what it has learnt by studying the training set. With the second one, there are no previously annotated examples so an algorithm must model the set of inputs and make decisions based on that model.

2.2.6 WEKA

WEKA [WF05] (<http://www.cs.waikato.ac.nz/ml/weka/>) stands for the Waikato Environment for Knowledge Analysis and is a data mining software written in Java. It holds a set of machine learning algorithms that can be applied to various kinds of data sets.

Typically data sets are presented in feature vectors. A feature vector is a vector of n dimensions of features. The features themselves can take various values defined by the user at the top of an .arff file, which is the format WEKA requires. WEKA offers tools for data pre-processing, classification, regression, clustering, association rules, and visualisation. In this thesis, I use the pre-processing and classification tools (NB and DT, see next two sections).

Typically, WEKA can perform an n -fold cross-validation on labeled data or an actual testing on a test set to obtain results. One fold of cross-validation involves separating the labeled data into a training set and a random testing set. It is common to use several folds so as to get averaged results. An actual testing on a test set requires training a classifier on the labeled data and testing it on a separate set (usually unlabeled).

2.2.7 Naive Bayes (NB)

The Bayes theorem is derived from conditional probability. Conditional probability says that

$$P(C|A) = \frac{P(A,C)}{P(A)} \quad (2.10)$$

This means that

$$P(A|C) = \frac{P(A,C)}{P(C)} \quad (2.11)$$

The Bayes theorem can be obtained by replacing $P(A,C)$ in equation 2.10 by its definition in equation 2.11:

$$P(C|A) = \frac{P(A|C)P(C)}{P(A)} \quad (2.12)$$

In Equation 2.12, C represents a hypothesis and $P(C)$ represents the prior probability of C . $P(C)$ is the probability that the hypothesis C is correct without knowing anything about the data contained in A yet.

If there is a class C we are trying to predict given a set of attributes A_1, \dots, A_n , we wish to find the value of C that gives the highest value of $P(C|A_1, \dots, A_n)$. According to Bayes theorem,

$$P(C|A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n|C)P(C)}{P(A_1, \dots, A_n)} \quad (2.13)$$

this means we wish to find the value of C that gives the highest value of $P(A_1, \dots, A_n|C)P(C)$.

An NB classifier is based on the Bayes theorem as well as a naive approach that assumes all attributes are completely independent of one another. This implies that

$$P(A_1, \dots, A_n | C_j) = P(A_1 | C_j) P(A_2 | C_j) \dots P(A_n | C_j) \quad (2.14)$$

Consequently, given a set of attributes A_1, \dots, A_n , the class C_j will be assigned if $P(C_j) \prod_{i=1}^N P(A_i | C_j)$ is maximal.

For example, say NB was trying to classify an unseen sentence such as “Fibrillarin is located in the nucleolus.” based on the following attributes: *p1* (one protein name), *c1* (one compartment name), *ph0* (no phase name), *loc1* (one loca stem, see Section 2.2.2). These features are presented in Section 3.4.1. NB would compute based on the training data (with R meaning Relevant and I, Irrelevant) both

- $P(R)P(p1|R)P(c1|R)P(ph0|R)P(loc1|R)$ and
- $P(I)P(p1|I)P(c1|I)P(ph0|I)P(loc1|I)$

where $P(R)$ and $P(I)$ represent the prior probability that the sentence is relevant and irrelevant respectively without knowing anything about the data the attributes contain yet. Should the first formulae reach a higher value than the second one, the sentence will be classified as relevant, irrelevant otherwise.

2.2.8 Decision Tree (DT)

WEKA performs a version of DT called “J48”, which is an implementation of the Quinlan algorithm [Qui93]. The algorithm uses the training data to create a decision tree, which is literally a tree with a decision taken at each branch. To classify unseen data from a test set, the algorithm will simply go down the tree following the attributes’ values and reach a final decision.

The tree is created in a simple and recursive manner. The first step is to identify the attribute that is the most discriminative within its values in the training data. This attribute is said to have the “highest information gain”. If, for this chosen attribute, a value always triggers the same classification, a termination is reached and a branch can be ended. Otherwise the algorithm goes back to the first step and identifies the next attribute with the highest information gain. It is then a combination of various attributes’ values that is considered to terminate a branch. If it is not possible to find a value that always triggers the same classification, the algorithm chooses the value that triggers the same classification in most cases.

For example, the tree generated by J48 for my cross-validation results in Section 3.7 is shown below. The following tree shows DT's branches and how they terminate, please refer to Chapter 3 for a detailed explanation of the attributes used in this tree. The format used is "name of attribute = value of attribute: y (yes) or n (no) (items correctly classified by this decision/ items incorrectly classified by this decision)".

```

rule_b = rb0
| rule_c = rc0
| | rule_a = ra0: n (2015.0/72.0)
| | rule_a = ra1
| | | loca_stem = loc0: n (141.0/49.0)
| | | loca_stem = loc1
| | | | percentage <= 20
| | | | | protein_keywd = pk0: y (4.0/1.0)
| | | | | protein_keywd = pk1: n (3.0)
| | | | | protein_keywd = pk2: n (0.0)
| | | | | protein_keywd = pk3: n (0.0)
| | | | | protein_keywd = pk4: n (0.0)
| | | | | protein_keywd = pkn: n (0.0)
| | | | | percentage > 20: y (24.0/1.0)
| | | | loca_stem = locn: y (3.0/1.0)
| rule_c = rc1
| | compartment_name = cn0: y (0.0)
| | compartment_name = cn1
| | | protein_name = pn0: y (0.0)
| | | protein_name = pn1: y (88.0/27.0)
| | | protein_name = pn2
| | | | phase = ph0
| | | | | protein_keywd = pk0
| | | | | | percentage <= 21.95122: n (5.0/1.0)
| | | | | | percentage > 21.95122: y (13.0/4.0)
| | | | | protein_keywd = pk1: n (11.0/2.0)
| | | | | protein_keywd = pk2: n (3.0)
| | | | | protein_keywd = pk3: n (0.0)
| | | | | protein_keywd = pk4: y (1.0)
| | | | | protein_keywd = pkn: y (1.0)
| | | | phase = ph1: y (4.0)
| | | | phase = phn: n (4.0)

```

```

| | | protein_name = pn3: y (11.0/3.0)
| | | protein_name = pn4: y (9.0/4.0)
| | | protein_name = pnn: n (3.0)
| | compartment_name = cn2
| | | protein_keywd = pk0: y (61.0/16.0)
| | | protein_keywd = pk1
| | | | compartment_adj = ca0
| | | | | percentage <= 25.925926: n (2.0)
| | | | | percentage > 25.925926: y (9.0/1.0)
| | | | compartment_adj = ca1: n (4.0/1.0)
| | | | compartment_adj = ca2: y (0.0)
| | | | compartment_adj = ca3: y (0.0)
| | | | compartment_adj = can: y (0.0)
| | | protein_keywd = pk2
| | | | protein_name = pn0: y (0.0)
| | | | protein_name = pn1
| | | | | wd_of_interest = int0: y (0.0)
| | | | | wd_of_interest = int1: n (2.0)
| | | | | wd_of_interest = intn: y (3.0)
| | | | protein_name = pn2: y (0.0)
| | | | protein_name = pn3: y (2.0)
| | | | protein_name = pn4: n (1.0)
| | | | protein_name = pnn: y (0.0)
| | | protein_keywd = pk3: n (1.0)
| | | protein_keywd = pk4: y (0.0)
| | | protein_keywd = pkn: y (0.0)
| | compartment_name = cn3: y (37.0/6.0)
| | compartment_name = cnn: y (9.0/1.0)
rule_b = rb1: y (164.0/18.0)

```

On the third line of the DT, we see that if the three Boolean rules were set to 0 (or false) then the sentence will automatically be classified as irrelevant according to this tree. The last line of the DT shows that a sentence with rule b set to 1 (or true) will be classified as relevant. The other lines represent more complicated cases with nested branches of ifs.

2.2.9 Maximum Entropy (MaxEnt)

The principle behind maximum entropy is to compute all the models which satisfy the constraints (features) of the training data and select the one with the maximum entropy, that is the model which preserves the most randomness, disorder or uncertainty, the model which does not add any extra constraint to the training data.

If the training data D provides a set of constraints on the conditional distribution $P(c|d)$ for document d and class c , [NLM99] explains the function of the document d and class c as $f_i(d, c)$ and:

$$\frac{1}{|D|} \sum_{d \in D} f_i(d, c(d)) = \sum_d P(d) \cdot \sum_c P(c|d) f_i(d, c) \quad (2.15)$$

As the document distribution $P(d)$ is not known, it is replaced by an approximation of it using the training data as follows:

$$\frac{1}{|D|} \sum_{d \in D} f_i(d, c(d)) = \frac{1}{|D|} \sum_{d \in D} \cdot \sum_c P(c|d) f_i(d, c) \quad (2.16)$$

2.2.10 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are a statistical method created by the field of ML. For each item we wish to classify (*e.g.*, sentence, document), a feature vector is gathered where the last position is occupied by the Boolean class assigned to this vector (+1 or -1). This class is given for the training set and generated by SVM for a test set. When the data is represented in space, it is as if a hyperplane separates both classes. The SVM training finds the hyperplane that maximises the distance between the hyperplane itself and the feature vectors closest to it. Automatically assigning classes to test vectors then involves looking up on which side of the hyperplane determined during training they find themselves located on.

2.3 Related work on tools detecting relevant information in abstracts and full-text articles

Although biomedical text-mining is a relatively new field, it has triggered sufficient interest among researchers that a few tools have emerged to help the work of biologists and database curators. For each tool, I will introduce the state of the art in automatic highlighting/extraction of relevant information. Furthermore, at the end of each section, I will conclude with a summary of the significant features of that particular tool and justify the work carried out in this thesis.

2.3.1 iHOP

iHOP [HV04] stands for Information Hyperlinked Over Proteins and retrieves sentences (from PubMed abstracts) related to a gene or protein name entered in a search box (<http://www.ihop-net.org/>). In each sentence various NEs are highlighted using a colour code for each NE category. The sentences in this first page represent interactions between the searched gene/protein and other NEs. A feature allows users to access a summary overview of the information provided for the first page. This summary Table of the interactions of the searched gene/protein displays four columns. The first column gives a gene or protein symbol while the second column gives its full name. The third column shows which organism this gene/protein belongs to. Finally, the fourth column provides the number of sentences that relate to this particular interaction (between the searched gene/protein and the one on this particular row).

In order to identify gene and protein names in free text, iHOP uses a dictionary approach. The dictionary is based in LocusLink [PM01] and UniProt [Con07] both extended with orthographical variations. It contains 534,000 original terms and 3 million terms derived from the original ones. iHOP then looks at sentences that contain zero, one or two genes and can fill a gene-verb-gene pattern. Performance of correct gene-publication associations from the LocusLink database was evaluated. The average recall was of 87% and the average precision of 94%. iHOP's Web services are still used at the present time, for example, in EcID [LEG⁺09] ("a database for the inference of functional interactions in *E. coli*") to extract possible interactions from articles.

Summary: The most significant feature of iHOP is that it provides a gateway to PubMed abstracts by navigating them from one gene/protein name (or hyperlink) to the next. The tool retrieves sentences based on gene-verb-gene patterns, it is general and not customised to any specific needs (for any specific database).

2.3.2 BioRAT

BioRAT stands for "Biological Research Assistant for Text-mining" and can be considered as a research assistant that is given a query and finds a set of papers, parses them and highlights the most relevant protein-protein interactions in each. [CBLJ04] shows the three components integrated into BioRAT:

- The *document search interface*, where the user can enter a query. BioRAT returns a list of titles from articles it retrieved by searching PubMed. The user can then choose to view the abstract or to download the full-length paper of any result from the list.
- The *template design interface*, where the user can view a document and choose specific target words or phrases to be identified by BioRAT in the text. The user may also define

its own templates in terms of POS, gazetteer headings or word stems. Lexicons can directly be edited through this same interface.

- The *results interface*, where BioRAT finally shows the protein-protein interactions it found in the free text.

In [CBLJ04], the authors state that BioRAT is able to retrieve more information from full-text articles rather than from abstracts alone. On average, less than half of the extracted information comes from the abstract when working on full-length publications. BioRAT achieves 20.31% recall and 50.07% precision on abstracts against 43.6% and 51.25% respectively on full-text papers.

Summary: The most significant feature of BioRAT is that it allows users to design their own templates and lexicons and to directly apply them on any text of interest. Moreover, the study presented in [CBLJ04] on abstract and full-text articles prompted me to work on full text and motivated my thesis hypothesis that it is only with respect to full-length publications that a text-mining system can tell relevant and irrelevant facts apart.

2.3.3 BioIE

BioIE is a Web application that can operate on MEDLINE abstracts if the user chooses to use the built-in PubMed IR facilities to retrieve text, or on any text that the user loads into the system (up to 200 Kb). It is an IE tool that the authors of [DA05a] define as “a rule-based sentence extraction system”. The tool offers five predefined categories of types of information related to proteins:

- structure,
- function,
- diseases and therapeutic compounds,
- localisation, and
- familial relationships.

The tool selects sentences based on predefined templates they may contain. The templates can be one or more words and represent ways in which the five predefined categories (see above) can be expressed. All the templates were handcrafted and based on keywords. They vary in complexity from pair of words (composed of two keywords, or a keyword and a preposition), to more sophisticated patterns (comprising keywords, prepositions, and allowing for a given number of words in between). BioIE stores different templates for each of these five categories. The templates used for the localisation category are displayed in Table 3.14. Section

3.5 compares my method with BioIE's. The section gives results obtained from performing BioIE on my data.

Summary: The most significant feature of BioIE is that pre-defined templates were manually created and customised to five distinct categories. My work concentrates on one of those categories - localisation.

2.3.4 METIS

METIS [MDK⁺05] is a Web-based annotation tool that integrates PRECIS [MRA03] - which produces protein family reports given a query sequence - and improves its results by adding relevant sentences to it found in the biomedical literature. It can be accessed through Minاتور (“MINing Online Text - A User-friendly Resource”, <http://www.bioinf.manchester.ac.uk/dbbrowser/minotaur/about.html>). METIS is composed of two sentence classification components. The first component consists of a set of three SVMs (see Section 2.2.10). They are each trained on different annotated corpora: one on protein structure, one on protein function and the last one is on disease. The second component is the BioIE tool presented in Section 2.3.3. The two components are not combined in METIS. The users can elect to use either the BioIE templates or the Support Vector Machines to perform sentence classification, but not both at the same time.

Topic	Precision		Recall	
	SVM	BioIE	SVM	BioIE
Structure	51	33	74	85
Function	31	16	61	91
Disease	48	56	56	79

Table 2.1: Sentence classification results obtained by METIS

As shown in Table 2.1, BioIE achieves better results than SVMs for the disease category and SVMs perform better than BioIE for the structure category. The authors suspect the disappointing results obtained for the function category is due to the fact that the terms used for this purpose are “*polysemic*” and “*not specific to descriptions of function alone*”.

Summary: The most significant feature of METIS is that it offers these two different approaches that have different strengths and weaknesses: The templates generally provide higher recall, whilst the SVMs generally give higher precision. My final tool combines two approaches to highlighting relevant sentences. The first one is based on a DT classifier and gives higher precision, while the second one relies on NE co-occurrences and offers higher recall. (see Sections 4.2 and 4.3).

2.3.5 EBIMed

EBIMed [RSKA⁺07, RSKA⁺06], like BioIE, is a Web application. It is similar to BioRAT in that it offers an IR service as well as IE. While BioRAT deals with protein-protein interactions in full-text articles, EBIMed provides an overview of all relations found between UniProtKB/Swiss-Prot [BBA⁺03] protein/gene names, GO annotations and Drugs and Species NEs in abstracts. EBIMed creates relations such as protein-protein interactions if two UniProtKB/Swiss-Prot protein names occur in the same sentence, drug-protein associations if a protein name and a drug name co-occur; protein function if a protein name and a GO term co-occur, it also relates a protein to a certain organism when a protein name and a species name co-occur. EBIMed does not use any sentence pattern recognition.

The resulting overview is rank-ordered and is displayed in a summary Table (see Figure 2.5), where links to sentences in the context of abstracts are available. This user interface is similar to mine (see Section 5.1.5), although mine offers links to sentences in the full text of papers.

EBIMed extracts protein names with 90% precision. As the authors state in [RSKA⁺06], this high precision is due to unambiguous protein names selected for this particular assessment. The latter also revealed that the protein pairs extracted were meaningful in 37% of the cases and in 50% of the cases for the drug protein pairs. While my NER does not do as well as EBIMed's, my tool extracts protein-localisation pairs (or localisation relations) at a higher rate as discussed in Section 3.4.4.

Summary: The most significant feature of EBIMed is that it offers an overview of co-occurrences of a large number of NEs in abstracts. Again, my work focusses on a certain type of NEs (related to localisation) in full text. iHOP, BioRAT, BioIE and EBIMed provide users with IR facilities based on PubMed. My work is driven by the requirements for the annotation of the NPD. Indeed, the IR facilities are customised to the need of day-to-day annotation (by accepting results from PubCrawler as input queries to the tool).

2.3.6 TXM project

[AGH⁺08] presents experiments conducted in order to evaluate the extent to which annotation can be accelerated when supported by assistance from NLP. The paper also reports on scores the curators gave to various aspects of the authors' annotation tool, which provided them with NLP hypotheses for protein-protein interactions (PPIs). The tool itself relies on ML for NER and relation extraction and uses several components explained in [AHG07, Nie06, HM07].

An annotated corpus, named the enriched protein-protein interaction (EPPI) corpus, was produced by a team of domain experts. It comprises 217 full-text papers containing experimental evidence of PPIs. The papers are retrieved in XML or HTML format and are then converted

2.3. Related work on tools detecting relevant information in abstracts and full-text articles 43

The screenshot shows the EBIMed web interface in a Mozilla Firefox browser window. The search term 'nucleolin' is entered in the search bar. The summary section displays '500 Medicine Abstracts' and a table of NE types. The HitPair table shows co-occurrences for 'Nucleolin' across various NE categories.

Type	Hits	HitPairs
Protein/Gene	684	6174
Cellular component	83	1365
Biological process	129	1858
Molecular function	39	565
Drug	63	499
Species	135	1649
Total	1133	12080

Protein/Gene	Protein/Gene	Cellular component	Biological process	Molecular function	Drug	Species
Nucleolin <small>(score: 2392)</small>	B23 (54/96)	nucleolus (74/95)	transcription (53/81)	binding (124/182)	gel (13/15)	human or man (67/91)
	phosphoprotein (35/35)	cell surface (34/64)	localization (47/73)	RNA-binding (34/47)	laser (11/13)	yeast or Saccharomyces cerevisiae or S. cerevisiae (23/29)
	nucleophosmin or NPM (31/46)	nucleus (33/38)	phosphorylation (37/74)	casein kinase II (10/20)	via (6/6)	mouse (21/32)
	RNA binding protein (29/27)	cytoplasm (29/37)	ribosome biogenesis (34/35)	DNA binding (6/7)	stress (5/7)	glycine (20/23)
	topoisomerase or topoisomerases (20/27)	ribosomes (20/21)	apoptosis (23/32)	mRNA-binding (6/6)	maps (3/3)	rat or Rattus norvegicus (19/26)
	UBF (15/22)	chromatin (18/21)	cell cycle or cell division cycle (22/34)	nucleic acid binding (5/9)	heparin (3/3)	cancer (14/30)
	a protein (15/15)	nucleoplasm (15/15)	cell proliferation (21/27)	kinase activity (2/2)	dexamethasone (2/6)	HIV (12/28)
	cdc-2 (14/18)	intracellular (12/14)	rRNA processing (19/21)	protein binding (2/2)	taxol- (2/3)	Xenopus laevis or X. laevis (10/11)
	helicase or helicases (12/15)	membrane (9/11)	mitosis (18/25)	rRNA binding (2/2)	progesterone or medroxyprogesterone acetate (2/2)	Xenopus (7/11)
	NSR1 or nuclear localization sequence-binding protein (11/15)	hnRNP (9/11)	ribosome assembly (14/14)	DNA-dependent ATPase activities (2/2)	adenosine (2/2)	Chinese hamster (7/8)
	protein kinase or protein kinases (10/13)	cell membrane or plasma membrane (7/10)	interphase (12/13)	ATP binding (1/2)	luminal (2/2)	Escherichia coli (7/7)
	bcl-2 (8/22)	dense fibrillar component (7/9)	development (11/13)	E2 (1/1)	cascades or L2 (2/2)	onion or Allium cepa (6/8)
	NRE or NREs (8/20)	chromosomes (6/10)	Translation (10/18)	tRNA or transfer RNA (1/1)	spectrum (2/2)	bovine (6/8)
	RBD or	fibrillar center or fibrillar centres (5/8)	DNA replication (9/15)	ATPase activities (1/1)	label or trigger (2/2)	leaf or Sugar (2/2)
		granular components (5/5)	rRNA transcription (8/8)	chromatin-binding (1/1)	cap (2/2)	
			S phase or S phases (7/8)	ligase activity (1/1)		
			embryogenesis (5/6)	single-stranded-DNA-binding (1/1)		
			biosynthesis (5/5)	mRNA 3'-UTR binding (1/1)		

Figure 2.5: EBIMed screenshot (<http://www.ebi.ac.uk/Rebholz-srv/ebimed/processing.jsp?queryId=QueryDMYHMSms04052008141346906>) The summary section shows the first 500 abstracts retrieved were analysed and for each type of NE a Table displays the number of hits (the number of entities of that particular type found in the 500 abstracts) as well as the number of hitPairs (the number of permutations possible when these entities are shown along with the other NEs they co-occur with). The HitPair Table shows the co-occurrences found for nucleolin. Each column represents a category of NEs. For each co-occurrence, EBIMed gives in brackets the number of and links to abstracts and sentences the information was found in.

to an internal XML format. The annotation was of:

- nine types of entities (Complex, CellLine, DrugCompound, ExperimentalMethod, Fusion, Fragment, Modification, Mutant, and Protein),
- PPI relations,
- FRAG relations (which link Fragments or Mutants to their parent proteins).

One experiment is particularly relevant to the current work. In this experiment, they used 4 curators who annotated 4 papers in 3 different conditions:

- Unassisted: without assistance
- GSA-assisted: with integrated gold standard annotations
- NLP-assisted: with integrated NLP pipeline output

The authors of [AGH⁺08] further explain that each curator annotated an article only once with either no assistance at all, GSA assistance or NLP assistance. However the curators were not told under which condition they were working. Unassisted annotation gave the fewest records (121) for all four publications. The curators annotated another 20 records (+ 16.5%) when working with NLP assistance while they managed an extra 49 records (+ 40.5%) with GSA assistance. This shows that providing NLP output helps curators to find more information.

The average length of time for annotating a record is inevitably the highest in the unassisted annotation condition. NLP-assisted annotation is 22% faster, GSA-assisted annotation 34% faster. Offering assistance to curators enables them to perform their work in less time, indeed a third less time in the optimal condition, which is GSA-assisted annotation.

After annotating each paper, curators had to complete a questionnaire. They gave GSA assistance a slightly better score than NLP assistance and although the experiments results prove otherwise, the questionnaire revealed the curators were not sure if either GSA or NLP assistance accelerated their work.

Summary: The most significant feature of the TXM project is that it offers a very interesting user-based study of the evaluation of the tool provided. The study demonstrates that NLP assistance helps curators in their annotation work.

2.3.7 PolySearch

PolySearch [CKY⁺08] is a Web-based tool, which sets out to retrieve rank-ordered sentences related to queries of type “Given X, find all associated Y” (*e.g.*, given a disease, find all associated genes). Once users have selected their X and Y, they are asked to specify search terms for their X. Advanced search options are also available before submitting the query. One of the

advanced option is to select which databases to search for results. Indeed, PolySearch is able to look through PubMed as well as other databases such as OMIM [OMI], Swiss-Prot [BBA⁺03], DrugBank [WKG⁺06], the Human Metabolome Database (HMDB) [WTK⁺07], the Human Protein Reference Database (HPRD) [MSK⁺06] and the Human Genome Mutation Database (HGMD) [SMB⁺09]. It displays results in a similar way to EBIMed. After submitting the query, PolySearch provides users with status updates until the results are finally displayed. PolySearch also offers facilities to analyse and study variations in DNA sequences.

The text-mining system of this tool uses a dictionary approach based on 9 lexicons:

- lexicons 1 and 2 - genes/proteins
Sources (manually edited): Swiss-Prot, Entrez Gene [Ent], Human Genome Organisation Gene Nomenclature Committee [WLDAP02] and HPRD.
- lexicon 3 - diseases
Sources (manually edited): Unified Medical Language System (UMLS) [HL93] and OMIM.
- lexicon 4 - drugs
Source: DrugBank.
- lexicon 5 - metabolites
Source: HMDB.
- lexicon 6 - pathways
Sources (manually edited): Kyoto Encyclopedia of Genes and Genomes (KEGG) [OGS⁺99] and BioCarta (<http://www.biocarta.com/>).
- lexicon 7 and 8 - tissues and organs
Source (manually edited): Swiss-Prot.
- lexicon 9 - Subcellular localisations
Source: HPRD.

Like iHOP and EBIMed, PolySearch ranks the associations retrieved. The order of relevance is calculated based on whether a database term, a query term, an association word (equivalent to my ‘words of interest’ - see Section 3.2) and a pattern were found. A sentence ranks highest when all four are detected. The rules of their pattern recognition system mainly focus on the number of words allowed between words identified as relevant in sentences. PolySearch identifies gene and protein names with slightly better precision (0.9), recall (0.85) and F-score (0.87) than iHOP. Their rule-based patterns and extensive evaluation is further explained at <http://wishart.biology.ualberta.ca/polysearch/cgi-bin/help.cgi>.

PolySearch seems to combine strengths from both iHOP (pattern recognition) and EBIMed (flexible search whereas iHOP supports searching for genes only). Moreover, it also supports IE from databases other than PubMed. Traditionally, the NPD curator only relied on PubMed articles for IE (see Figure 1.10). Like EBIMed, my final tool does not contain any pattern recognition system. Its ranking is based on the number of instances found for each protein localisation association (see Section 3.7 for a discussion).

Summary: Although it is interesting to see how iHOP, EBIMed and PolySearch retrieve and rank sentences, they are general tools that are not customised to the needs of any database annotation. Benefiting from a specific annotation task and an annotated corpus, my final tool uses a supervised ML approach (see Section 3.4) to find relevant sentences. The most significant feature of PolySearch is that it is capable of searching through several databases, including PubMed. Its numerous lexicons allow for query synonym expansion. Moreover, users can define their own “association” words to be recognised in patterns.

2.3.8 FACTA

FACTA [TTA08] is similar to EBIMed and PolySearch in its results display. It stands for “Finding Associated Concepts with Text Analysis”, takes keyword(s) as an input and uses co-occurrence statistics to produce results. The categories of NEs covered are: human gene/proteins, diseases, symptoms, drugs, enzymes and chemical compounds. Its main difference with EBIMed is that it works with concepts as well as words. FACTA’s index of concepts uses UniProt accession numbers [Con07], HMDB [WTK⁺07], KEGG [OGS⁺99] and DrugBank [WKG⁺06], while its index of words uses the BioThesaurus [LHZW06] for gene/protein names and aliases and the UMLS [HL93] for names of diseases and symptoms. Its main difference with PolySearch is that, using pre-indexes, this text search engine allows users to submit Boolean queries composed of concept identifiers and keywords and to be presented with their results instantly.

Summary: Whilst querying is not the main priority for the NPD annotation process, FACTA certainly outperforms recent tools in terms of real-time response by storing and pre-indexing data (both indexes mentioned above as well as sentences from MEDLINE abstracts) in memory.

2.4 Previous work reported in open evaluation contests

Many open evaluation contests have been organised over the past few years to allow different research groups to compare their techniques against each other by testing them on the same data and tasks in biomedical text-mining. Such challenges are important for the field to progress.

The Text REtrieval Conference (TREC [Har93]), sponsored by the National Institutes of Standards and Technology (NIST), organised a lot of these contests for a variety of areas ranging from the video track to the cross-language track. TREC was founded in 1992 and offers several tracks each year. For each track, different data sets and tasks are set. Evaluation is performed by NIST judges.

This section will start with a presentation of the Knowledge Discovery and Data mining (KDD) Challenge Cup, which took place in 2003. I will then report on some TREC Genomics tracks' tasks that were relevant to my work between the track's first year, 2003, to 2005. (From 2006 onwards, TREC Genomics tracks concentrated on retrieving passages rather than sentences.)

Even though the Question Answering track in TREC 2006 was targeted towards the newswire rather than biomedical domain, I will present an interesting paper [SZK⁺06] that describes two methods implemented to deal with duplicate removal (see Section 2.4.3).

Indeed, in this thesis work, I looked at intra-document novelty and, more precisely, grouping sentences containing the same type of information together. To this extent, it involved duplicate clustering rather than catching the first sentence of a document that manifests a certain piece of information. Section 4.3 presents how sentences were tagged with labels according to what localisation relations were found in them, and were then grouped based on the labels they had in common.

TREC also had tracks on Novelty detection in 2002, 2003 and 2004. They were specific to the newswire domain and focussed on finding the first sentence of a document that manifested a piece of information. Nevertheless, some of the submitted runs used approaches relevant to this thesis work, which I will present in subsection 2.4.4.

2.4.1 KDD Challenge Cup 2003

Until the KDD Challenge Cup [YHM03] in 2003, there had been no open evaluation of techniques and systems in the field of biomedical text-mining. This contest provided participants with data from the FlyBase database [Con03], which gathers information on the *Drosophila* (fruit fly) genome. The training data were composed of 862 journal articles referenced in FlyBase and contained a total of 283 positives. The test set comprised 213 papers. Eighteen groups built systems that categorised articles for IR and IE. Contestants were given full-text papers and had to return a rank-ordered list of papers from the most likely to the least likely to contain extractable information (that is, experimental evidence about products, such as mRNA transcripts and proteins/polypeptides, associated with a specific gene), along with a yes/no answer as to whether they considered the article as relevant for IR/IE. The contestants were also given a list of genes that were discussed in each of those articles, for which they had to provide a yes/no answer as to whether they had found experimental evidence in the corresponding paper for the

related gene products.

The winning team [RFLF02] used an approach similar to that of BioIE, in that they manually created rules and patterns considered of interest. Their score on the rank-ordering was not reported, but they obtained 0.78 F-score on the yes/no relevance decision and 0.67 F-score on the yes/no experimental evidence decision. The next team [SEM⁺02] handcrafted a set of keywords and, for each paragraph of text, computed the distance between a gene name and a keyword from this set. They store the minimum distance for each <gene, keyword> pair, as well as the number of occurrences with that minimum distance, and use NB to obtain results. They achieved 0.81 AROC (see Section 2.1.3) on the ranking and 0.73 F-score on the yes/no relevance decision. F-score on the second task was not reported. Finally, a team from Imperial College and a company called Inforsense [GGLZ02] performed ranking with an AROC of 0.84, 0.58 F-score for the yes/no relevance decision and 0.59 for the yes/no experimental evidence decision. They automatically extracted 335 regular expressions patterns from the training corpus and applied those regular expressions to sentences in the test set containing a gene name. The final decisions were made by an SVM (see Section 2.2.10) using the patterns found as features.

Although the tasks set were only a small part of the FlyBase annotation process, this challenge still represented a valuable first contribution towards automating parts of this annotation workflow, and progress in biomedical text-mining in general.

2.4.2 TREC Genomics tracks

The first task of TREC Genomics tracks in 2003 [HB03], 2004 [HBR⁺04] and 2005 [HCY⁺05] was always an *ad hoc* retrieval task that required a document collection and topics for retrieval. For the first year, the collection consisted of a year of MEDLINE records (525,938 documents) and the topics were 50 gene names taken from the NLM's (National Library of Medicine) Gene Reference Into Function (GeneRIF) resource. For each gene name, the contestants had to find all MEDLINE references which concentrated on its structure, genetics or function in normal or disease states.

The second year (2004), the collection was much larger, as it represented 10 years of MEDLINE records (4,591,008 documents). The topics were also more sophisticated as they were the results of interviews with biologists discussing their real information needs. In 2005, the document collection was the same as in 2004. However, the track's organisers studied the previous year's topics and created a set of 5 generic topic templates (GTTs). For each GTT, ten information needs were chosen by surveying biologists, making up to 50 topics again.

Results were evaluated based on the MAP. As noted in section 2.1.3, average precision describes the performance of a single strategy with respect to a single query, while MAP can be computed when there is more than one query: it is the mean value of the average precision

for each query.

In 2003, 25 groups participated and submitted 49 runs. The best two teams were NLM-based and achieved MAPs of 0.4165 and 0.3994. They operated NER in non-text fields such as MeSH (Medical Subject Headings) and RN (Chemical Abstracts Service Registry Number) and identified species using MeSH. They also used very general keywords, such as “genetics” and “sequence”. The next team (UC Berkeley, MAP of 0.3912) utilised an ML approach based on gene name occurrence rules for document classification as well as document ranking.

In 2004, 27 groups submitted 47 runs. Patolis Corp. obtained the best results (MAP of 0.4075). They used LocusLink [PM01] to expand symbols, Porter stemming [RRP80] and Okapi weighting [RJ88]. The University of Waterloo was the next winning team that year. While in 2003 they used NER in RN fields and query expansion based on Okapi weighting as well as gene name bigrams, in 2004 they also tried pseudo-relevance feedback and general domain-specific query expansion that made use of lexical variants, acronyms, gene and protein name synonyms.

In 2005, 32 groups submitted 58 runs. The winning team (York University [XML05]) implemented two new query expansion algorithms in order to deal with acronyms, homonyms and synonyms, which represent important problems in biomedical IR. Their first algorithm was run for their automatic submission and made use of “break-points” (positions where a string can be separated by a space) and “replacements” (substrings within strings that can be substituted with another character(s), *e.g.*, alpha by a or 2 by ii).

Their second algorithm was run for their manual submission and made use of two databases (AcroMed [PCC⁺01] and LocusLink [PM01]) to produce two lists of variants of a gene name. After the merge, a domain expert was asked to manually correct the resulting list. Their automatic run obtained a MAP of 0.2888 and their manual one 0.3020.

In 2003, the second task of the TREC Genomics track [HB03] was an “exploratory” IE task, which consisted of extracting the GeneRIF statement from the MEDLINE record or full-text article, without any training data. The testing data were composed of 139 GeneRIFs, for which the organisers managed to obtain full-text access from “Highwire”. The evaluation measures were derivatives of the Dice coefficient, which calculates the overlap between two strings (in this case, a proposed GeneRIF and a real one) as follows:

$$\text{Dice}(a, b) = \frac{2 * \text{bigram overlap}}{\text{bigrams in } a + \text{bigrams in } b} \quad (2.17)$$

For example, in order to calculate the Dice coefficient of the words “cold” and “cool” we look at the bigrams in each word: {co, ol, ld} and {co, oo, ol}. Each set contains three elements and they have two elements in common: co and ol. Therefore,

$$\text{Dice}(\text{cold}, \text{cool}) = \frac{2 * 2}{3 + 3} = \frac{2}{3} \quad (2.18)$$

The three derivatives overcame the limitations of the Dice coefficient (*e.g.*, not taking into account stop words, the order of the words, *etc.*). The four evaluation measures were the following:

1. The *classic Dice* coefficient measures the overlap between 2 strings.
2. The *unigram Dice* gave additional weight to terms that were present several times in both strings.
3. The *bigram Dice* gave extra weight if words occurred in the same order, as it was measured based on bigrams rather than unigrams.
4. The *phrase Dice* was introduced to allow capture of bigrams that were separated by stop words.

Fourteen groups participated and submitted a total of 24 runs. The winning team was a group from Erasmus University, which obtained 57.83 classic, 59.63 unigram, 46.75 bigram, 49.11 phrase Dice scores. They utilised classifiers to rank-order sentences from the most likely to contain the GeneRIF statement to the least likely. For example, in [BNSH03], the ranking was performed using TF.IDF (see Section 2.2.4) weights on non-Boolean features. Participants realised that, in most cases, the GeneRIF statement was taken from sentences in the title, sometimes the abstract, rarely the rest of the full text. A baseline was then computed using titles only which resulted in the following Dice coefficients: 50.47 classic, 52.60 unigram, 34.82 bigram and 37.91 phrase. Only a handful of participating teams actually outperformed this baseline.

2.4.3 TREC 2006 QA track

In order to detect intra-document novelty (see Section 4.3), it is important to identify all the sentences in the document that refer to the same type of information, *i.e.* duplicates. In 2006, for the TREC Question Answering track, [SZK⁺06] describes two methods the authors implemented to deal with duplicate removal.

The first method uses the BLEU score. BLEU stands for BiLingual Evaluation Understudy and is a metric developed by IBM to evaluate the performance of Machine Translation (MT offers automatic text translation from one language to another) techniques. The authors consider a duplicate sentence as a paraphrase of the original sentence and explain that given BLEU was created to measure redundancy between sentences, a high BLEU score means the two sentences are very similar. Their approach involves calculating BLEU scores for all possible pair of sentences and storing the results in a matrix. They then choose a threshold and declare any pair of sentences that obtained a score above it a duplicate of one another.

The second method the authors present uses word-level edit distance. They used a clustering algorithm in order to group similar sentences, where the distance between sentences is

defined by the edit-distance metric. Then they tried to identify the longest sentence from each group of duplicates.

The BLEU metric approach outperformed the edit-distance metric method in the authors' comparative study where four different questions were asked and a human standard was set as to which answers ought to be removed because they were duplicates. The BLEU metric approach's best F-score was of 0.98 against 0.93 for the edit-distance metric. Furthermore, the BLEU metric approach's worst F-score was of 0.95 against 0.77 for the edit-distance metric. The authors explain that the reordering of words in duplicate sentences affects the edit-distance whereas it does not influence BLEU in any way.

2.4.4 TREC Novelty tracks

The first TREC Novelty track [Har02] took place in 2002 and was used as a trial run. The data created that year were of poor quality and results were low, with F-scores around 0.2.

On the other hand, the corpora developed for TREC Novelty tracks 2003 and 2004 represent valuable data sets for the research field of Novelty detection in the newswire domain. Most groups identified novel sentences by assessing their dissimilarity to past sentences. Term expansion was a popular approach amongst participants, which they used in order to enhance their sentence similarity detection.

In 2003 [SH03], the winning team was Tsinghua University who obtained an F-score of 0.5. They used sentence clustering and also tried calculating sentence redundancy by measuring unsymmetrical sentence overlap. They utilised a supervised redundancy threshold learning and developed a new tool for their experiments called TMiner.

In 2004 [Sob04], Tsinghua University used an approach based on cosine similarity computed between sentences after Principal Component Analysis (PCA), which is a technique similar to Singular Value Decomposition (SVD, see Section 3.6). They were again amongst the top runs but did not win.

The best team that year was the University of Iowa [EZB⁺04], with an F-score of 0.8. They chose a "new entity threshold" and computed for each sentence the number of NEs and noun phrases previously unseen. They then classified a sentence as novel if this number exceeded their threshold.

2.5 Related work on machine-assisted database maintenance

This section reports on how other biological databases are annotated, focussing on databases that involve annotation information from the literature for all or part of their data, with the rest

provided directly from experiments.

2.5.1 BRENDA

BRENDA stands for BRAunschweig ENzyme DAtabase. It houses molecular and metabolic data on 83,000 enzymes from 9,800 different organisms. Information contained in the database was extracted from the literature by domain experts, and represents a valuable resource to researchers in the domain of biochemistry and medicine. The url for BRENDA is <http://www.brenda-enzymes.info/>.

BRENDA is structured by EC (Enzyme Commission) number. For each EC entry, the database contains information on a recommended name, names of the different enzymes for different organisms, a list of substrates and products for the catalysed reaction, a list of inhibitors, a list of activating compounds as well as a list of cofactors, all referenced to PMIDs where the information was found in the literature. The Website states the database holds over 500,000 enzyme-ligand relationships, over 46,000 chemical compounds operating as ligands and 34,500 structures of ligands issued from over 56,000 papers.

In [HS05], the authors describe a method to automatically annotate enzyme classes with disease-related information extracted from the biomedical literature for inclusion in the database. The method uses a dictionary approach to recognise enzymes from the BRENDA database in abstracts. The dictionary contains six names per enzyme on average and concepts from the UMLS are identified using the MetaMap program [Aro01]. MetaMap parses free text and, for each noun phrase it finds, generates variants based on acronyms, abbreviations, synonyms and spelling variations. The automatic annotation is then based on the co-occurrence of those UMLS concepts and the enzyme names found in each sentence of abstracts. An SVM classifier was trained with a 1500 annotated sentences of which 18.2% involved one or more relations between an enzyme and a UMLS disease. The method was assessed. If an enzyme name and a disease-related term were both identified in a sentence, the co-occurrence would incur a relation between the two concepts. Out of the 430 relations annotated by a domain expert, 84.8% were retrieved correctly with 82.1% precision.

2.5.2 FlyBase

The FlyBase database [Con03] houses information on the *Drosophila* (fruit fly) genome and related species. It incorporates data extracted from the literature with data of different provenances such as other genome projects. The FlyBase annotation is supported by domain experts manually extracting information of interest from full-text articles coming from 35 journals.

In [KLS⁺07], the authors address the FlyBase curators' real needs and present a generic tool developed to assist them. An interface called PaperBrowser was developed, which shows

curators the full-text article with gene names highlighted. The user can have access to an ordered list of gene names as they appear in the text (using the PaperView navigation panel), as well as a list of all the groups of words that were identified as related to the same gene name (through the EntitiesView navigation panel). Both navigation panels can redirect the user to the corresponding paragraph in the text window where a particular gene name or related entity can be found highlighted in context. The tool is used almost daily by FlyBase curators as they find working from PaperBrowser easier than looking at a PDF viewer on screen or a printed out version of the article.

There are three main steps to their curator assistance tool [KSL⁺08]:

- the NER step is performed using Conditional Random Fields (CRFs) [Vla07] and achieves 61.4% recall, 89.2% precision and 72.7% F-score
- the sentence parser [BCW06]
- the anaphora resolution component [Gas06] performs with 53.4% recall, 63.0% precision and 57.8% F-score (the authors did not evaluate the impact of the anaphora resolution module on the annotation process separately)

The PaperView navigation panel uses the results of the NER step whereas the EntitiesView navigation panel makes use of the anaphora resolution results on top of the NER ones. Indeed, the combination of the two allow the tool to create the list of related entities previously mentioned.

2.5.3 Protein fingerprint database (PRINTS)

PRINTS [Att02] is a protein fingerprint database. A protein fingerprint can be defined as “a collection of aligned, unweighted sequence motifs”. PRINTS is annotated by analysing results from the fingerprinting method. To start with, a Multiple Sequence Alignment (MSA) is performed. Conserved motifs are then extracted for iterative scanning of a Swiss-Prot [BBA⁺03] and TrEMBL [BA96] composite in order to identify further family members. The iteration stops when no more family members can be found. The fingerprint is then ready to be annotated.

PRINTS provides for each fingerprint detailed and handcrafted annotation. In order to help PRINTS annotation, a system was developed called BioIE [DA05a] which I previously presented in Section 2.3.3. BioIE was tested to extract pertinent sentences from the literature and to match them to the annotation statements [DA05b]. On average, 50% of the statements were matched. Out of the unmatched annotations, 85.2% were actually not available in the original text. Indeed, PRINTS annotators had interpreted and summarised some relevant statements they had found in the literature. Moreover, they also added statements based on their own

knowledge so that the annotations would make sense as stand-alone comments. 3.6% of the unmatched annotations were down to missing patterns in BioIE (such as variants of unit terms) which were consequently added to the tool. The last 11.2% were due to idiosyncrasies of biomedical text. In [DA05b], the authors conclude that the main problem was actually getting hold of the correct pieces of text automatically.

2.6 Summary

Some biological databases (see Section 2.5) have used biomedical text-mining in order to reduce annotation time. In [AGH⁺08], the authors offer user-oriented experiments and show text-mining can indeed assist in annotation.

Like most systems presented in Section 2.3, my tool (see Section 1.4, Chapters 4 and 5) provides the user with both IR and IE. It displays all the facts found in a summary Table, and highlighted full text is available by clicking on items in the Table in the same style as EBIMed, PolySearch and FACTA. What this thesis addresses that has not been done so far is providing customised IR, IE and novelty with regard to a particular database's content - in this case the NPD. The tool developed in this work allows the curator to make supported decisions all the way through the annotation process and is yet extensible to other databases (see Section 6.1).

Chapter 3

Retrieval of relevant sentences in full text biomedical papers

The question this chapter addresses is, given papers of interest, how best to retrieve relevant sentences from full text of biomedical articles? In my final tool, documents of interest are retrieved using a combined classifier developed during my MRes [Can04] presented in Section 4.1.

Different techniques can be used to retrieve sentences from a document. In this chapter, I present a supervised ML method (see Section 3.4) I developed during my PhD and compare its performance to two other types of method on this task:

- rule-based with BioIE (see Section 3.5),
- and unsupervised ML with Infomap (see Section 3.6).

Unsupervised ML claims advantages of portability and less work on the part of the expert, as they do not have to manually classify every data element in a possibly large training set. BioIE (see section 2.3 and [DA05a]) has the advantage of being an existing system with already specified templates for the localisation domain. This chapter first focuses on presenting my supervised method before introducing the other two and comparing them.

3.1 Text pre-processing

Full text of journal articles can be retrieved from the Internet, using journal Websites, as HTML files. They can then be converted from raw text to “clean” text that is ready to be processed by text-mining tools. This section goes over the text pre-processing steps.

First of all, the “lynx” command is used under linux in order to convert the HTML file into plain text. Then some cleaning scripts are used so that noise (for example, HTML links) and

sections not to be considered in the study can be removed from the text file.

Full stops are quite important as they are used to tokenise sentences later on. Therefore, I make sure full stops that do not represent the end of a sentence (*e.g.*, “Fig.”) will not be regarded as such. Also, words that are not part of the next sentence (*e.g.*, section title such as “Introduction”) need to be separated from the following text by adding a full stop.

Finally, it is common in biomedical articles to have titles for paragraphs within a section or subsection of the paper. Sometimes these titles do not end with a full stop, they are still recognised by a human eye, as they appear in bold, with a paragraph then starting on the next line. Unfortunately, without a full stop, my sentence tokeniser would not separate the title from the first sentence of the paragraph; this is why the cleaning process also adds full stops in these cases.

The cleaning process involves:

- Getting rid of noise created by HTML links, examples of such lines:
 - [1][title.gif]
 - [3][Home][4][Help]
 - [25][Top]
 - Right arrow [11] Abstract freely available
 - View larger version (32K)
 - View this table
 - 1. file://localhost
 - 256. <http://www.pnas.org/>
- Getting rid of all the blank lines and metadata before the title
- Getting rid of Figure and Table legends (because they can sometimes appear in the middle of a sentence and create problems)
- Getting rid of the end of files, Acknowledgments and References (not Footnotes as they may contain sentences of interest)
- When a section title is found (such as Title, Abstract, Introduction, Results, Discussion, Materials and Methods, Footnotes, Acknowledgments, References), a full stop is added after it (so that it will not be considered as the first word of the section’s paragraph).
- Making sure the full stops in “Fig.” and “Tab.” are not considered the end of a sentence (otherwise it breaks a sentence into two)
- Making sure a sentence only contains valid characters before validating it

- Adding full stops at the end of subsection/paragraph titles (so that the title is not considered as part of the first sentence of the paragraph). Going through the text, I look out for empty lines, as a sequence of two or more empty lines might indicate the start of a new paragraph. Then I need to check
 - there is not already a full stop at the end of the title (two full stops would also confuse the sentence tokeniser),
 - the next line after the title (which might have been on one or two lines) started with a capital letter.

Although porting to a new set of journals would require changes to the pre-processor, the current version is very well adapted to its current task. It was evaluated on the test set (see Section 3.3.2) and performs with 94% accuracy (the sentence tokeniser with 92%).

3.2 Set of gazetteers

As noted in Section 2.6, this thesis work offers customised IE to a database annotation's needs. In order to provide for this level of customisation, I created eight gazetteers so as to automatically recognise NEs of interest. The protein names gazetteer was built based on all the protein entries' names and aliases in the NPD. The compartment names gazetteer contains all the compartment names known to be in the nucleus, as well as some important compartments of the cell, such as the ribosome and the ER (endoplasmic reticulum). I handcrafted all the other lexicons by observing positive sentences in the training data. Advice from an expert - the curator of the NPD - was also sought.

There is one gazetteer per NE category, and terms cannot appear in more than one gazetteer. The categories are given in Table 3.1, along with the number of instances found in each (which includes aliases and different forms of the terms). The gazetteers are specific to extracting data about proteins in the nucleus. Section 6.1 discusses how extensible the approach is, given how easy it is to modify or add new elements to gazetteers, provides ideas of what resources could be used in order to achieve this, and considers methods that could be applied to improve results.

3.3 Corpora

Professor Wendy Bickmore - the creator and curator of the NPD - would regularly extract information from the literature and update her database through the process shown in Figure 1.10. In order to create a training corpus as well as some testing data, I asked her to manually highlight sentences she was normally interested in when updating the NPD. The articles contained in the

Category	Abbreviation	No in gazetteer
protein name	PN	7787
protein keyword	PK	12 (see Appendix D)
compartment name	CN	89 (see Appendix C)
compartment adjective	CA	11
compartment keyword	CK	60
phase name	PHAS	26
loca stem	LOC	45
word of interest	INT	163

Table 3.1: Categories of NE, their abbreviation and number of instances in each gazetteer

training set (see Section 3.3.1) and testing set (see Section 3.3.2) were part of the next batch of publications she was going to extract information from. The number of papers contained in both corpora were the result of what Bickmore had highlighted in a certain timeframe rather than a conscious decision of limiting the sets to those numbers.

3.3.1 Training set

She chose a set of 14 articles from the next publications she was about to annotate ([PPRMV04, ALF⁺02, SWJ⁺00, CDG⁺03, CSK98, DMO00, DO98, KZCJ02, LS02, MKC⁺05, PDF00, RRB⁺03, SSS⁺05, SPL00]) that constituted a representative selection of the kind of papers she, the manual curator, has to deal with on a regular basis. However, the 14 documents have one thing in common: they all relate to the localisation of proteins in the nucleolus (see Section 1.2.1). See Section 6.1 for a discussion of extending the annotation of protein localisation beyond papers on the nucleolus.

Once those 14 papers were identified, Bickmore manually annotated them by highlighting sentences that were relevant to the localisation of a nuclear protein within a cell at a particular stage of the cell cycle. The cleansed version of each HTML file was mapped automatically to a file consisting of a sequence of feature vectors (see Section 2.2.6), one per sentence. Highlighting on the paper version of the article was added manually as a positive class label to the corresponding sentences. This is to say, a sentence was represented by its feature vector followed by this class label, yes or no $\{y, n\}$ (e.g., “pn3, pk0, cn1, ca1, ck0, ph0, loc0, int1, ra1, rb0, rc1, 30, y”). The first eight features relate to the number of words found in the sentences that belong to certain categories (see Table 3.1). The next three features are Boolean rules. All features are discussed in Section 3.4.1.

PMID	Number of sentences	Number of positives	Percentage of positives
14973189	106	43	40.56%
10931858	252	33	13.09%
11062265	198	63	31.82%
12397076	158	27	17.09%
11074001	200	25	12.5%
9725903	208	33	15.86%
12963707	165	36	21.82%
11790298	197	12	6.09%
9420331	205	38	18.54%
12134069	206	38	18.44%
11948183	198	34	17.17%
16186106	172	41	23.84%
16129783	180	41	22.78%
10891491	193	21	10.88%

Table 3.2: Number of sentences (total and positives only) for each full-text paper in the training corpus and percentage of positives in each paper

On the other hand, sentences that were not highlighted were manually reviewed, as I discovered that Bickmore, like other curators, did not highlight all sentences containing the same relevant information. Hence, a lack of highlighting cannot be interpreted as an assertion of irrelevance. I augmented the set of positive sentences in order to have a more complete and representative set of training data.

Indeed, I confirmed with Bickmore that she had sometimes omitted to highlight some relevant sentences. The reasons for this were that some sentences were repeating sentences she had already highlighted previously, or some sentences were giving information that was already contained in the NPD. Neither of these two reasons for not highlighting material should be relevant to the “interesting/not-interesting” decision to be made, as opposed to the subsequent “novel/seen” decision. Therefore, adding these sentences to my set of positive examples is justified.

The training corpus is composed of 2638 sentences:

- 485 positive sentences (about 18.4%)
- 2153 negative sentences
- with the frequency of positive sentences ranging from 6 to 40.5% just in the 14 articles constituting the training set (see Table 3.2).

Positive examples from [PPRMV04] include:

“BIG1 was also concentrated in nucleolar areas and was coimmunoprecipitated with nucleolin.”

“In HepG2 cells incubated without FBS, virtually all nuclei contained some FKBP13, but the fraction in the cytoplasm was greater than that of BIG1.”

“It seemed clear also that BIG1 and FKBP13 were quite differently localized in the nuclei, just as they were in the cytoplasm.”

Negative examples from [PPRMV04] include:

“Incubation of those cells with FK506 had increased the recovery of both BIG1 and ARF in membrane fractions, but no proteins were precipitated by the FKBP13 antibodies after FK506 treatment, presumably because FK506 binding alters the FKBP13 epitope.”

“Differences between nucleoporin p62 distribution in cells with and without BIG1 in the nucleus are shown in fields of FBS-starved cells.”

“From HepG2 cell nuclei purified by density gradient centrifugation, antibodies against BIG1, nucleoporin, or nucleolin each precipitated also the other two proteins.”

Entity type	PN	PK	CN	CK	CA	PH	LOC	INT
Number in corpus	2290	1237	1946	620	735	413	453	2583
Density/sentence	0.868	0.469	0.737	0.235	0.278	0.156	0.171	0.979

Table 3.3: Detailed description of the training corpus in terms of the types and frequencies of NEs (see Table 3.1)

Table 3.1 introduces the different kinds of NE while Table 3.3 gives a more detailed description of the training corpus based on those. It shows the density of each entity type per sentence (*i.e.* the average number of tokens of each type in a sentence), and the number of entities of each type in the corpus.

3.3.2 Test set

The test set is composed of three papers ([MNM⁺00, OSWG02, SSE⁺98]) again, chosen by Professor Wendy Bickmore from the set she was about to annotate. Table 3.4 shows the percentage of positive sentences in each of these three articles.

PMID	Number of sentences	Number of positives	Percentage of positives
10716735	99	10	10.10%
12045181	173	30	17.34%
98447599	221	29	13.12%

Table 3.4: Number of sentences for each paper in the test set

Positive examples from [OSWG02] include:

“Nuclear localization was mediated by the COOH terminus of c-erbB-3, and a nuclear localization signal was identified by site-directed mutagenesis and by transfer of the signal to chicken pyruvate kinase.”

“Moreover, c-erbB-3 was found in the nucleoli of differentiated polarized MTSV1-7 and exported into the cytoplasm upon addition of exogenous HRG.”

“In MCF10A, MCF-7, T47D (Fig. 2 A), and BT474 cells (unpublished data), LMB clearly caused nuclear concentration of c-erbB-3.”

Negative examples from [OSWG02] include:

“Heregulin (HRG) binds to c-erbB-3 or -4 and induces heterodimerization of these receptors with c-erbB-2. ”

“All c-erbB receptors can form functionally active heterodimers.”

“This drug specifically blocks the chromatin region maintenance (CRM)1 nuclear export factor by covalent modification.”

3.4 Supervised method

Supervised learning uses training data labelled with both features and an output value to train a classifier to learn a good correspondence between features and output. It can then classify new data by reproducing the learned classification.

3.4.1 Set of features

The objects of interest here are sentences paired with an output annotation of “interesting for annotation”/“uninteresting”, hence the need to represent sentences as feature vectors (see Section 2.2.6). A lot of features can be considered valid candidates to represent a sentence. Most of them arise when parsing the sentences: they can be considered *basic features*. Others can be computed from basic features. The following list contains all the features that I have considered to represent sentences.

- number of protein names (*e.g.*, “fibrillarin”)
- number of protein keywords (*e.g.*, “protein”)
- number of protein-related terms, which groups both protein names and protein keywords
- number of compartment names (*e.g.*, “nucleolus”)
- number of compartment adjectives (*e.g.*, “nucleolar”)
- number of compartment keywords (*e.g.*, “compartment”)
- number of compartment-related terms, which combines compartment names, compartment adjectives and compartment keywords
- number of cell-cycle phase names (*e.g.*, “interphase”)
- number of loca stems (see Section 2.2.2, *e.g.*, “localized”)
- number of verbs of observation (verbs such as “appears”, “contains” and “identified”)
- number of verbs of movement (verbs such as “shuttles”, “moves” or “trafficking”)
- number of words of interest (observation or movement term, *e.g.*, “appears”, “found”, “visible”, “present”, “shuttles”, “moves”)

- the percentage of relevant words in a sentence (without counting stop words, see Section 2.2.3 and Appendix E)

I have also considered the following Boolean features derived from the above basic features.

- Rule A is set to 1 if
 - the number of protein names > 0
 - the number of compartment adjectives > 0 .

(*e.g.*, “nucleolar fibrillarin”)

- Rule B is set to 1 if
 - the number of protein names > 0
 - the number of compartment names > 0
 - the number of loca stems > 0 .

(*e.g.*, “Fibrillarin is localized in the nucleolus”)

- Rule C is set to 1 if
 - the number of protein names > 0
 - the number of compartment names > 0
 - the number of words of interest > 0
 - the percentage of relevant words in the sentence (calculated without counting the stop words) is ≥ 20 .

(*e.g.*, “Fibrillarin was found in the nucleolus”)

Before settling for a final set, features were tested by experimenting with different combinations of them and studying the cross-validation results on the training set. As a result, the following features were discarded:

- the number of verbs of observation and the number of verbs of movement. Both were interesting features but not as powerful as the feature “words of interest”. The latter counts all the words (not just verbs) whose stems refer to terms of observation and movement.
- the number of protein related terms, which grouped both protein names and protein keywords. Two separate features gave better results. As Section 3.4.2 explained, protein keywords constitute a much more reliable feature than protein names. Therefore, it is important for the classifier to be able to access this number independently.

- the number of compartment related terms, which combined compartment names, compartment adjectives (*e.g.*, the word “nucleolar”) and compartment keywords (*e.g.*, the words “compartment”, “region”, “site”, “foci”). Three separate features gave better results.

My final chosen set is therefore composed of eleven features:

- number of protein names
- number of protein keywords
- number of compartment names
- number of compartment adjectives
- number of compartment keywords
- number of phase names
- number of loca stems
- number of words of interest
- rule A (Boolean feature)
- rule B (Boolean feature)
- rule C (Boolean feature)

Other possible features would have been:

- the number of adverbs of interest.
- regular expressions (Boolean feature set to 1 if the sentence matches the regular expression, 0 otherwise). Regular expressions allow users to define flexible patterns they wish to look for. For example, here we would be interested to capture the following flexible pattern or “frame”: protein name, up to five words, the verb “localizes” followed by the preposition “in” and a compartment name. BioIE (see Sections 2.3.3 and 3.5) uses such regular expressions.
- other rules, such as A, B and C above. Example of another possible rule: set the rule to 1 if
 - the number of protein names or keywords > 0
 - the number of loca stems or words of interest > 0
 - the number of prepositions of interest > 0

- the number of compartment names or keywords > 0.

(*e.g.*, “These proteins were visible in CBs”)

I used the Curran and Clark (C&C) tagger [CC03b] to identify basic features (such as the number of protein names or adverbs of interest per sentences), and then a separate program to compute the derived features.

Many other features could have possibly been considered, but as this was only one part of a thesis aimed at demonstrating that a complete end-to-end system could be developed for part-time curators, I stopped with these. This could clearly be developed in future work. For example, parsing would give more features, as would the name of the section in which the sentence was found, whether the sentence was a Figure caption, whether it also occurred in the abstract, *etc.*

3.4.2 From sentences to feature vectors

In order for the training corpus to reflect all these features, I needed to develop a procedure that processed each sentence in the training data, computed results for each feature, and store these results in a vector, called a feature vector (see Section 2.2.6). I used the C&C tagger [CC03b] to recognise NEs of interest.

Each line in the C&C tagger input file follows the same format: WORD POS CLASS. POS stands for Part Of Speech (see Section 2.2.1) while CLASS is the category the word has been classified as by the tagger. Moreover, it is possible to add extra columns to this input file, which the C&C tagger will take into account when making its decisions. A fourth column was added to show whether the word was contained in any of my lexicons (see Section 3.2). The C&C tagger input file is then of the following format: WORD POS CLASS LEXICON.

The C&C tagger uses the IOB system for this CLASS column:

- I- the word on this line is the continuation of an NE (**I**nside the NE),
- O- the word on this line is not an NE (**O**ut),
- B- the word on this line is an NE (**B**eginning of the NE, *i.e.* could be continued on the next line).

Examples of input lines are:

- Nuclear JJ B-CA CA
- localization NN B-LOC LOC

The word “Nuclear” is an adjective (POS JJ), it has been classified by the tagger as the beginning word of an NE of type “compartment adjectives” (B-CA) and it had also been found in

that particular lexicon (CA). The word “localization” is a noun (POS NN), it has been classified by the tagger as the beginning word of an NE of type “loca stems” (see Section 2.2.2, B-LOC) and it had also been found in that particular lexicon (LOC).

For the training corpus, the CLASS column in the C&C tagger output file required a little bit of manual annotation in order to use the IOB system to its full potential. The lexicon column was automatically copied over to the CLASS column. If the lexicon value was not null, it was simply a case of deciding whether an “I-” or a “B-” should go in front of the lexicon value for the class value. The C&C tag or class value indicates which category (see Table 3.1) a word has been classified as. For example, Table 3.5 displays the four columns for the third sentence of the abstract of article [SPL00]:

“Immunofluorescence analysis shows that Bop1 is localized predominantly to the nucleolus.”

Immunofluorescence	UH	O	NONE
analysis	NN	O	NONE
shows	NNS	B-INT	INT
that	IN	O	STOP
Bop1	NNP	B-PN	PN
is	VBZ	O	STOP
localized	VRN	B-LOC	LOC
predominantly	RB	O	STOP
to	TO	O	STOP
the	DT	O	STOP
nucleolus	NN	B-CN	CN
.	.	O	NONE

Table 3.5: Four column-output from the C&C tagger for the third sentence of the abstract of article [SPL00]. (The POS-tags for “shows” and “to” were incorrectly assigned. The word “shows” was incorrectly POS-tagged but correctly recognised in the “words of interest” lexicon.)

The first entity recognised is “shows”, a term from the “words of interest” lexicon. “Bop1” is recognised as a protein name, “localized” as a loca stem and “nucleolus” as a compartment name. A few stop words are recognised too. Stop words do not represent an NE category of interest and infer the C&C tag O. Stop words are only recognised by the fourth column in order to count the number of non-stop words per sentence so as to generate the percentage of relevant words in a sentence. Appendix F provides other examples of sentences.

For the test set, the trained C&C tagger is run and gives an output corresponding to the input file with the CLASS column having been modified based on what was learned from the training corpus. The fourth column is used when training the tagger and when the C&C tagger produces an output file for a test set. However, the C&C tagger does not make all its decisions solely on the basis of gazetteers it consults, as it is trained with my training data before being applied to test data.

A perl script was written to convert the output of the C&C tagger into feature vectors for each sentence. Counting B- starting entities in the third column ensured counting an NE spread over several words only once. Finally, a simple function calculates the percentage of relevant words in a sentence (without counting the stop words). Another one sets up the correct Boolean flags for the rules (RULE_A, RULE_B, RULE_C).

For the example in Table 3.5, the following feature vector is obtained: “pn1, pk0, cn1, ca0, ck0, ph0, loc1, int1, ra0, rb1, rc1, 57.14285714, y”. One protein name or “B-PN” (in this case Bop1) was found. No protein keyword, compartment keyword nor phase name were identified. One compartment name, one loca stem and one word of interest were found. Two rules were set to 1 and one to 0. Finally, 57.14% of the “non-stop” words in this sentence were identified as NEs and therefore relevant. Again, Appendix F gives further examples. Table 3.6 shows the possible values for each feature in the vector. Their number was adjusted to achieve the best cross-validation results on the training data.

Features	Values
protein name	{pn0, pn1, pn2, pn3, pn4, pnn}
protein keyword	{pk0, pk1, pk2, pk3, pk4, pkn}
compartment name	{cn0, cn1, cn2, cn3, cnn}
compartment adjective	{ca0, ca1, ca2, ca3, can}
compartment keyword	{ck0, ck1, ck2, ck3, ckn}
phase name	{ph0, ph1, phn}
loca stem	{loc0, loc1, locn}
word of interest	{int0, int1, intn}
rule A	{ra0, ra1}
rule B	{rb0, rb1}
rule C	{rc0, rc1}
percentage	real number
relevant	{y, n}

Table 3.6: Feature vector’s possible values. Should more than one phase name of the cell cycle occur in a sentence, the feature vector would show “phn”.

The C&C tagger output on the test set shows the NER is performed with:

- 51% recall and 82% precision for the protein names (PN),
- 99% recall and 99% precision for the other categories.

It is well-known that variation in how proteins are specified means that NE recognisers for proteins either must allow for a wide range of variation or will miss a significant number of protein references (false negatives). Even though the protein NE recogniser misses many instances, in 82% of cases, the protein keyword feature picks up the presence of a protein. Protein keywords can therefore be considered as a reliable safety net.

The features can be tested on how well they represent the data by running cross-validation experiments on the training corpus. Table 3.7 shows these results for three different methods: DT (see Section 2.2.8), NB (see Section 2.2.7) and MaxEnt (see Section 2.2.9). WEKA (see Section 2.2.6) was used to obtain results from DT and NB. The MaxEnt tool used in my experiments is Zhang Le's toolkit [Le03], which only outputs an accuracy percentage (rather than precision, recall and F-score results) on cross-validation results. Three quarters of the sentences are correctly identified as interesting when using cross-validation. The next section applies this trained model to test data.

	Decision Tree	Naive Bayes	MaxEnt
Precision	0.755	0.738	
Recall	0.713	0.767	
F-score	0.734	0.752	accuracy: 81.673%

Table 3.7: 10-fold cross-validation on the training corpus

In order to ascertain whether the size of the training data set was adequate, or whether the range in variation from paper to paper meant that a larger training set was required, I characterised the learning curve of my system. I identified the best results I could achieve on the test set from training on the full set of 14 articles, then only 13, 12, 11, 10 articles, *etc.* Results are shown in Figure 3.1. Performance starts plateauing early so it is safe to say that I have sufficient data.

I would not say the curve is plateauing too early because, although the x axis on the graph represents the number of documents, the classifier is actually learning how to categorise sentences. As Table 3.2 shows, these articles contain between 106 and 252 sentences each. Therefore, the curve is not showing that one element is enough to learn how to classify another element, the curve is showing one set of elements is enough to reach a certain level of confidence to classify other elements. Furthermore, recall does not plateau until the 7th document. Nevertheless, a larger training set could potentially yield better results.

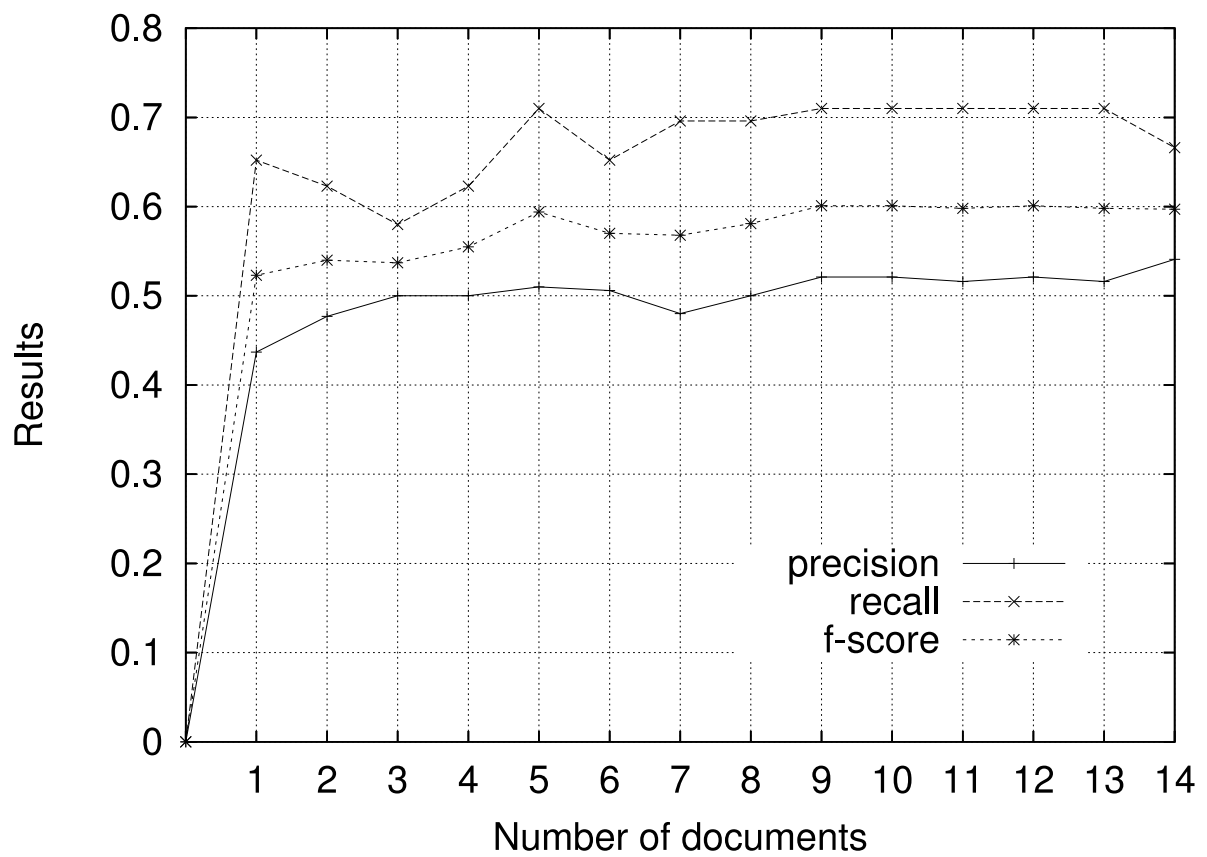


Figure 3.1: Learning curve on the size of the training corpus showing there are enough labelled documents in this collection.

3.4.3 Baseline

A baseline is required to assess system performance. Precision and recall for this baseline show, in part, the contribution of NER to the decision. The baseline used here was created using a classifier developed by Tamara Polajnar, based on the same approach as the SVM (see Section 2.2.10) used for PreBIND [DMdB⁺03]. The MexSVM (<http://webpace.ship.edu/thbrig/mexsvm/>) Matlab interface for the SVMlight classifier is used for these SVM experiments. Table 3.8 shows the results obtained by this classifier categorising sentences in the training and testing sets as relevant or irrelevant.

	Cross-validation on training set	Testing on test set
Accuracy	87.98	84.80
Precision	67.67	0
Recall	66.53	0
F-score	66.70	0

Table 3.8: Baseline results using SVM on cross-validation on the training set as well as testing on the test set

The results show SVM guesses mainly negative, which gives it a high accuracy but a precision, recall and F-score of zero as no positives were present in the returned set. High accuracy is a consequence of the large number of true negatives in the training and test sets, which the SVM-based method was very good at recognising. This experiment demonstrates that using SVM bag of words is, in this case, similar to guessing negative all the time.

Comparing the results in Table 3.8 with those in Table 3.7 shows that the SVM approach performs worse with respect to cross-validation than the other three approaches – Decision Tree, Naive Bayes and MaxEnt. The next section presents experiments and results achieved on the test data set. The SVM baseline obtained on the test set will serve as the baseline of reference for these experiments.

3.4.4 Experiments and Results

Table 3.9 gives the results obtained for the same three methods on the test set. Ensemble methods make decisions based on the individual decisions of the different classifiers. In Ensemble methods, the individual classifiers vote on each item. Their votes can be equally weighted, or one might learn a better weighting.

I tried Ensemble methods weighted differently on the results from Decision Tree, Naive Bayes and MaxEnt. In most cases, Ensemble Learning did not achieve better results than the best technique out of the three (DT). Usually, Ensemble can correctly classify some FNs as TPs

	Decision Tree	Naive Bayes	MaxEnt
Precision	0.569	0.519	0.533
Recall	0.652	0.580	0.348
F-score	0.608	0.548	0.421

Table 3.9: Results on the test set

that the best technique would have missed out. Table 3.10 shows DT, NB and MaxEnt do not learn the data differently and must agree most of the time. Because NB and MaxEnt are lower versions of DT in this case (*i.e.* they do not perform as well as DT), there is no gain obtained by using an Ensemble method.

DT	Ensemble		Precision	Recall	F-score
	NB	MaxEnt			
1	1	1	0.541	0.666	0.597
1.2	1	0.8	0.527	0.695	0.6
1.5	1	0.5	0.527	0.695	0.6
1.5	0.75	0.75	0.527	0.695	0.6
1.8	0.6	0.6	0.527	0.695	0.6
2	0.5	0.5	0.539	0.695	0.607
2	0.7	0.3	0.539	0.695	0.607
2	0.4	0.6	0.539	0.695	0.607
prio/1	/1	/1	0.527	0.695	0.6
prio/1	/0.6	/1.4	0.51	0.71	0.594
prio/1	/1.4	/0.6	0.51	0.71	0.594
prio/0	/0.5	/2.5	0.51	0.71	0.594
prio/0	/2.5	/0.5	0.51	0.71	0.594
prio/0	/1.3	/1.7	0.51	0.69	0.586
prio/0	/1.5	/1.5	0.51	0.69	0.586

Table 3.10: Ensemble results on the test set. Numbers in the first 3 columns correspond to the weight given to each method (DT, NB and MaxEnt) by the Ensemble method; “prio” means a method gets priority if it has classified an instance as positive itself.

The last 7 rows of Table 3.10 show Ensemble methods where DT - performing best out of the 3 methods - gets priority (“prio”) if it has classified an instance as positive. If it has not then different weights are given to the three methods. Although for some combinations recall

is higher than DT's performance, none of the different combinations of weights produced a higher precision and DT's F-score is never outperformed. Precision is more important to the curator as any important piece of information will be repeated and hopefully caught at some point within an article (see Section 3.4.5). Therefore, I will use DT's results through the rest of the thesis.

In order to make my classifier most useful, I used MaxEnt to order the results on my testing sets. It seems that even if the ranking was not performing very well, it can only be better than lists ordered by sentence number. Ranking is an interesting idea as it allows the curator system interface to show the user which sentences are "very likely" to be relevant and which are "possibly likely". This can be shown by different shades of colour (see Section 4.5). While ranking using MaxEnt percentages might not be the most effective method, it can still be useful if precision@n scores are high enough above a threshold that one can ascertain and low below the threshold.

Based on DT's results, I computed ranked lists of sentences ordered by their MaxEnt percentage. From these lists, it was possible to calculate the results @n, as shown in Table 3.11.

Results@n, n =	10	20	30	40	50	60	70
Precision	0.5	0.5	0.5	0.525	0.54	0.533	0.557
Recall	0.071	0.143	0.214	0.3	0.386	0.57	0.557
F-score	0.124	0.222	0.3	0.382	0.45	0.55	0.557

Table 3.11: Results on the test set @n

3.4.5 Analysis of the results

Table 3.7 shows that three quarters of the sentences are correctly identified when using cross-validation. When this trained model is applied to test data, it does not do as well. However, it still delivers approximately 60% of the sentences correctly (see NB and DT in Table 3.9).

The difference in performance between cross-validation and application of the model to the test set could have been the result of the way cross-validation was performed. Indeed, assigning sentences at random to a fold (like WEKA does by default, see Section 2.2.6) may result in folds in which sentences from the same document (possibly expressing the same content) appear in both training and testing sets. This would mean that the test set, while unseen, may be more similar to the training set than would be the case with a held-out test set, resulting in higher cross-validation scores than the scores on the test set.

The following experiment was conducted in an attempt to rectify the problem by ensuring that, in each fold, the sentences in the test set come from different documents from the sen-

tences in the training set. As there are 14 documents in the training set, these were used to make seven folds, each using 12 documents in the training set and two in the test set. Table 3.12 shows the results for this experiment.

The results are more or less the same, sometimes lower, sometimes higher. Another reason for the difference of performance between cross-validation and application of the model to the test set could be that the NER is not performing as well on the test set as it was on the training corpus. Section 3.4.2 provides NER results on the test set and explains how the protein keyword feature aids to recognise a sentence discusses a protein when the protein name itself was not caught (or in the case of an anaphora).

Fold no	DT prec	DT rec	DT F-sc	NB prec	NB rec	NB F-sc
1	0.745	0.661	0.701	0.705	0.693	0.699
2	0.753	0.813	0.782	0.744	0.853	0.795
3	0.581	0.895	0.705	0.572	0.934	0.710
4	0.829	0.812	0.821	0.800	0.833	0.816
5	0.926	0.431	0.588	0.919	0.586	0.716
6	0.805	0.644	0.716	0.807	0.744	0.774
7	0.875	0.645	0.742	0.831	0.711	0.766

Table 3.12: ‘DIY’ cross-validation results (precision, recall and F-score) on the training corpus

Looking at the results @n, a slow but steady increase with n on recall and F-score is observed, where precision is always about 0.5. Indeed, the results in Table 3.11 show that relevant sentences do not congregate at the top of the ordering. As I go down the ranking, I appear to be picking up an approximately equal number of relevant and irrelevant sentences.

There is a lot of repetition in biomedical papers. As a result, the same information can be extracted from several different sentences within an article. As in other kinds of publication, there is repetition between the Abstract and the Discussion sections of a paper. In biomedical publications, the end of the Introduction section usually summarises the article too. Trying to convince the readers that their points are valid, the authors will confirm the main facts of the paper several times in the Results section when elaborating and justifying their arguments based on different techniques and results. Also, a piece of information introduced at one point in the text can subsequently be presented as background information. For example, if at some stage in the paper it is established that protein X is localised in the nucleolus, several times in the rest of the document, protein X can be referred to as “nucleolar protein X”.

On top of these results, and as a transition to the next chapters of the thesis, I also computed results for another metric, which I call A@n (see Section 2.1.3). This metric is like recall@n

where only the first member of an equivalence class of answers is counted. Results on the test set for $A@n$ are shown in Table 3.13.

To get these figures, I manually assessed how many actual different “topics” were in the relevant sentences, by grouping sentences per type of information (which is done automatically in the annotation system - see Section 4.3). I then counted the number of clusters obtained. While I found 28 clusters in the three-paper test set, not all of them are important. Important topics are core to an article and thus tend to get repeated a lot within the full text paper. For example, throughout the 14 articles comprising the training corpus, on average 83.063% of the sentences addressing localisation of a protein are sentences covering “major” topics, *i.e.* topics referring to the main protein(s) of interest in the publication, being located in the nucleus or a subnuclear compartment. I considered there were 5 major topics in the test set: c-erbB-3 in nucleus, c-erbB-3 in nucleolus, DEDD in nucleus, DEDD in nucleolus, DEK in nucleus. Results shown in Tables 3.13, 3.17, 3.20 are based on these 5 main topics in the test set. There is a discussion about this in Section 3.7, where these results are also given when taking into account all of the 28 topics.

n	5	10	14
$A@n$	0.4	0.8	1

Table 3.13: Results on the test set $A@n$

The results $A@n$ show that if not all the occurrences of a type of information are caught with my classifier, at least no information is lost. Not only did I catch at least one instance of each type of information in the test set, I also got those instances ranked high in the ordered list produced by my supervised method. In the top five sentences of my ranked ordered list, almost half of the topics have been covered already (see Table 3.13). I only need to go down to the 14th sentence of the list to get all the topics covered.

3.5 Rule-based method with BioIE

This section compares the method I developed and presented in the previous section to an existing one based on pre-defined rules. Comparing results obtained by different methods on the same data offers an interesting perspective as to what their respective strengths and weaknesses are (see Section 3.7).

In Section 2.3 of the last chapter I presented BioIE [DA05a], a rule-based sentence extraction system. The tool offers five predefined categories of types of information related to proteins:

- structure,

- function,
- diseases and therapeutic compounds,
- localisation, and
- familial relationships.

BioIE stores different templates for each of these five categories. The templates used for the localisation category are displayed in Table 3.14, where regular expressions have been used to formulate the patterns sought. For example,

- *exist[a-z]0,3 in* refers to “exist in”, “exists in”, “existed in”, “existing in” as *[a-z]0,3* stands for any letter between ‘a’ and ‘z’ repeated 0, 1, 2 or 3 times.
- *locali[s|z]e[a-z]0,1 in* refers to “localize in”, “localizes in”, “localized in”, “localise in”, “localises in”, “localised in” as *[s|z]* gives a choice of using an ‘s’ or a ‘z’.

Out of the 22 patterns BioIE uses, 12 contain words that are present in two of my lexicons (local stems and words of interest). The first column of the Table displays those 12 items while the second column shows patterns containing words my lexicons do not use.

contained in	derived
detected in	extracellular
distributed in	encoded in
distributed along	discovered in
found in	common in
found within	inside
found only in	expressed in
found throughout	intracellular
found at	exist[a-z]0,3 in
observed in	allocat[a-z]0,3
locali[s z]e[a-z]0,1 in	
colocali[s z]e[a-z]0,1 with	

Table 3.14: BioIE templates for the localisation category [DA05a]

The results obtained by loading the training corpus onto BioIE’s Webpage¹ are indicated in Table 3.15. Results on the training set give a precision of 0.401, a recall of 0.361 and an F-score of 0.380.

¹I would like to thank Dr Anna Divoli for altering the BioIE’s 200 KB limit for a weekend so that I could compute the tool’s results on my training corpus.

$$\begin{aligned} TP &= 175 & FN &= 310 \\ FP &= 261 & TN &= 1892 \end{aligned}$$

Table 3.15: Confusion Matrix for BioIE on training set

The results obtained by loading the test set onto BioIE's Webpage are indicated in Table 3.16. The first column of the Table gives "regular" precision, recall and F-score on BioIE's results as a whole, whereas the next columns show results for precision, recall and F-score @n.

Results	BioIE	@10	@20	@30	@40	@50	@60	@70
Precision	0.280	0.2	0.25	0.266	0.275	0.24	0.266	0.271
Recall	0.328	0.029	0.071	0.114	0.157	0.171	0.229	0.271
F-score	0.303	0.051	0.11	0.161	0.2	0.2	0.246	0.271

Table 3.16: Results produced by BioIE on the same test set

The sentences extracted by BioIE are ranked in order of importance, according to the number and complexity of templates they contain. The templates that BioIE uses exhibit a range of syntactic complexity. The more complex a template is, therefore more specific and thus containing more precise information, the more it is weighted. Less complex templates are also considered but are weighted less. As BioIE's results are rank-ordered, I can calculate its A@n results as well. These results are shown in Table 3.17. It is only by the 39th result that all five major topics were caught.

n	10	20	30	39
A@n	0.4	0.8	0.8	1

Table 3.17: Results produced by BioIE on the test set A@n

3.6 Unsupervised method: Vector Space Models with Infomap

After presenting results obtained on my data by BioIE, a rule-based method, another interesting method to compare mine with is that of Infomap. Section 3.7 will discuss all the results presented in this chapter.

As noted in Section 3.4, unsupervised learning, unlike supervised learning, does not benefit from any labelled data. Instead, it looks for patterns in the data. Groupings within the patterns

	Sentence 1	Sentence 2	Sentence 3
Term 1	x_1	y_1	z_1
Term 2	x_2	y_2	z_2
Term 3	x_3	y_3	z_3
Full query	$x_1 + x_2 + x_3$	$y_1 + y_2 + y_3$	$z_1 + z_2 + z_3$

Table 3.18: Normalised term-sentence matrix. The last row shows semantic composition by simple vector addition (if the full query is composed of the three terms in the first three rows).

may be taken to indicate distinct classes. Patterns can sometimes be seen by mapping each data item to a point in a vector space whose dimensions correspond to the features. This is called a *Vector Space Model (VSM)* of the data.

Infomap [WP] is an information retrieval system based on VSM. It can retrieve items containing free text at any level. The article level is used for document retrieval. Here, I used Infomap to retrieve items at the sentence level.

Infomap initially uses the VSM to interact with the user and create a query that reflects both the concept of interest to the user and how it is realised in the document collection. My original query was simply the word “localized”. Infomap then offered me a list of 50 related words that I could add to my query if I felt it increased the accuracy of its meaning, or disregard otherwise. I rejected terms such as “accessible”, “aspect”, “beneath”, “distant”, “involvement” and “overlying”. The final query contained the most relevant 25 terms: “confined”, “densely”, “detected”, “diffusely”, “discovered”, “distributed”, “localised”, “localization”, “localize”, “localized”, “localizes”, “localizing”, “located”, “locates”, “locations”, “migrate”, “moves”, “predominant”, “predominantly”, “present”, “region”, “regions”, “site”, “sites”, “situated”.

Infomap then computed a normalised term-sentence matrix that represented sentences as vectors in the same space as the query words. This allowed semantic composition by simple vector addition as illustrated in Table 3.18. The most relevant sentence was the sentence whose vector had the highest cosine similarity with the full query vector. For example, the three sentences in Table 3.18 could be represented in a three-dimensional space (term 1, 2 and 3 from the Table now correspond to 3 elements in a vector) where

- $Sentence1 = (1,0,0)$,
- $Sentence2 = (0,1,0)$ and
- $Sentence3 = (0,0,1)$.

Then

- $\cos(\text{Fullquery}, \text{Sentence1}) = x1 + x2 + x3$,
- $\cos(\text{Fullquery}, \text{Sentence2}) = y1 + y2 + y3$ and
- $\cos(\text{Fullquery}, \text{Sentence3}) = z1 + z2 + z3$.

Whichever cosine value is highest (lowest) would show which sentence is the most (least) relevant. Infomap can thereby return a rank-ordered list of best matching sentences to the query.

Infomap's results are further improved by using a statistical technique called **Singular Value Decomposition (SVD)**. SVD maps a vector space to a lower dimension and looks at word relationships by studying the distribution of its co-occurrences in order to associate terms with similar meanings. For example, SVD might be able to learn that a protein name and its aliases are related terms. The results obtained by loading the test set in Infomap are indicated in Table 3.19. The first column of the Table gives "regular" precision, recall and F-score on Infomap's results as a whole, whereas the next columns show results for precision, recall and F-score @n.

Results	Infomap	@10	@20	@30	@40	@50	@60	@70
Precision	0.32	0.8	0.75	0.566	0.475	0.42	0.4	0.371
Recall	0.457	0.114	0.214	0.243	0.271	0.3	0.343	0.371
F-score	0.376	0.2	0.333	0.34	0.345	0.35	0.37	0.371

Table 3.19: Results produced by Infomap on the same test set

The Infomap output is a list of sentences best matching the query in descending order of relevance. As Infomap's results are rank-ordered, I can calculate its A@n results. These results are shown in Table 3.20.

n	10	20	30	40	49
A@n	0.4	0.8	0.8	0.8	1

Table 3.20: Results produced by Infomap on the test set A@n

The results presented above were for the test set. For completeness, I tried to obtain results on the larger training set. Unfortunately, testing the training corpus on Infomap was more difficult as Infomap will at most retrieve 200 documents (sentences in my case). The training corpus contains 485 positive sentences (and 2638 sentences in total). According to support groups on Infomap, it is possible to alter the source code in C and get the tool to retrieve a higher number of results. Unfortunately, this requires a lot more than simply changing the

number “200” in the original code. Based on what I know Infomap retrieved for the first 200 results, Table 3.21 sums up my calculations.

Results	hypothetical	worst case	best case	regular case
Number of +ves	200	485	485	485
TPs retrieved	104	104	389	104
Sentences retrieved	200	485	485	200
Precision	0.52	0.214	0.8	0.52
Recall	0.52	0.214	0.8	0.21
F-score	0.52	0.214	0.8	0.299

Table 3.21: Infomap's results on the training set. The first column shows the results if the training corpus contained 200 positives rather than the actual 485. The second column gives the results in the worst case scenario (Infomap does not find any more positives). The third column gives the results in the best case scenario (Infomap finds all the remaining positives). Finally, the last column shows the results as they are (only 200 sentences retrieved but 485 actual positives in the corpus).

The results obtained in this section were for a particular choice of initial query word (“localized”) and for a particular manual pruning that I performed, retaining 25 terms out of the 50 returned by Infomap. Other terms and/or other pruning of the set of words suggested would produce somewhat different results.

3.7 Discussion

Tables 3.9, 3.11, 3.13, 3.16, 3.17, 3.19, 3.20 show that the classifier I developed using supervised ML outperforms the rule-based and the unsupervised ML methods. The best F-score for supervised ML was achieved using Decision Tree (Table 3.9): 0.608. BioIE struggled to attain half this F-score (Table 3.16): 0.303. Infomap scores a little bit better than BioIE, with an F-score of 0.376.

The A@n metric measures the number of sentences retrieved before picking up all the different topics in the test set. It confirms my supervised classifier performs better than the other two tools again, as 14 sentences were enough to cover all the topics in the test set (Table 3.13), against 39 for BioIE (Table 3.17) and 49 for Infomap (Table 3.20).

As I explained in Section 3.4.4, 28 clusters of different sizes were found in the three-paper test set but not all of them were considered important. Therefore, I did not use the number 28 to calculate the results shown in Tables 3.13, 3.17, 3.20. Instead, I worked out there were 5

main topics in the test set, the 5 larger clusters, and used that number. The reason for this is that I wanted to show there is at least one instance of every important topic being picked up high in the rank-ordered list of results.

Is counting only important topics justified? In multi-document summarisation, there is a heuristic that material occurring in many documents - *i.e.* multiple times - is probably more important than material that occurs in only a few documents - *i.e.* a few times. It seems that the above heuristic applies within a document being assessed for IR as it does across a set of documents being summarised. Indeed, in biomedical papers, only facts that are core to the article tend to be repeated. Pieces of information that are in a paper as part of the arguments (*e.g.*, background knowledge) are repeated less often than the actual message authors are trying to put across.

For example, one of the three publications that constitute the test set is mainly about protein c-erB-3 (see [OSWG02]). Proteins c-erB-1, c-erB-2 and c-erB-4 are mentioned in 18 sentences in the full text paper (which comprises 172 sentences). Out of these 18 sentences, 6 talk about localisation of these “satellite” proteins (1 sentence for c-erB-1, 3 for c-erB-2 and 2 for c-erB-4), and 2 were highlighted by our expert. 93 sentences in the document mention c-erB-3, 30 of them talk about its localisation and 20 were originally highlighted by our expert. This means the localisation of c-erB-3 is discussed five times more than the localisation of these three “satellite” proteins all together. In my supervised method, only 1 out of these 6 sentences about satellite proteins was caught. But when I want to evaluate how well my method is doing, I want to check that all the important topics have been caught high in the list, *i.e.* the localisation of c-erB-3.

Moreover, the curator of the NPD only adds to her database facts that are backed up by experimental evidence. If a piece of information is not repeated, either it is a well-known fact in molecular biology (and it is most likely already in the NPD), or there is no (or only preliminary) evidence to support it in the article. In this case, she prefers to wait for a future paper that will give full evidence for this new piece of information before adding it to the database.

Having discussed how results were calculated in Tables 3.13, 3.17, 3.20, I also worked out what the results of A@n would be like if I considered the total 28 topics. The results in Table 3.22 show that none of the three methods retrieve sentences covering all the 28 topics. BioIE only starts performing better than my supervised method somewhere between the top 30 sentences and the top 40. After that, BioIE actually manages to catch sentences about half the total number of topics in the test set. Infomap gives better results than my supervised method in this Table right from the beginning, and manages to retrieve sentences about 42.8% of the total 28 topics.

It seems BioIE's many patterns allows it to open up to more possibilities than my super-

	Supervised	BioIE	Infomap
A@10	0.143	0.107	0.178
A@20	0.178	0.178	0.357
A@30	0.214	0.178	0.357
A@40	0.214	0.321	0.357
A@50	0.286	0.357	0.393
A@60	0.286	0.428	0.393
A@70	0.321	0.428	0.393
A@80	-	0.5	0.393
A@90	-	-	0.428

Table 3.22: Results obtained by 3 different methods on the same test set for A@n considering the total 28 topics

vised method, which focusses on particular features quite quickly. These features are very useful for recognising specific forms associated with important topics, but lack the breadth needed to catch all the different topics.

As for Infomap, the key property for exploiting word co-occurrence patterns is that SVD finds the optimal projection to a low-dimensional space. SVD represents terms and sentences in the lower dimensional space as well as possible. In the process, some words that have similar co-occurrence patterns are projected onto the same dimension. As a consequence, the similarity metric will make topically similar sentences and queries come out as similar even if different words are used for describing the topic. Again, this approach seems more open than my supervised method and I speculate this might be the reason why it captures more of the 28 topics.

But what is more important to a “spare time” database curator? Catching as many topics as possible per document OR making sure that all the main topics of a document are caught early on in the rank-ordered list? When reading the paper mentioned earlier about c-erB-3 [OSWG02], the curator cares about sentences confirming where the protein localises within a compartment of the cell nucleus (*i.e.*, in this case, sentences mentioning c-erB-3 is in the nucleolus). Topics such as locating the protein in the cytoplasm, or about “satellite” proteins (c-erB-1 or c-erB-4) do not matter as much.

Therefore, Tables 3.13, 3.17 and 3.20 reflect that my supervised method is the best out of the three techniques I experimented with in this chapter for the purpose at hand. It is no surprise that supervised learning is doing better in my experiments. Indeed, my classifier benefits from a lot of manually annotated training data and handcrafted gazetteers. With these advantages, it should do better than techniques solely based on templates or VSMs.

More generally, results could be improved by using syntactic features derived from deep-parsing analysis. For example, as mentioned in Section 3.4.2, the protein keyword feature can help to pick up on a relevant sentence that does not contain a protein name (or that does but the latter was not recognised). However, a real anaphora resolution module such as the one described in [Gas06] and used in [KSL⁺08] could enhance results.

Furthermore, as the tool's features are mainly based on the recognition of NEs, improving NER results - notably of protein names - would benefit it. Although, FACTA itself (see Section 2.3.8) does not deal with disambiguation at present and has separate indexes for concepts and for words, working with concepts seems to present an advantage. Indeed, all names and aliases of a protein can be kept under a unique concept identifier or concept accession number. This approach could facilitate merging multiple dictionaries from different databases containing protein names and their variants, thereby enriching the protein name lexicon and keeping track of conceptual NEs.

The use of conceptual features was proven to benefit IR results as early as 2003 in the TREC Genomics *ad hoc* retrieval task ([HB03], see Section 2.4.2), when Medical Subject Headings (<http://www.nlm.nih.gov/mesh/>) and other unambiguous terms were used to obtain better results. Concept recognition tools were first openly evaluated during BioCreative II [KLRPV08], which set tasks such as gene mention tagging [STnA⁺08] and gene normalisation [MLW⁺08]. Based on this recent community-wide effort, [BLJ⁺08] introduces an integrated system that achieves better results than traditional protein name recognition methods by performing concept recognition. This kind of method could significantly enhance our NER results.

Finally, to keep the training data automatically updated and growing, training data could be collected on an ongoing basis. For example, BIND (Biomolecular Interaction Network Database) is curated using an IE system developed to automatically identify protein-protein interactions and present results through an interface for BIND curators. This system is called Pre-BIND/Textomy (text anatomy). In [DMdB⁺03], the authors explain that the training corpus of PreBIND and Textomy is continually improved as the feedback of curators using the tool can be saved whilst they are working. This allows their system to enhance its performance on a regular basis. A similar approach could be implemented for the NPD Curator System Interface.

My final system (see Chapter 5) suggests to the curator what the most important topics are, given a new paper, based on which topics gathered the highest number of instances (see similarity clustering in Section 4.3). The system shows major and minor topics found in the new article. It also gives a counter revealing how many instances were caught per type of information, and gives links to all these sentences highlighted in the full text.

In conclusion, the classifier I developed using supervised ML is the method I used to develop my tool. The A@n measure enabled us to realise that if not all occurrences of a piece of

information are caught by my classifier, at least one of them is for each and every one of them. This brings us to the next step towards developing a tool to help annotate the NPD. When more than one occurrence per topic is caught by my classifier, how can I detect that two or several sentences in my ranked ordered list are actually referring to the same piece of information?

3.8 Summary

In this chapter, I have presented my corpora and my methods for retrieving sentences relevant to protein localisation in full text papers. I have also compared my results to results obtained by performing other tools on my data. Finally, from analysis of the results and discussion emerged a clear conclusion for what my final tool (see Chapter 5) should use for detecting relevancy.

Chapter 4

Elements of automated annotation assistance

Before introducing the Curator System Interface in the next chapter, this chapter presents the technical side of all the different components of automated annotation assistance it provides. While Section 4.1 explains how relevant documents were retrieved (work achieved during my Masters by Research which was integrated as a pre-step to the final interface) and Section 4.2 talks about detecting sentences relevant to the localisation of nuclear proteins (which was covered in the previous chapter), the following two sections present features that were not previously explained in this thesis or elsewhere. Section 4.3 talks about detecting redundancy and grouping sentences that refer to the same kind of information. Section 4.4 talks about detecting novelty with regard to the NPD.

4.1 Document retrieval

4.1.1 Text categorisation task

For my Masters by Research thesis [Can04], I tackled the first issue encountered when maintaining a database, which consists of retrieving documents of interest to the database. The thesis explored different ways to perform automatic classification of articles from the biomedical literature into two classes: articles of interest to the NPD (which therefore should be linked from the database), and articles that are not of interest.

In order to present the curator with results she can easily and quickly exploit, the final tool uses a combination of different classifiers including Rainbow, DT, NB, MaxEnt and displays rank-ordered lists of articles. While sentence-classification did not benefit from an ensemble of multiple methods (see Section 3.4.4), this TC task did.

DT, NB, MaxEnt were introduced in Section 2.2. Bow is a library of C code for statistical

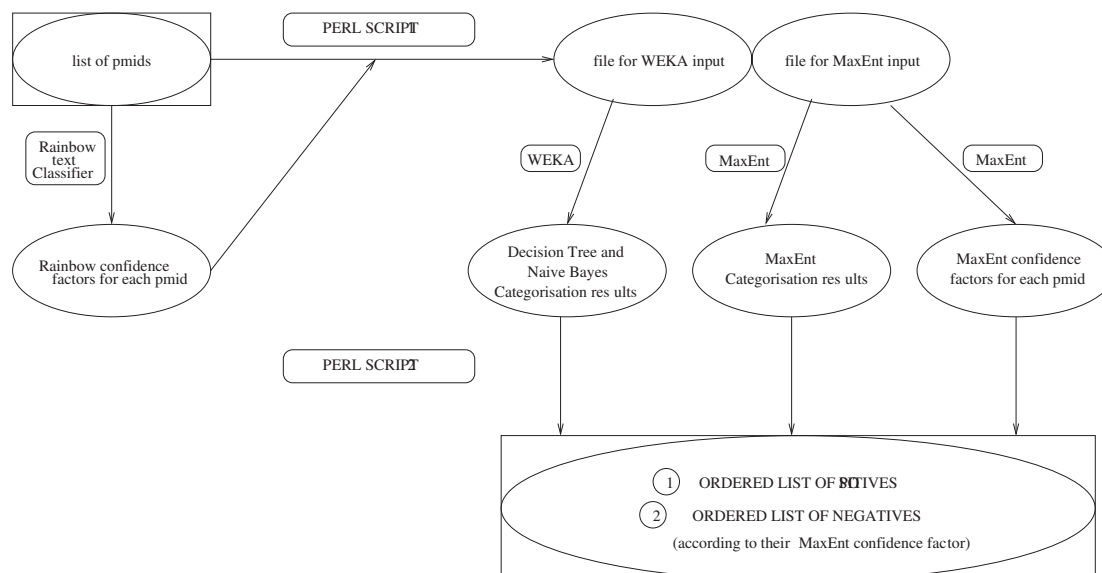


Figure 4.1: Categorisation of PMIDs returned from PubCrawler. The tool uses a combination of existing tools and perl scripts to produce rank-ordered results.

text analysis, language modeling and information retrieval. Rainbow [RAI] is the front-end to the library that supports text classification. It was used as a feature for this tool in order to mimic a domain expert searching the title and abstract for relevant evidence.

Finally, the rank-ordering is performed based on the percentages MaxEnt gives out as shown in Figure 4.1. The tool achieved a precision of 0.7, a recall of 0.636 and an F-score of 0.667.

Before my MRes, Professor Wendy Bickmore would get a daily email from PubCrawler [HW04], a gateway to the biomedical literature that lets users set keywords according to the kind of articles they would like the tool to retrieve for them (see Figure 1.10). After my MRes and until July 2007, I would send her an email every fortnight with the output of the tool, as shown in Figure 4.2, *i.e.* rank-ordered lists of positive and negative PubMed identifiers, along with links to the corresponding PubMed page. Bickmore would study the list of positives and check the very top of the list of negatives for FNs. She estimated the tool decreased her reading time by a factor of 10.

Since July 2007, Bickmore can use the interface I developed during my PhD to upload PubCrawler's results and compute the lists of positives and negatives by herself as the MRes tool has been incorporated into the final interface. Moreover, she can then directly launch the IE on the full-text paper (work achieved during my PhD) by simply clicking on the PubMed identifiers in the lists (see Section 5.1.3).

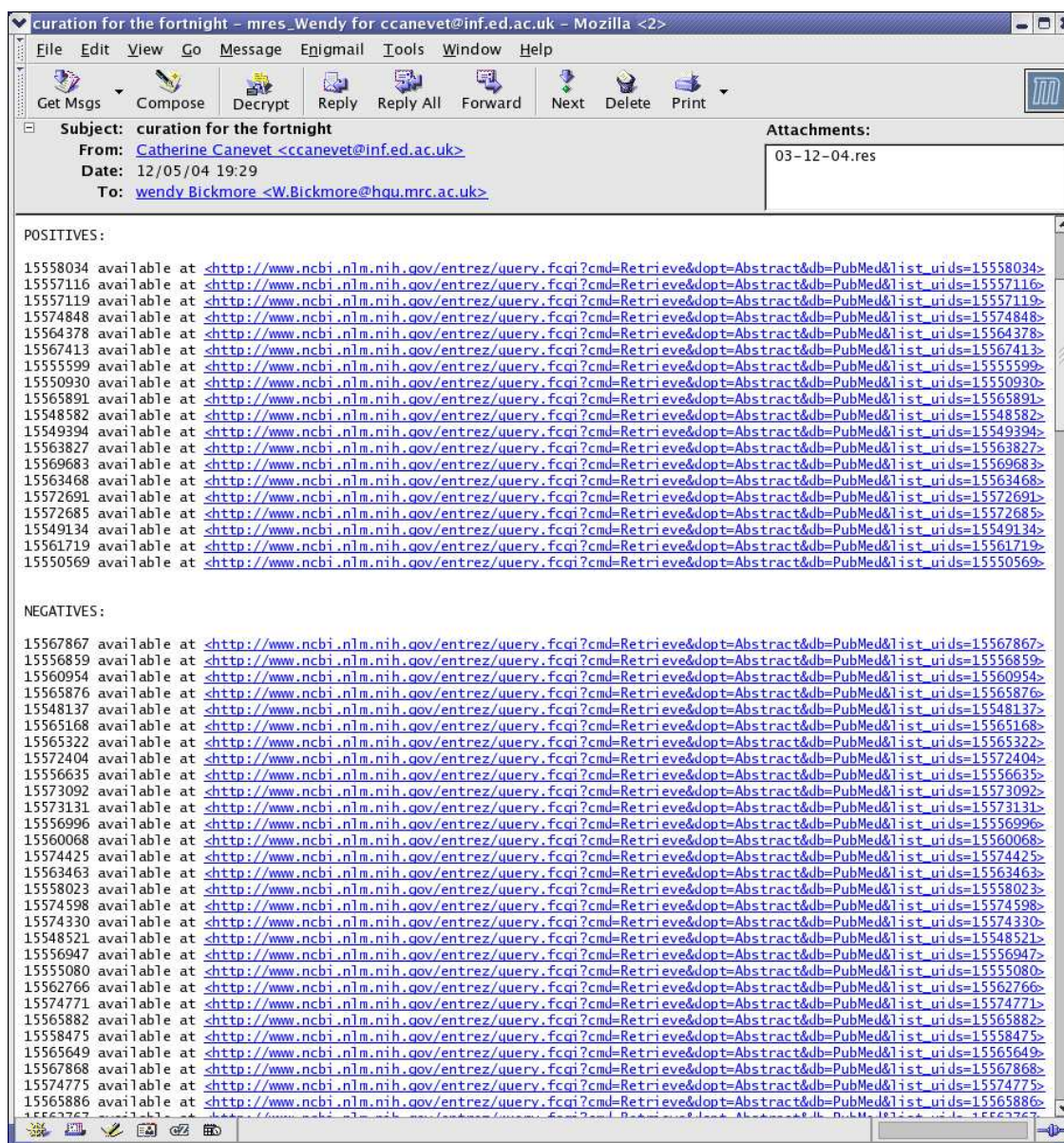


Figure 4.2: Example of email sent to curator every fortnight. Each line starts with a PMID, followed by its clickable Entrez URL. This output contains less FPs than PubCrawler's alerts and therefore saves time to the curator.

4.1.2 Format issues

The annotation tool I have developed deals with all articles, working with abstracts when full text is not accessible. While abstracts are, in most cases, easily accessible through central collections such as PubMed [Pub], the availability of full-text papers is more sparse. Indeed, some articles may be obtainable via publishers' Websites, whereas others would be through PubMed Central or from local repositories.

Sometimes, the full-text articles are simply not freely available online and require paid subscriptions to access them. (Advent of "open-access" publications should reduce this problems in the future.) Sometimes they will only be available in PDF format rather than HTML format. When full text is available online, owing to existing problems with analysing PDF, the tool has been developed for freely-available or locally-licensed articles in HTML format.

To sum up, in this thesis work, I am only dealing with full-text papers available in HTML format, or abstracts if full text is not available in HTML (see Chapter 5). The following paragraphs discuss the reasons why.

The FlySlip project¹ has developed an interface to help the annotation process of scientific papers. In a talk on FlySlip, Nikiforos Karamanis noted that the project used a commercial tool applied to the output of Optical Character Recognition (OCR) in order to be able to deal with PDF papers. However, he admitted that parts of the text were getting lost through the conversion. Rather than working on correcting errors from PDF conversion tools, he affirmed it makes more sense to contribute towards the work being done to generate SciXML structured text from PDF documents. In [Lew07], Ian Lewin explains that SciXML will probably become the common markup language of choice for scientific publications and that this should help the interoperability of text-mining efforts.

To conclude, I decided not to work on full-text papers in PDF format for three different reasons. Good PDF conversion was only available using commercial tools. PDF conversion, no matter how good, loses text. In addition, it seems now is not the time to investigate ways of making free PDF conversion better as, hopefully, SciXML will make a breakthrough and will be more and more used for scientific papers. SciXML came along too late to be an integral component of my PhD research.

4.1.3 Abstracts vs. Full Text

Beyond format issues and availability, why is it preferable to work on full-text publications rather than on abstracts only? Inevitably, a full-text article does provide more information than an abstract. In [SWS⁺04], the authors explain that although information density is higher in the abstract, information content is higher in full text. Moreover, they argue that for IE tools that

¹http://www.cl.cam.ac.uk/nk304/Project_Index/index.html

can distinguish relevant sentences from irrelevant ones in full text, information density should not be of concern and conclude full text should be used for text mining.

The only reason why Professor Wendy Bickmore would extract information from abstracts rather than full texts in the past was that it would have been too time-consuming. However, since the Curator System Interface (described in Chapter 5) was put in use, she has benefited from extracting information located in abstracts as well as other sections.

This thesis claims it is only with respect to the full text that it is possible to distinguish major localisation relations the authors are trying to put across from minor ones. Indeed, this is achieved through a study of the frequency of occurrence of all the localisation relations within an article (see Section 4.3).

In order to support this argument, I examined two papers: [PPRMV04] from my training corpus and [SSE⁺98] from the testing corpus (see Section 3.3 for corpora). Figures 4.3 and 4.4 show the PubMed entries for these two articles.

The main facts in [PPRMV04] are protein BIG1 in the nucleus and protein BIG1 in the nucleolus. Similarly, the main facts in [SSE⁺98] are protein DEDD in the nucleus and protein DEDD in the nucleolus. I first looked at these four main facts, where and how often they appeared in the text. For the record, both abstracts contain a total of 10 sentences each.

Before looking at the results in Table 4.1, let us comment on the importance of major facts in a paper. Other facts can be, and indeed are, extracted from an article to update a database. However, in order to make sure we are not missing any important information and in order to provide the curator with a list of facts ordered by their importance, we need to focus on the number of times a piece of information is stated. As mentioned earlier, the more authors repeat a certain piece of information the more we can be confident they are trying to put this message across in particular.

	[PPRMV04]	[PPRMV04]	[SSE ⁺ 98]	[SSE ⁺ 98]
	BIG1	BIG1	DEDD	DEDD
	in nucleus	in nucleolus	in nucleus	in nucleolus
Abstract	6	2	3	2
Rest	37	7	30	15

Table 4.1: Number of sentences covering the most important facts of two articles from my corpora in the abstract alone or the rest of the paper. BIG1 (also called Brefeldin A-inhibited GEP 1) and DEDD (also named DEDPRO1) are two nuclear proteins.

In total for these two papers and their two main facts each, 13 sentences were from the abstract and 89 were retrieved from other sections in the publications. The conclusion is both most important localisation relations for both papers I manually assessed were found in the

Nuclear localization and molecular partners of BIG1, a brefeldin A-inhibited guanine nucleotide-exchange protein for ADP-ribosylation factors.

Padilla PI, Pacheco-Rodriguez G, Moss J, Vaughan M.

Pulmonary-Critical Care Medicine Branch, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, MD 20892, USA. padillap@nhlbi.nih.gov

Brefeldin A-inhibited guanine nucleotide-exchange protein 1 (BIG1) is an approximately 200-kDa brefeldin A-inhibited guanine nucleotide-exchange protein that preferentially activates ADP-ribosylation factor 1 (ARF1) and ARF3. BIG1 was found in cytosol in a multiprotein complex with a similar ARF-activating protein, BIG2, which is also an A kinase-anchoring protein. In HepG2 cells growing with serum, BIG1 was primarily cytosolic and Golgi-associated. After incubation overnight without serum, a large fraction of endogenous BIG1 was in the nuclei. By confocal immunofluorescence microscopy, BIG1 was localized with nucleoporin p62 at the nuclear envelope (probably during nucleocytoplasmic transport) and also in nucleoli, clearly visible against the less concentrated overall matrix staining. BIG1 was also identified by Western blot analyses in purified subnuclear fractions (e.g., nucleoli and nuclear matrix). Antibodies against BIG1, nucleoporin, or nucleolin coimmunoprecipitated the other two proteins from purified nuclei. In contrast, BIG2 was not associated with nuclear BIG1. Also of note, ARF was never detected among proteins precipitated from purified nuclei by anti-BIG1 antibodies, although microscopically the two proteins do appear sometimes to be colocalized in the nucleus. These data are consistent with independent intracellular movements and actions of BIG1 and BIG2, and they are also evidence of the participation of BIG1 in both Golgi and nuclear functions.

Related Articles

- ▶ Identification and localization of two brefeldin A-inhibited guanine n [Proc Natl Acad Sci U S A. 2000]
- ▶ Purification and cloning of a brefeldin A-inhibited guanine nucleotide-exchange prc [J Biol Chem. 1999]
- ▶ Interaction of FK506-binding protein 13 with brefeldin A-inhibited [Proc Natl Acad Sci U S A. 2003]
- ▶ **Review** Activation of toxin ADP-ribosyltransferases by [Mol Cell Biochem. 1999]
- ▶ **Review** Arf, Sec7 and Brefeldin A: a model towards the therapeutic ir [Biochem Soc Trans. 2005]

» See Reviews... » See All...

Patient Drug Information

- ▶ Tacrolimus (Prograf®) Tacrolimus is used along with other medications to prevent rejection (attack of a transplanted organ by the immune system of

» read more ...

Figure 4.3: PubMed entry showing the abstract of [PPRMV04] and its sentences identified as relevant by the curator highlighted in yellow.

DEDD, a novel death effector domain-containing pro...[EMBO J. 1998] - PubMed Result - Mozilla Firefox

http://www.ncbi.nlm.nih.gov/sites/entrez

Getting Started Latest Headlines

NCBI PubMed A service of the U.S. National Library of Medicine and the National Institutes of Health

My NCBI [Sign In] [Register]

All Databases PubMed Nucleotide Protein Genome Structure OMIM PMC Journals

Search PubMed for [] Go Clear Advanced Search (beta) Save Search

Limits Preview/Index History Clipboard Details

Display AbstractPlus Show 20 Sort By Send to

All: 1 Review: 0

We found 1 article:

1: EMBO J. 1998 Oct 15;17(20):5974-86.

THE EMBO JOURNAL FULL TEXT FREE FREE full text article in PubMed Central Links

DEDD, a novel death effector domain-containing protein, targeted to the nucleolus.

Stegh AH, Schickling O, Ehret A, Scaffidi C, Peterhänsel C, Hofmann TG, Grummt I, Krammer PH, Peter ME.

Tumor Immunology Program, German Cancer Research Center, Im Neuenheimer Feld 280, D-69120 Heidelberg, Germany.

The CD95 signaling pathway comprises proteins that contain one or two death effector domains (DED), such as FADD/Mort1 or caspase-8. Here we describe a novel 37 kDa protein, DEDD, that contains an N-terminal DED. DEDD is highly conserved between human and mouse (98.7% identity) and is ubiquitously expressed. Overexpression of DEDD in 293T cells induced weak apoptosis, mainly through its DED by which it interacts with FADD and caspase-8. **Endogenous DEDD was found in the cytoplasm and translocated into the nucleus upon stimulation of CD95. Immunocytological studies revealed that overexpressed DEDD directly translocated into the nucleus, where it co-localizes in the nucleolus with UBF, a basal factor required for RNA polymerase I transcription. Consistent with its nuclear localization, DEDD contains two nuclear localization signals and the C-terminal part shares sequence homology with histones.** Recombinant DEDD binds to both DNA and reconstituted mononucleosomes and inhibits transcription in a reconstituted in vitro system. The results suggest that DEDD is a final target of a chain of events by which the CD95-induced apoptotic signal is transferred into the nucleolus to shut off cellular biosynthetic activities.

PMD: 9774341 [PubMed - indexed for MEDLINE] PMID: PMC1170924

Display AbstractPlus Show 20 Sort By Send to

Done

Related Articles

- Nuclear localization of DEDD leads to caspase-6 activation through its death e [Cell Death Differ. 2001]
- DEDD and DEDD2 associate with caspase-8/10 and signal cell death. [Oncogene. 2003]
- Identification and characterization of DEDD2, a death effector domain-containing [J Biol Chem. 2002]
- Review** The death effector domain protein family: regulators of cellular homeostasis [Nat Immunol. 2003]
- Review** FADD/MORT1, a signal transducer that can promote cell death [Int J Biochem Cell Biol. 1999]

> See Reviews... > See All...

Figure 4.4: PubMed entry showing the abstract of [SSE⁺98] and its sentences identified as relevant by the curator highlighted in yellow.

abstract and more than once too. However, other localisation relations were also found in the abstract, as shown below, making it impossible to identify the main facts of the paper simply based on the abstract.

[PPRMV04]'s abstract contain another 8 localisation relations:

- nucleoporin p62 in nucleus (2 sentences)
- nucleoporin p62 in nucleolus (1 sentence)
- nucleolin in nucleus (1 sentence)
- BIG2 in nucleus (2 sentences)
- BIG1 in Golgi (1 sentence)²
- BIG2 in Golgi (1 sentence)

These occur as often or not much less often than the two main facts. Similarly [SSE⁺98]'s abstract contain another 3 localisation relations:

- DEDD in cytoplasm (1 sentence)
- UBF in nucleus (1 sentence)
- UBF in nucleolus (1 sentence)

These results suggest that abstracts do not yield to as simple a method as frequency of occurrence for determining the main facts of a paper. It is not that the facts listed above are not interesting for extraction, it is simply that if they are not the main facts the authors are trying to put across. This means these facts are most probably not new and original research, therefore they might already be present in the NPD.

4.2 Relevance detection

When going through free text and checking for particular information, rather than a story or an argument, not all parts of the text are equally relevant. For simple information, such as a two-place relation (a protein and its localisation), the locus of that information may be as small as a single sentence or even a single clause or phrase. More complex information might only be conveyed in a sequence of sentences, or the text as a whole. However, with simple information, some sentences may convey the relation and others not. Therefore, the first are relevant, the others not.

²BIG1 in cytosol was not captured as “cytosol” is not listed in the compartment gazetteer.

In order to automatically highlight sentences in full-text papers that are relevant to the localisation of nuclear proteins, I chose to automatically go through the free text and check for a set of different features a sentence carries. Chapter 3 compared the performance of three different types of method for retrieving sentences of interest from a document:

- supervised Machine Learning (see section 3.4),
- rule-based with BioIE (see section 3.5),
- and unsupervised Machine Learning with Infomap (see section 3.6).

Using DT, the method that performed best, the interface highlights sentences that are considered relevant to the sublocalisation of nuclear proteins (see Section 5.1.4).

4.3 Redundancy detection

In biomedical articles, the main piece of information is often repeated several times throughout the text. For example, simply in the abstract of [SWJ⁺00] from the training corpus, the authors state fibrillarin is in the nucleolus and in Cajal Bodies (abbreviated CBs, see Section 1.2.1) several times:

- *“presence of nucleolar proteins such as fibrillarin in CBs”*
- *“structural domains of fibrillarin are required for correct intranuclear localization of fibrillarin to nucleoli and CBs”*
- *“appear to target fibrillarin, respectively, to the nucleolar transcription centers and CBs”*

As discussed in Section 3.7, the same piece of information may be stated several times in an article for different purposes (*e.g.*, as speculation, as claim, as given, *etc.*). Therefore, the more sentences refer to a particular localisation relation, the more this localisation relation is likely to be important within the document.

This section looks at redundancy of information within a document, and indeed grouping sentences that refer to the same kind of information together. The next section (Section 4.4) will look at detecting novelty to the NPD.

In order to group similar sentences together, I decided to tag each sentence with all the localisation relations (a protein and its localisation) it represented. Figure 4.5 illustrates how this is an extra step towards domain-specific annotation. Sentences may manifest zero, one or more than one localisation relation(s). The number of instances for each localisation relation can be counted, and determine what localisation relations are important to a paper.

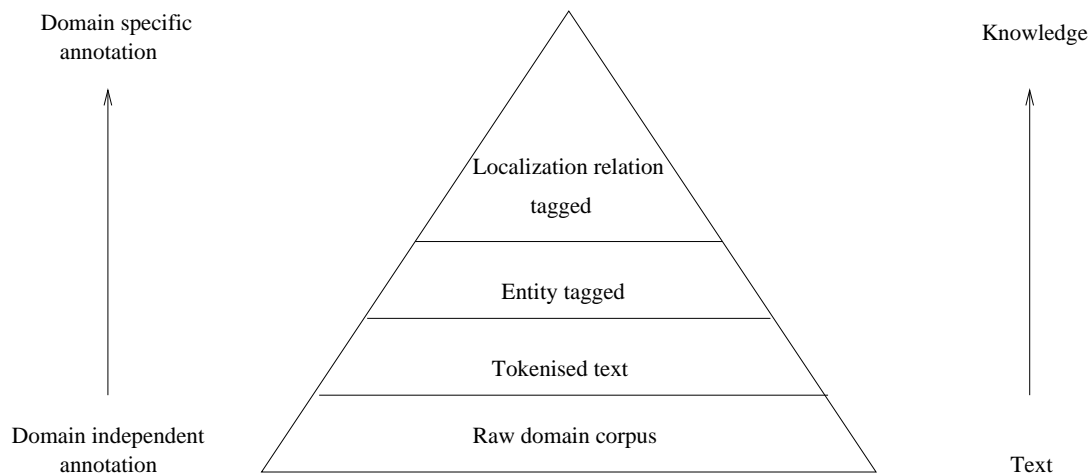


Figure 4.5: Each extra layer of tagging allows us to transform raw text into a domain-specific annotation-rich corpus. The layers displayed in the triangle represent the steps taken in this work.

The group of sentences manifesting a single relation can be identified. My interface (see Chapter 5) then displays a Table containing the list of relations found in the article ordered by importance (or size of the group). For each localisation relation, the curator can have access to all the sentences that manifest that same fact in context, as they are highlighted in the full-text document.

In order to tag sentences with localisation relations, I decided to use a locally developed tool called *lxtransduce* [GMT06]. Unlike my previous work, *lxtransduce* uses XML and adds markup containing NE tags (like the C&C tagger to some extent, see Section 3.4.2) to the text. I will first introduce the tool itself, explain how its output can be used to tag sentences with localisation relations, then show the results it achieved and discuss them.

4.3.1 Introduction to *lxtransduce*

TTT2 [GMT06] is the second release of a Text Tokenisation Tool written by the Language Technology Group (LTG, <http://www.ltg.ed.ac.uk/>) at the University of Edinburgh. The system tokenises text and offers the possibility of adding XML markup tags at specified levels of the text. The main component of TTT2, “*lxtransduce*”, is a transduction program that works on XML files. It lets users write their own lexicons (lists of terms of interest) and grammars of rules that specify where the tool should place its markup tags signalling to the users the presence of NEs in the text.

I used TTT2-bundled utilities to convert my training corpus into the XML format required by *lxtransduce*. I converted my original gazetteers (see Section 3.4.1) to create lexicons re-

specting the TTT2 format. I also created a grammar for Ixtransduce to tag my corpus with NEs of interest. Ixtransduce provides a rule (the `<seq>` element) that allows users to ask for some NE to appear one after the other (in sequence). Using this specific rule, I was able to provide Ixtransduce with an equivalent to rule A (see Section 3.4.1). Ixtransduce can then annotate each sentence with NE tags. At this stage, it is possible to group sentences that manifest the same localisation relation by analysing the tags Ixtransduce has associated them with. The method used to perform this grouping is described below.

4.3.2 Grouping sentences manifesting the same localisation relations

The output from Ixtransduce is an XML file with XML tags signalling the presence of NEs. Some sentences contain a single protein tag as well as a single location tag, and therefore will be labelled with a single localisation relation. Some sentences contain a single protein tag and multiple compartment tags, or a single compartment tag and multiple protein tags. Some sentences contain multiple tags for both.

I have developed a perl script that parses the output file of Ixtransduce and looks for XML tags signalling the presence of NEs. I wrote the program in such a way that any protein name and any compartment name in a sentence would be paired up as a potential localisation relation in that sentence. For example, if an author writes that “Protein P1 is located in compartment C1 whereas protein P2 is found in compartment C2”, my program says P1-C2 and P2-C1 are valid localisation relations for that sentence, which is not the case.

The program can be improved by only matching protein and compartment names that are closest to each other in the sentence distance-wise (see Section 4.3.4) or, even better, by using parsing or chunking as a way of actually seeing which proteins and compartments do go together. I also considered dealing with negation, but realised that as I am looking for the localisation relations that are mentioned most often in an article, I either would have a lot of tokens of the same relation appearing in a negative context or I would have some positive and some negative tokens. Both cases would probably be of interest to the curator.

4.3.3 Results

Section 3.3 presented the training corpus (composed of 14 publications) and the testing corpus (composed of 3 articles). Results show the evaluation of the lists of sentences retrieved, one for each `<protein, compartment>` pair found in an article. Precision in Table 4.2 corresponds to an average for all the different lists obtained. The precision for one particular list relates the number of sentences that truly convey the localisation relation associated to that list to the total number of sentences retrieved by the algorithm for that list or `<protein, compartment>` pair.

Corpora	training	testing	Both
Precision	0.532	0.635	0.538

Table 4.2: Grouping precision on training and testing corpora

Recall would then be the number of sentences conveying the correct localisation relation to the total number of sentences, which should have been retrieved from that particular localisation relation, and is estimated very high. I have not calculated it as it would be too time-consuming. It involves reading through both corpora and checking for any FNs, which would include, for example, anaphora as in the following sequence of sentences from [DMO00]:

“In early anaphase *protein B23* and *fibrillarin* had nearly identical patterns of localisation (Fig 8, A-C). **Both proteins** were localized in perichromosomal regions (PRs) and NDF³ as well as being distributed generally in the cell plasm.”

4.3.4 Discussion

When analysing the results of the algorithm explained in Section 4.3.2, I found that two types of error predominated. The first one was expected because of the way the algorithm was written, as explained in the previous section.

“The nucleolar localization of DEDD suggests that it may affect some important nucleolar functions, and therefore interferes with ribosome biosynthesis.”

This sentence was tagged as containing two localisation relations. According to Bickmore’s manual annotation, the first one (“DEDD in nucleolus”) is correct whereas the second one (“DEDD in ribosome”) is not.

The second type of predominating error is encountered when a sentence indicates what would happen in conditions that are not natural.

“Since DEDD was found to associate with FADD and caspase-8 in vitro, we tested whether under conditions with significant amounts of DEDD in the cytoplasm an association with endogenous FADD and/or caspase-8 could be found in vivo.”

This sentence was tagged as containing two localisation relations. “DEDD in cytoplasm” is not an acceptable localisation relation, as the sentence mentions it in an experiment context. Also “FADD in cytoplasm” is incorrect and falls into the first type of predominating error. The pair “caspase-8 in cytoplasm” was not retrieved, as caspase is not part of the protein names lexicon.

³NDF stands for Nucleolus-Derived Foci.

The first type of error could be dealt with by improving my script. For example, a distance-based algorithm could resolve the problems mentioned above (“DEDD in ribosome” and “FADD in cytoplasm”). However, it could also miss out on some sentences such as

“DEDD, a novel death effector domain-containing protein, targeted to the nucleolus”

where the protein name is far away from the compartment name.

As far as the second main source of error is concerned, a filter could be installed in order to avoid sentences that contain keywords like “test”, “in vitro”, “under conditions”, “experiment”. This strategy, too, could have a negative effect on the results, where a sentence such as

“Immunofluorescence microscopy experiments showed that hGAR1, hNOP10, and hNHP2 are localized in the dense fibrillar component of the nucleolus and in Cajal (coiled) bodies.”

would not produce the localisation relations “hGAR1 in DFC” (Dense Fibrillar Component), “hNOP10 in DFC”, “hNHP2 in DFC”, “hGAR1 in nucleolus”, “hNOP10 in nucleolus”, “hNHP2 in nucleolus”, “hGAR1 in CBs”, “hNOP10 in CBs”, “hNHP2 in CBs” any longer.

As this algorithm’s results obtained scores of an acceptable quality, I decided to keep using it without experimenting any further. Further experiments might not have improved the results significantly. Moreover, it would have meant manually assessing the results again, which is a tedious and highly time-consuming task.

4.4 Novelty detection with regards to the NPD

Once different types of information have been detected in a paper, the next step towards assisting the annotation of the NPD is to check whether any of the localisation relations are new or novel to the NPD.

4.4.1 Getting the latest version of the NPD

The latest uploaded version of the NPD is publicly available on <ftp://ftp.hgu.mrc.ac.uk/pub/npd/> as a link from the NPD’s Website indicates. My tool checks this site for the date of the latest update. If it is more recent than the version the tool is currently working with, it downloads it.

Working on the latest “esql_add.txt”, I can look through the different existing Tables in the database and identify the ones I need. The mysummeta Table gives all the protein names and aliases, along with the NPD ID they have been associated with. The mysumprotsublocal Table gives details about the phase of the cell cycle, the localisation and extra information on conditions.

I convert both Tables into separate files. I use the linux commands “sort” and “uniq” to get the files sorted by NPD IDs and get rid of duplicates. The mytsumprotsubnlocal Table actually may have several lines for each NPD ID, so it is important that these lines should be grouped together for the next step.

4.4.2 Checking NPD flat file

Having created these two files sorted by NPD IDs, I can now easily search through them. Using the file containing all the protein names, it is possible to check whether a protein has an entry in the database or not.

If the protein does have an entry in the database, it is then possible to check the file with all the localisation information to see whether the localisation relation found in the article is already registered in the NPD, by looking for the protein NPD ID and checking whether the compartment the article mentions is in any of the NPD entries for that particular protein NPD ID.

For example, if an article talks about protein DDX18 in nucleolus, I check the protein file for DDX18 and find the ID “1NP00037”. Then I search the localisation file with 1NP00037 and get: “1NP00037; Interphase; nucleolus; NULL⁴”. As nucleolus appears in this entry, the interface would say that the protein is in the NPD and the protein localisation is too.

4.4.3 Checking aliases

The NPD might use different protein or compartment names from the paper in current automatic annotation. Therefore it is important to check for aliases.

Indeed, a paper can talk about a compartment such as the NDF but the NPD might only have information about the protein in the nucleolus derived foci. Without checking for aliases, the program would return the protein localisation is not present in the NPD, when in fact it is, only under a different name.

A subroutine in my script deals with this issue using lists of aliases computed previously, as explained in Section 3.4.1.

4.4.4 Results and assessment

I tested the tool on 10 papers from the training corpus ([PPRMV04, SCD⁺02, SWJ⁺00, CDG⁺03, CSK98, DMO00, DO98, KZCJ02, LS02, MKC⁺05]). For the 62 distinct proteins mentioned in these 10 papers,

- 36 were correctly matched to their NPD entry (TPs),

⁴The keyword NULL indicates there was no value entered for a particular field.

- 11 were incorrectly matched (FPs),
- 10 were not in the database and
- 6 were incorrectly identified as proteins in the first place.

According to the TPs and FPs, the tool achieves a precision of 0.766. In the 10 papers examined here, no FNs were found which gives a recall of 1 and an F-score of 0.867.

However, it is likely that some papers will contain proteins that are present in the NPD but which are talked about using an alias that is not entered in the NPD. Mistakes due to unknown aliases (as well as how often the ftp site is updated with a more recent version of the NPD) will incur FNs.

It would be possible to generate for each article a local list of proteins and their aliases. This way, when a sentence refers to an alias that is absent from the NPD, one could check whether any other local aliases to the paper are actually present in the NPD. Previous work in finding and resolving local aliases include:

- In 2002, [CSA02] implemented a statistical learning algorithm that relies on a training corpus of expert-annotated abbreviations to automatically produce identification rules to then match abbreviations with their expansions in text. Their method was tested on the Medstract corpus and performed with 88% precision and 83% recall.
- In 2004, [EYD04] built a dictionary-based system, called ProtScan, which identifies mammalian protein names in MEDLINE records with 98% precision and 88% recall. They also tested their approach on abstracts only (without MEDLINE fields) and achieved 98.5% precision and 84% recall.
- At the EBI, in 2005 [GKRS05] automatically created a dictionary of <abbreviation, sense> pairs by mining all MEDLINE abstracts from 1965 to 2004. They use this resource to resolve abbreviations. In some cases, abbreviations can have several senses, which can only be distinguished by studying their context. The authors train an SVM classifier (see Section 2.2.10) to associate a large number of words with a context for each sense. They obtained 5-fold cross-validation results of 98.9% precision and 98.2% recall.

Looking at entities that were incorrectly identified as proteins in the 10 tested papers, we found “FRAP” and “methylase”. There is an entry in the NPD for the FRAP protein:

- main name: FRAP
- aliases: mTOR, RAFT1, FRAP2, target of rapamycin, TOR

Unfortunately, FRAP is also an acronym for a technique in biology called Fluorescence Recovery After Photobleaching. Authors tend to mention this common technique in its abbreviated form. This is therefore quite a recurrent problem.

There is also an entry in the NPD that contains the word “methylase” (which is a collective name for a particular type of enzyme):

- main name: DNMT1
- aliases: DNA methyltransferase 1, maintenance methylase, DNA (cytosine-5-)-methyltransferase 1.

However, in the sentence “*As a fibrillar homolog in Methanococcus jannaschii has been shown to contain a methylase fold (Wang et al. 2000), this enzymatic activity of fibrillar may also be required in CBs.*” the authors of the publication [SWJ⁺00] are not referring to DNMT1.

False positives can be explained by looking at the way the script checks for novelty with regard to the NPD. When looking through the protein names file, the script stops at the first corresponding name it finds. However, for some proteins, their names can be followed by a space and a number.

For example, “nucleoporin” has 12 entries in the NPD:

- F9/17A4, F9/17D3, Nucl pore complex prot, NUP153, NUCLEOPORIN 153
- NUP62, nucleoporin 62
- nucleoporin 358, NUP358, RANBP2, RBP2
- nucleoporin 107
- NUP98, Nucleoporin 98, Nup98-Nup96 precursor
- NUP96, Nup98-Nup96 precursor, Nucleoporin 96
- NUP155, nucleoporin 155
- NUP188, nucleoporin 188
- NUP93, nucleoporin 93, KIAA0095, NIC96
- NUP205, nucleoporin 205, KIAA0255
- NUP54, nucleoporin 54
- NUP58, nucleoporin 58

If an article talks about NUP62 the tool will get the correct protein in this file. If an article talks about nucleoporin 62, the number will not be recorded and the tool will get the first “nucleoporin” entry it can find in the file. There are currently 2159 different proteins in the NPD: 1973 of them have numbered versions; only 186 do not. Using a protein’s canonical form could be a good way of avoiding this problem. Other ways of solving this problem could involve dealing with two or several tokens per NE so as to capture the full name of a protein when it contains one or more white spaces, or working with concept recognition (see Section 3.7) rather than NER.

4.5 Highlighting sentences related to a localisation relation using colour codes

The final tool (see Chapter 5) can highlight, for each type of information, all the sentences related to that particular type of information using a colour code.

The colour code will allow the user to distinguish:

- sentences that are related to a particular type of information in pink from
- sentences that are related to a particular type of information AND were also picked up by my relevance detector (see Section 4.2), in red.

Ultimately, the results combining the pink and red sentences should equal the ones reported in Section 4.3, as the tool is only using those results and colour coding them for the user interface. In order to compare the two different categories of sentences, I calculated precision based on 4 random papers from the training corpus ([PPRMV04, SCD⁺02, SWJ⁺00, CDG⁺03]). (Unfortunately, calculating precision for the whole of the training corpus would have been a too time-consuming task.) When looking at all the different types of information and all the sentences highlighted for each of them, I found these 4 articles contained:

- 34 pink sentences,
- 145 red sentences,
- 179 sentences in total.

Colour	Pink	Red	Both
Precision	0.529	0.786	0.737

Table 4.3: Highlighting sentences: precision based on 4 papers from the training corpus

Recall was not calculated as I only evaluated highlighted sentences. As explained in Section 4.3.3, it would be difficult to go through all the sentences to find false negatives that were missed by my system. Section 4.3.4 presented a discussion over the error analysis.

Table 4.3 shows a higher precision is obtained for the red sentences making them - as intended - more trustworthy for the curator.

4.6 Summary

This chapter covered all the different features the Curator System Interface uses. The document retrieval is performed by the tool I developed during my MRes with a precision of 0.7, a recall of 0.636 and an F-score of 0.667. Decision Tree performed best for the relevance detector and achieved a precision of 0.569, a recall of 0.652 and an F-score of 0.608. The redundancy detector achieved a precision of 0.538 over both training and testing corpora. Error analyses were carried out and used to point out areas where system performance could be improved. The novelty detector achieved a precision of 0.766 over ten articles from the training corpus. The highlighting of sentences related to a localisation relation achieved a precision of 0.737 based on 4 articles from the training corpus. The next chapter introduces the graphical user interface of the NPD Curator System Interface and evaluates the system as a whole on various levels.

Chapter 5

The NPD Curator System Interface for annotation assistance

This chapter presents the tool [CWB08] developed to test and support the claims made in this thesis. The first part of the chapter introduces all the different pages the interface offers. The second part gives an evaluation of the tool.

5.1 The Curator-System Interface

This section shows the interface the curator of the database is presented with in order to help her update the database.

5.1.1 Homepage

The homepage of the NPD curator system interface (see Figure 5.1) allows the user to:

- launch the tool working on a specific PMID,
- post comments in order to log feedback.

5.1.2 Documents retrieval

The “Documents retrieval” page and the “Retrieval results” page incorporate my Masters by Research [Can04] tool (see Section 4.1) into the NPD Curator System Interface. The “Documents retrieval” page (see Figure 5.2(a)) allows the user to load a file containing results from PubCrawler (see Section 4.1.1 for more details).

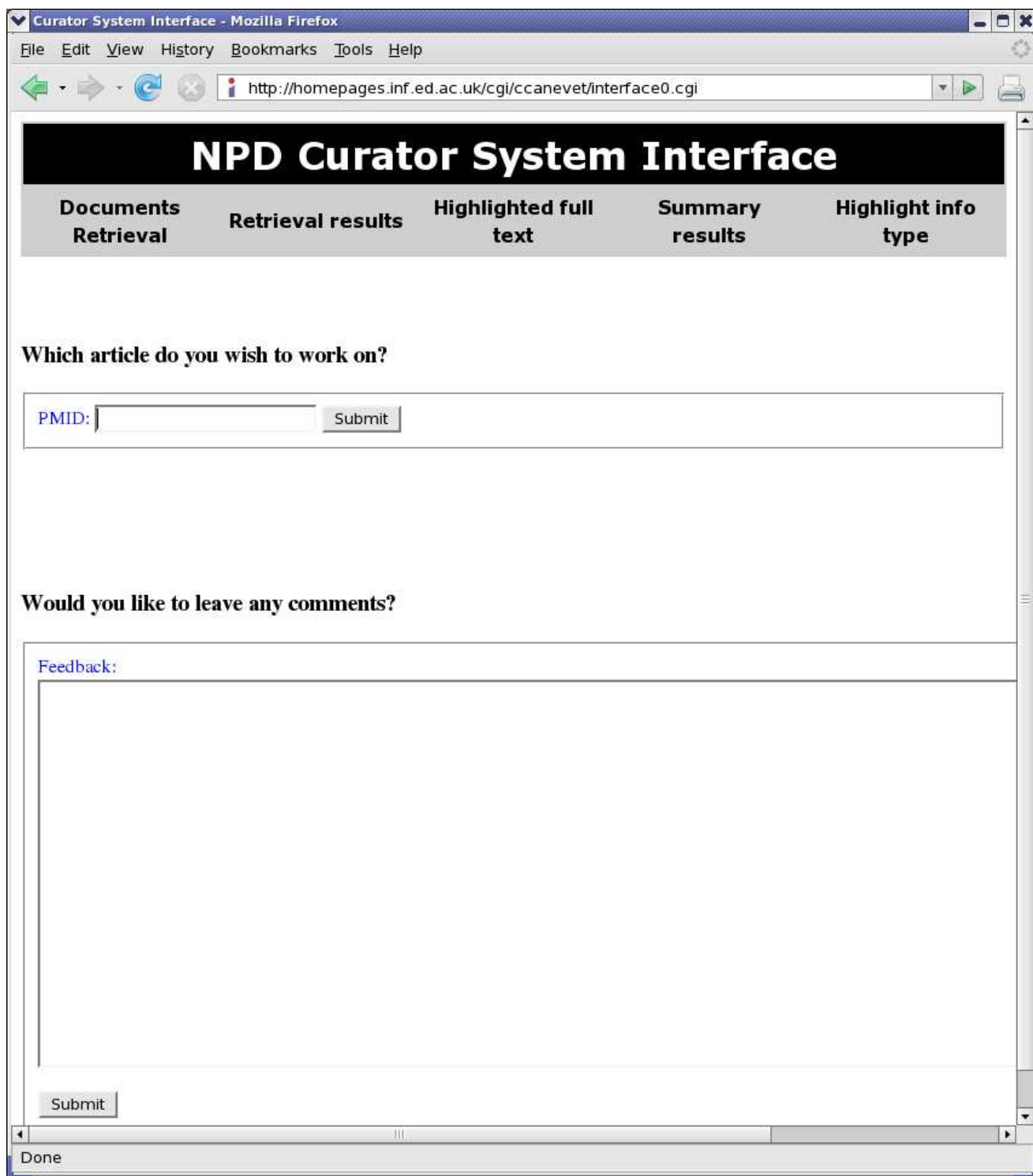
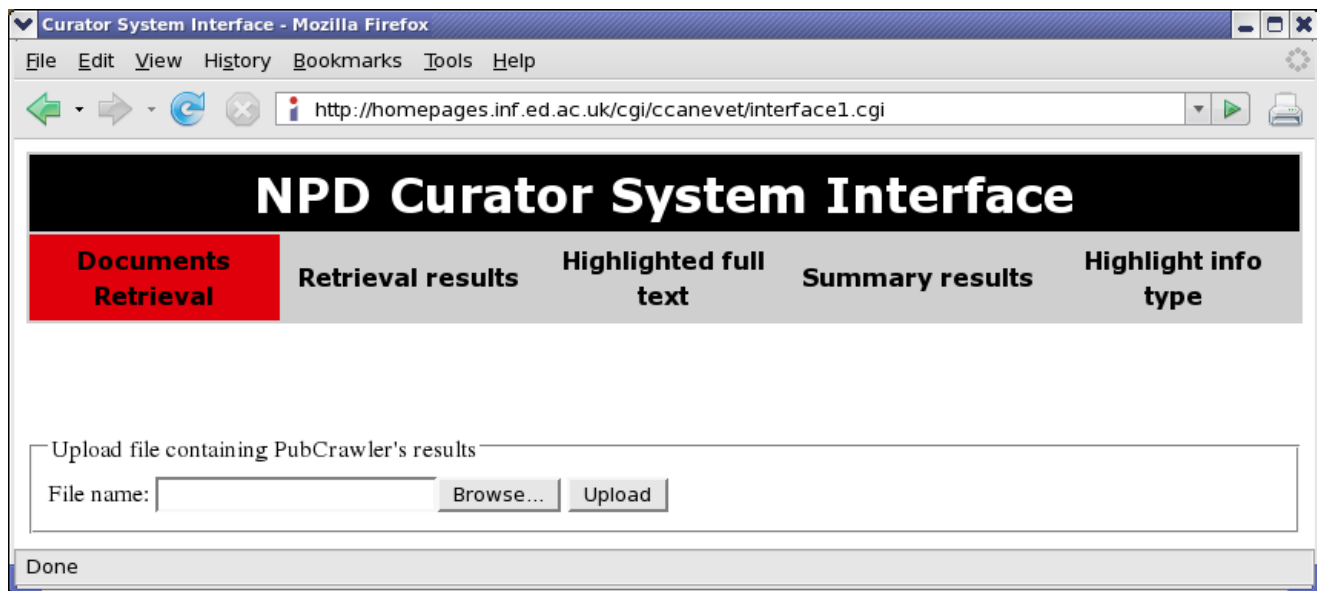
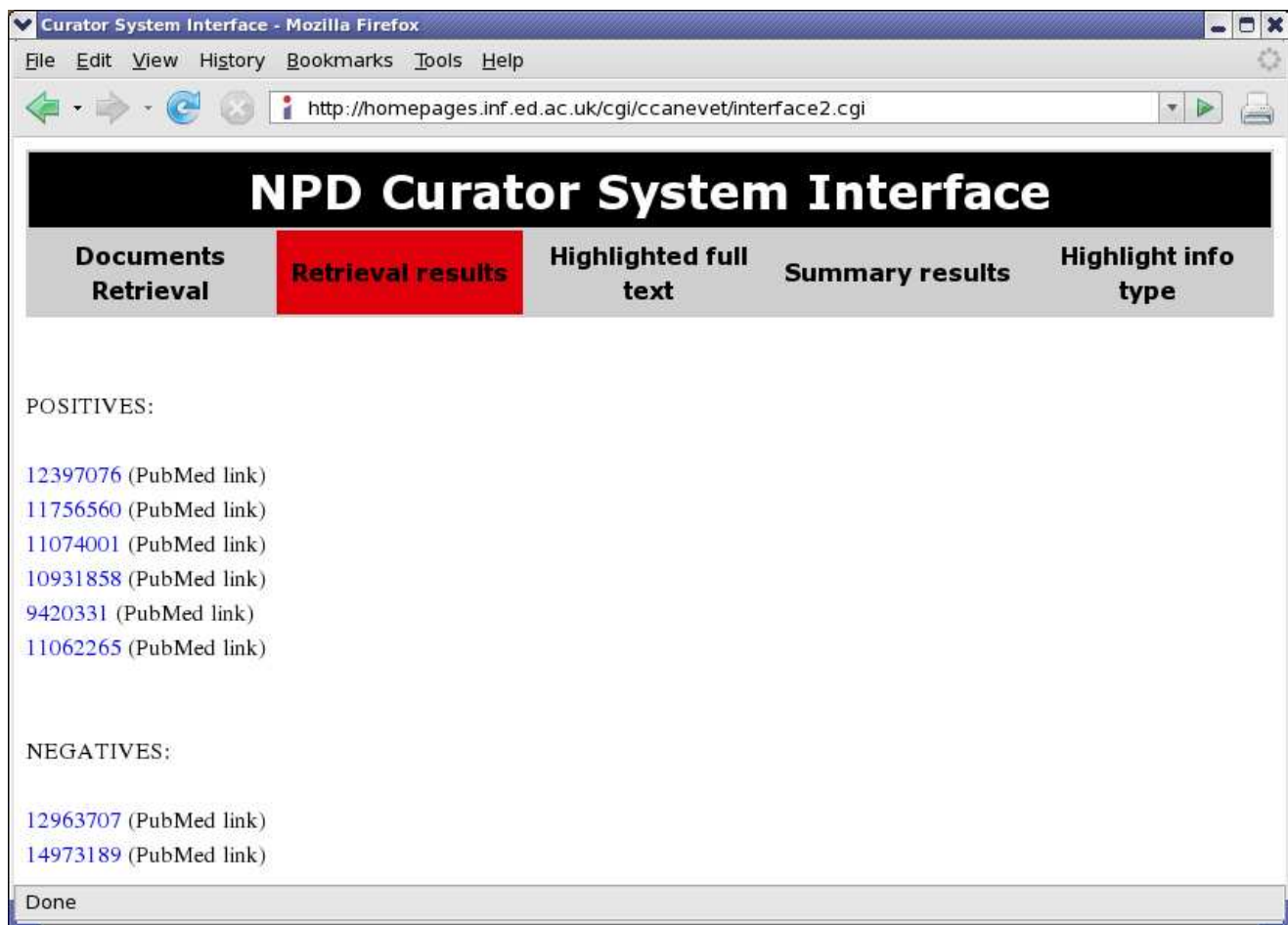


Figure 5.1: Screenshot of the NPD Curator System Interface Homepage



(a) Screenshot of the “Documents retrieval” page



(b) Screenshot of the “Retrieval results” page

Figure 5.2: Screenshots showing the documents retrieval tool (developed for my MRes [Can04]) integrated in the final interface

5.1.3 Retrieval results

The “Retrieval results” page (see Figure 5.2(b)) provides the user with the results my MRes tool returns. From this page, the user can click on:

- a PMID to launch the tool working on that particular paper,
- a “PubMed link” to be redirected to the PubMed page for that particular paper.

5.1.4 Highlighted full text

The “Highlighted full text” page shows:

- highlighted full text (see Figure 5.4) if full text was found online in HTML format,
- highlighted abstract otherwise (see Figure 5.3).

Relevant sentences are highlighted based on the results of a supervised method presented in Chapter 3.

5.1.5 Summary results

The “Summary results” page displays a Table summarising the localisation relations found in the article. Groups of sentences manifesting a single relation are identified using the algorithm described in Section 4.3. The number of instances for each localisation relation is counted and determines what localisation relations are important to a paper. The interface displays the list of relations identified in the article ordered by importance or size of the group (see Figure 5.5).

For each localisation relation found, the Table - using the algorithm described in Section 4.4 - also provides the user with information with respect to:

- whether the NPD contains an entry for this protein (“Protein” column),
- if it does, the Table displays details of the NPD’s current content for that particular entry (“Currently in NPD” column), and
- whether the NPD already has a record for this localisation relation (“Protein localiza-tion” column) accordingly.

5.1.6 Highlight info type

For each localisation relation shown in the Table (see “Summary results” page, Section 5.1.5), the curator can have access to all the sentences related to it. Indeed, clicking on the numbers of sentences found (“number of instances” column in the Table, see Figure 5.5) launches the last page of the tool where these sentences are highlighted in context. Figure 5.6 and its caption explain the colour code used on this “Highlight info type” page.

NPD Curator System Interface

Documents Retrieval Retrieval results **Highlighted full text** Summary results Highlight info type

UBF binding in vivo is not restricted to regulatory sequences within the vertebrate ribosomal DNA repeat.

PMID: [11756560](#)
 First author: [O'Sullivan AC](#)
 Last author: [McStay B](#)
 Journal: [Mol Cell Biol](#)
 Year: [2002](#)

UBF binding in vivo is not restricted to regulatory sequences within the vertebrate ribosomal DNA repeat .

The HMG box containing protein UBF binds to the promoter of vertebrate ribosomal repeats and is required for their transcription by RNA polymerase I in vitro .

UBF can also bind in vitro to a variety of sequences found across the intergenic spacer in *Xenopus* and mammalian ribosomal DNA (rDNA) repeats .

The high abundance of UBF , its colocalization with rDNA in vivo , and its DNA binding characteristics , suggest that it plays a more generalized structural role over the rDNA repeat .

Until now this view has not been supported by any in vivo data .

Here , we utilize chromatin immunoprecipitation from a highly enriched nucleolar chromatin fraction to show for the first time that UBF binding in vivo is not restricted to known regulatory sequences but extends across the entire intergenic spacer and transcribed region of *Xenopus* , human , and mouse rDNA repeats .

These results are consistent with a structural role for UBF at active nucleolar organizer regions in addition to its recognized role in stable transcription complex formation at the promoter .

AD - Biomedical Research Centre , University of Dundee , Ninewells Hospital and Medical School , Dundee DD 1 9SY , United Kingdom .

Done

Figure 5.3: Screenshot of the “Highlighted full text” page (abstract only, no full text available in HTML format). Sentences highlighted in yellow show the sentences classified as relevant by the supervised method presented in Chapter 3.

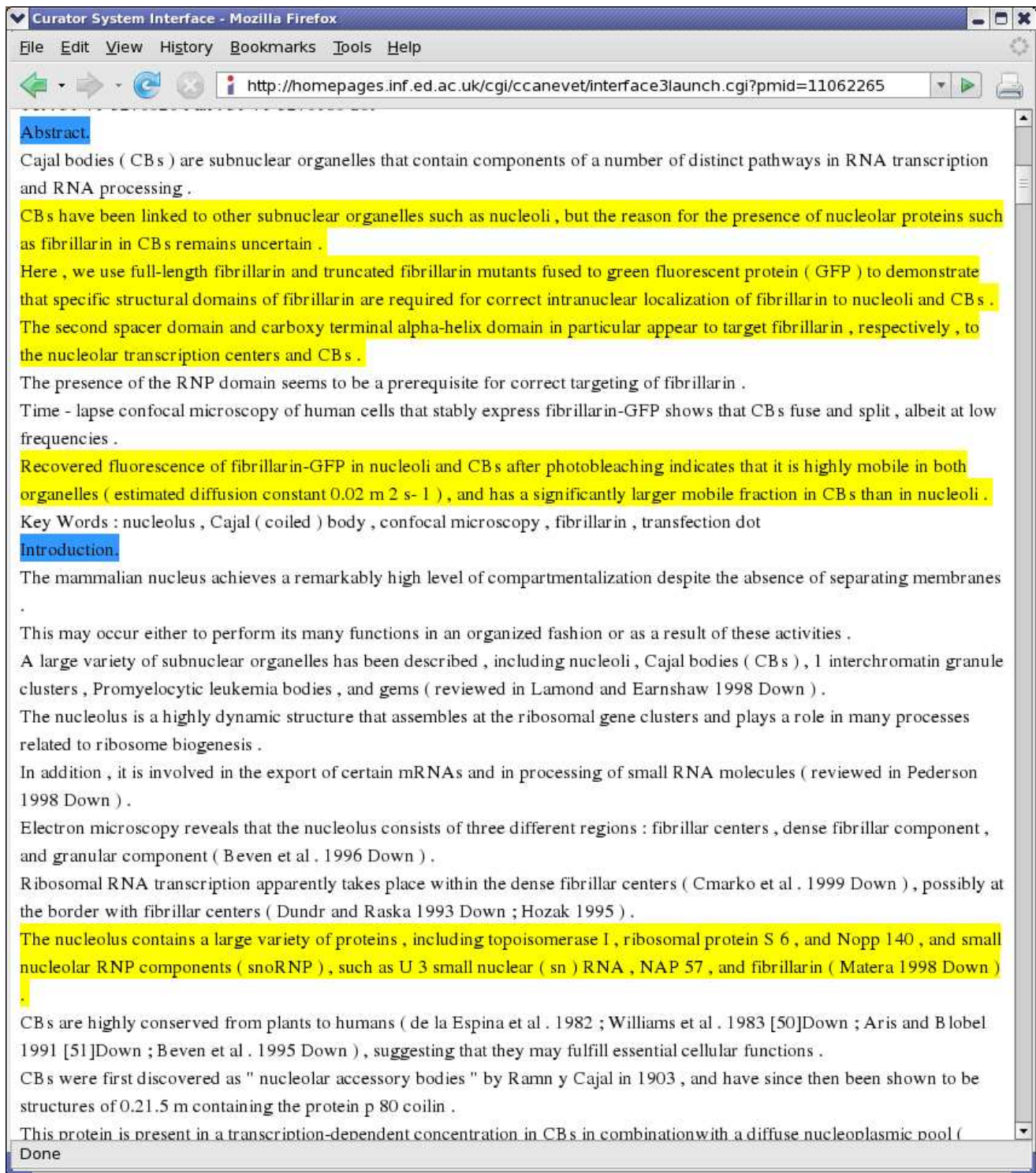
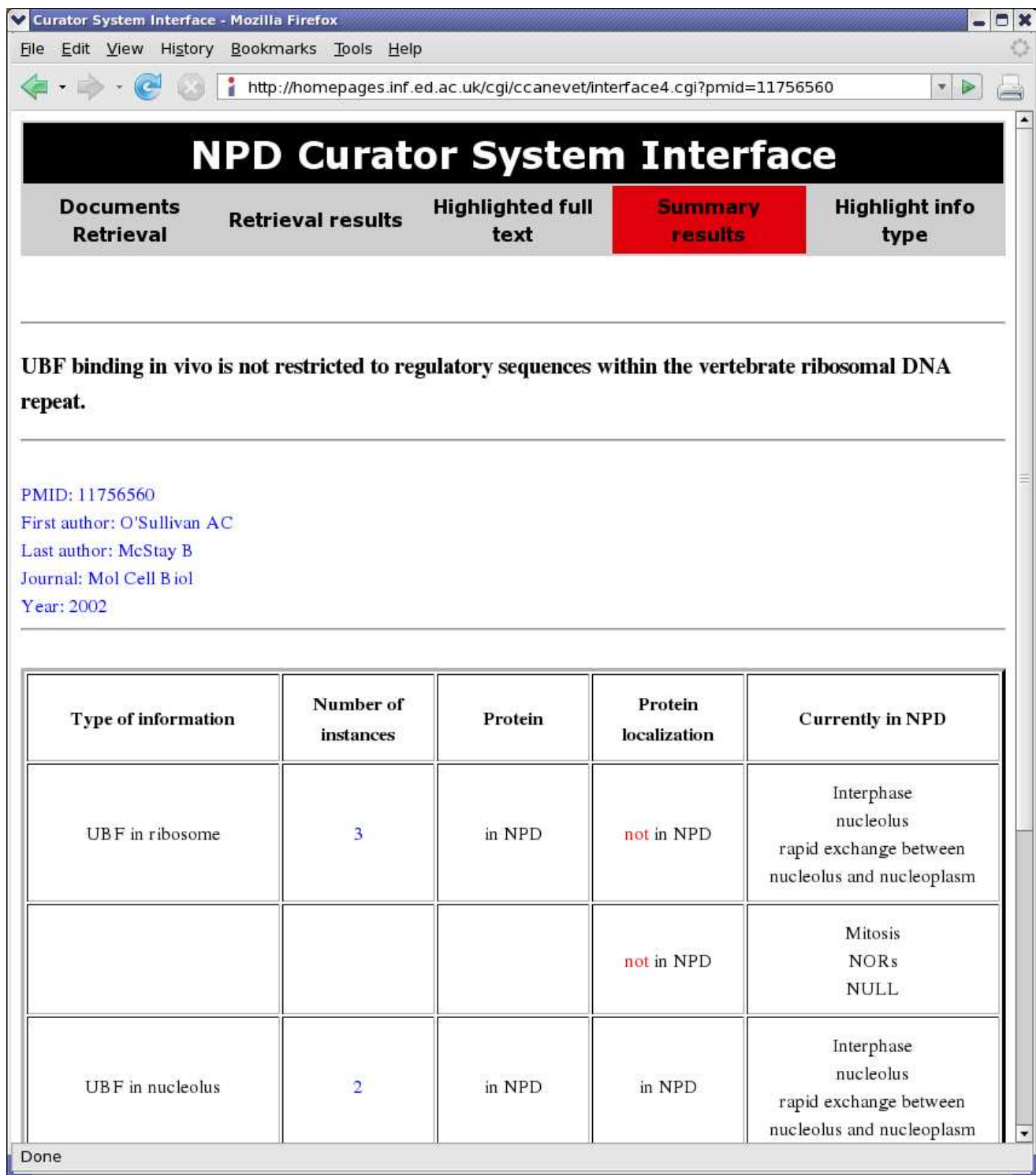


Figure 5.4: Screenshot of the “Highlighted full text” page (full text found, PMID: 11062265). Section titles in the full-text article are highlighted in blue. Sentences highlighted in yellow show the sentences classified as relevant by the supervised method presented in Chapter 3.



NPD Curator System Interface

Documents Retrieval results Highlighted full text **Summary results** Highlight info type

UBF binding in vivo is not restricted to regulatory sequences within the vertebrate ribosomal DNA repeat.

PMID: [11756560](#)
 First author: [O'Sullivan AC](#)
 Last author: [McStay B](#)
 Journal: [Mol Cell Biol](#)
 Year: [2002](#)

Type of information	Number of instances	Protein	Protein localization	Currently in NPD
UBF in ribosome	3	in NPD	not in NPD	Interphase nucleolus rapid exchange between nucleolus and nucleoplasm
			not in NPD	Mitosis NORs NULL
UBF in nucleolus	2	in NPD	in NPD	Interphase nucleolus rapid exchange between nucleolus and nucleoplasm

Done

Figure 5.5: Screenshot of the “Summary results” page. For each <protein, compartment> pair, the tool displays (in the last column of the summary Table) what information is currently held in the database about that specific protein. The format of this column is over several lines that correspond to the stage of the cell cycle, the localisation and conditions. If several localisation relations are kept in the database for this protein, several rows appear in the summary Table. The column before last draws conclusions based on the content of the NPD displayed on that row. If all rows for a <protein, compartment> pair indicate “not in NPD”, the protein localisation relation is novel to the database.

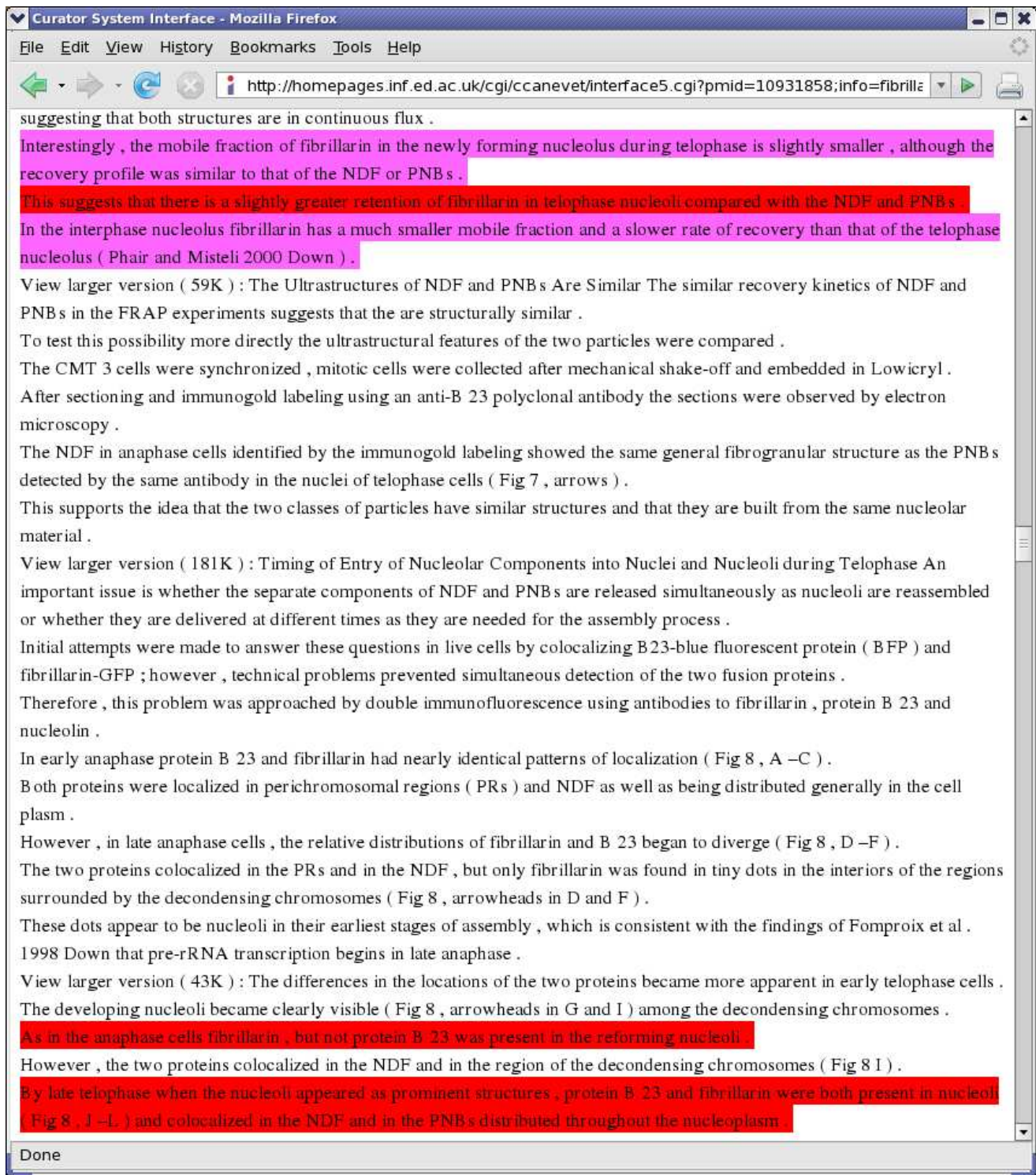


Figure 5.6: Screenshot of the “Highlight info type” page. Sentences in pink represent sentences that manifest the specific localisation relation. Sentences in red correspond to sentences that not only manifest the specific localisation relation, but also were classified as relevant sentences (see Section 4.2) and highlighted in yellow earlier on by the tool (see Figure 5.4).

5.2 Evaluation

This section compares Bickmore’s previous approach of extracting information from abstracts only and from a printed out paper version, with the new approach using my tool, which helps her extract information from full-text publications.

Prior to this evaluation period, Bickmore was asked to manually log how much time she spent extracting information from abstracts (on average ten minutes). Furthermore, it is possible to estimate how much information was extracted from abstracts or full-text papers by looking at the NPD entries (*i.e.* number of entries created, number of fields created or updated). This is what the following comparison study about the amount of information extracted is based on.

The evaluation process lasted for three months. During that time, Bickmore used the Curator System Interface to study papers that she was about to look at next for the annotation of the NPD. In total, she studied 31 full-text articles. Once the evaluation period was over, Bickmore was asked to fill in a questionnaire about how useful the tool had been. She was asked to give scores on a Likert scale¹ ranged from 1 to 5, 1 being the most helpful.

Category	Score
tool as a whole	3
sentences highlighted in yellow	3.5
summary Table	1.5
sentences highlighted in pink	3
sentences highlighted in red	2

Table 5.1: Scores indicating the usefulness of the tool (ranged from 1 to 5, 1 being the most helpful)

Table 5.1 shows that the summary Table is the most interesting component to the curator, closely followed by the sentences highlighted in red (see Section 5.1.6 for definition). Bickmore explained a summary of all the localisation relations found is quicker to study than a whole document with a few sentences highlighted. Furthermore, she commented the summary Table could at times be a little too long. This Table could be enhanced by displaying localisation relations of 2 or more sentences only, or displaying localisation relations that were captured as relevant by the DT classifier (see Section 4.2) only. The tool as a whole is scored a 3, which shows it can be improved to suit the curator’s needs, based on this evaluation.

Figure 5.7(a) shows how much time the curator spent on each paper (in minutes). Figure 5.7(b) shows the tool is able to load full-text papers (when available in HTML format, see

¹Psychometric scale used in questionnaires to measure and analyse users’ response.

discussion in Section 4.1.2) in under a minute. The Curator System Interface might be slow compared to other tools, however, it performs on full-length publications rather than abstracts. Moreover, it was developed as a prototype along different consecutive research experiments explained in Chapters 3 and 4. After this evaluation, the tool could be redesigned so as to avoid wasting time loading pages the curator benefits little from and to enhance its speed on pages the curator uses the most. While no particular bottlenecks were identified, it seems saving results that have already been loaded up (so that they can be retrieved again immediately, like in PolySearch - see Section 2.3.7) or, even better, allowing for results to be obtained offline could be the first steps towards minimising the amount of waiting time.

One recent tool that manages to obtain IE results immediately from all MEDLINE abstracts rather than the top few hundreds obtained by an IR step (*e.g.*, top 500 in the case of EBIMed - see Section 2.3.5) is FACTA (see Section 2.3.8). However, it uses a 2.2GHz server with 16GB memory to store its indexes alone. An external storage is also needed to keep sentences from MEDLINE abstracts. This would not be a possible solution unless those kinds of resources were made readily available.

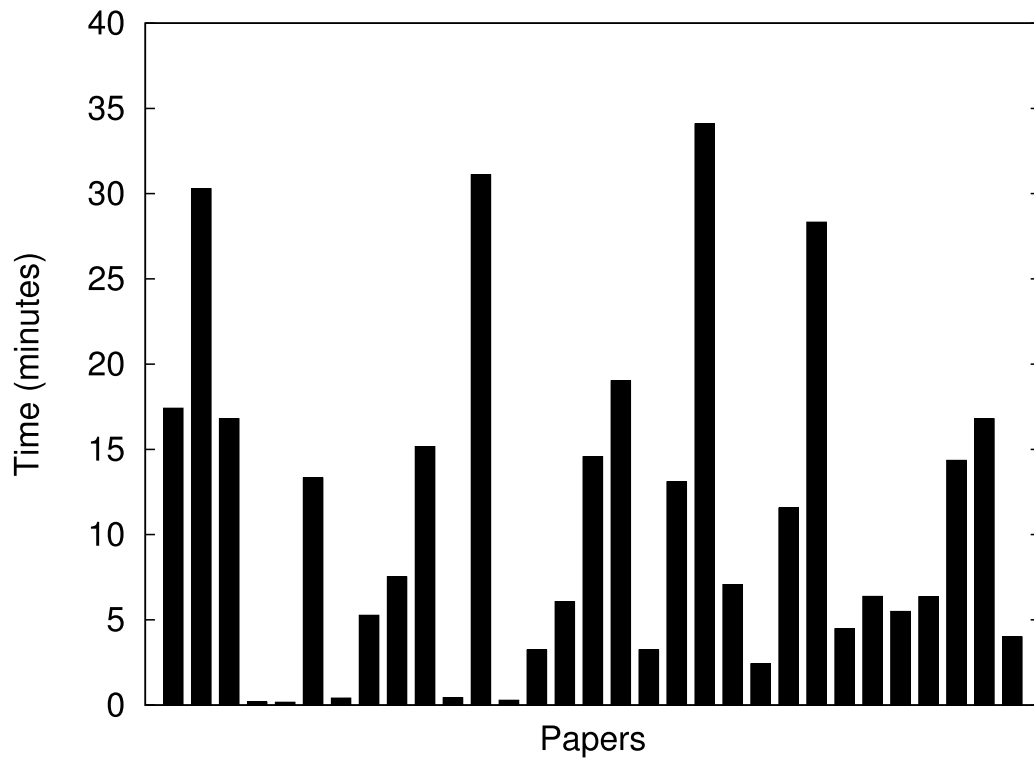
Looking at the different periods of time it takes the user to extract information from an article using the NPD Curator System Interface, it seems that the papers can be grouped into three main categories.

1. less than 5 minutes: the paper generally does not contain any information of interest for the NPD.
2. between 5 and 20 minutes: the paper does contain information of interest that can be added to the NPD.
3. more than 20 minutes: the paper contains information of interest and makes the curator check up on other issues (for example, about protein function), or may identify a new protein that needs checking through other databases (for example, PSORT [RAG⁺05] or SMART [SMBP98] for protein domains).

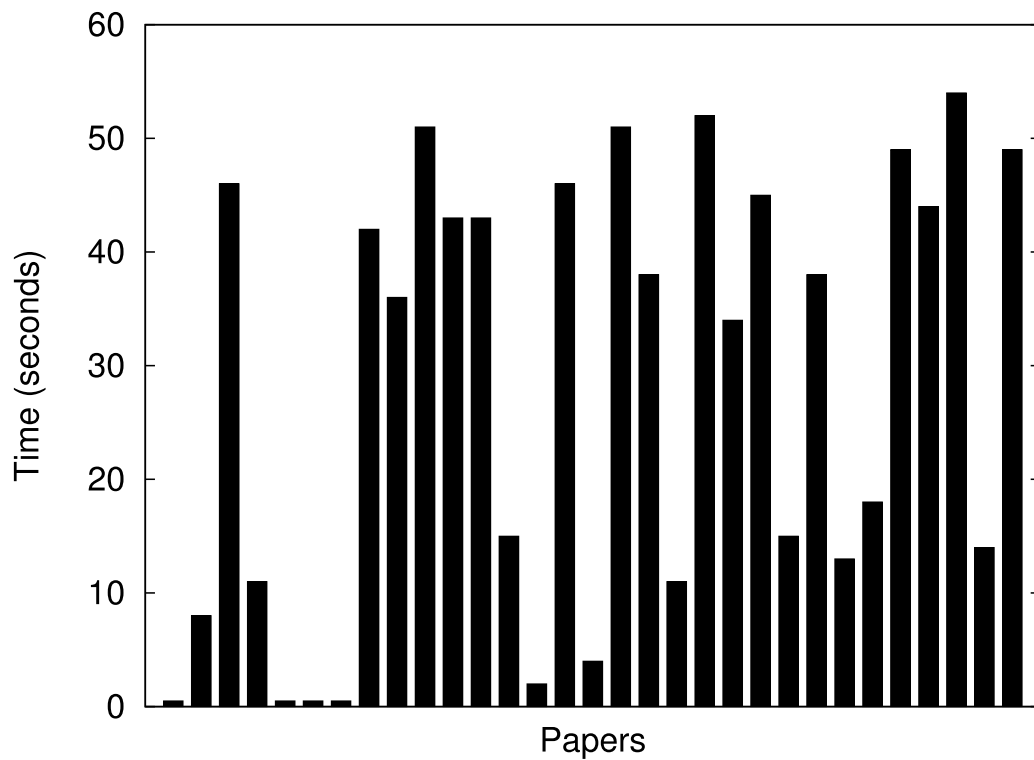
For each of these three categories, and for the 31 papers that were tested on the NPD Curator System Interface, Table 5.2 shows the number of publications that fell into each category, as well as the amount of time spent on each category on average for analysis.

5.2.1 Comparing amount of information extracted

Still looking at articles correctly identified as containing useful information (TPs), Bickmore now extracts 25% more information, as relevant sentences outside the abstract are highlighted and therefore easier to spot. She does not necessarily persevere in holding to only extracting the major localisation relations in a paper. She now takes advantage of being able to extract



(a) Graph showing the time spent on each paper by the curator in minutes



(b) Graph showing the tool loading time for full text in seconds

Figure 5.7: Graphs showing the time spent on each paper by the curator (in minutes) and the tool loading time for full text (in seconds)

Category	Number of articles tested with the interface	Average time (in minutes)
1	10	2.07
2	17	11.63
3	4	31.41

Table 5.2: Number of publications falling into each category and average time spent

information from full text without having to read through it all. Bickmore extracts facts from major and minor localisation relations in articles, as well as other facts, such as protein function. Because of this, it may be a problem that the system fails to identify a percentage of the localisation relations (FNs).

Three structured controlled vocabularies were developed by the GO project. They are “biological process”, “cellular component” and “molecular function”. The NPD fields distinguish functions of biological process type and of molecular type and associates GO terms to them. Sentences describing localisation also contain other information that is useful to extract. Indeed, of the 31 papers tested on the NPD Curator System Interface, Bickmore updated the NPD with information

- coming from 10 of the articles
- to 13 protein entries of the database
- about 7 protein localisations
- about 7 protein functions of biological process type
- about 2 protein functions of molecular type

This shows that even though sentences were highlighted based on their relevance to protein localisation, it has led to sentences that can also be relevant to protein function as well because the two are interlinked, as explained in Section 1.2.2. The next chapter gives an experiment showing how extensible the tool is from identifying sentences relevant to protein localisation to identifying sentences relevant to protein function (see Section 6.1.4).

It would have been interesting to directly compare the amount of time and information captured with and without the tool. However, this was not possible. Asking the curator to extract information from a publication twice using two different methods (with and without the Curator System Interface) would not make sense. Indeed, the results of the second method would always be falsified as the curator would already know what information the article holds. Asking a different curator to perform the second method could have been interesting. Unfortunately, my experiments were limited to a single curator. Moreover, had two curators been

available, inter-annotator agreement experiments would have been needed. Indeed, as mentioned in Section 2.1.3, curators usually work differently as they can take more or less time and extract more or less information.

5.2.2 Comparing amount of time spent annotating

For articles correctly identified as containing extractable information (TPs), Bickmore used to spend 10 minutes for each abstract alone. She now spends 11.63 minutes on average with the tool on full-text articles, and therefore 16.30% more time than she used to. The time spent varies depending on how much information is extracted as well as how many distinct NPD entries need updating and how many cross-references need adding. When Bickmore takes more than 20 minutes, she states she has actually taken time out to look at another paper or another Website in order to find more information, for example, about protein function.

Of the 31 papers tested so far on the NPD Curator System Interface, Bickmore updated the database with information coming from 10 of them. The average amount of time spent to analyse these 10 papers is 15.21 minutes per paper. Bickmore is very selective in what she annotates, which explains the fact that the number of papers in the group of category 2 (17 articles) is higher than the number of publications she updated the NPD with (10 articles). She can spend time on papers without annotating anything from them in the end. In fact, she extracted information from one paper in category 1, seven papers in category 2 and two papers from the third category.

5.2.3 Comparing amount of time wasted on FPs

Bickmore states she still looks at the abstract of articles to be able to determine whether they were incorrectly identified as containing extractable information (FPs), like she used to. On top of this, she now also takes a quick look at the summary Table (see Section 5.1.5), in case it might give interesting and novel localisation relations, which means she spends a few seconds longer than she used to.

5.2.4 Conclusion and future work

The interface allows the user to extract 25% more information from papers by spending 16.30% more time than she used to. The scoring of the tool in this evaluation part of the chapter shows the interface could be improved. First, Bickmore mentioned the summary Table can at times be especially long. The Table could be modified to be based on sentences picked up as relevant by my supervised method (sentences in yellow, see Section 5.1.4) only.

Bickmore also said she tends to go straight to the last pages of the tool. The “Highlighted

full text” page (sentences in yellow, see Section 5.1.4) could be removed and, indeed, the last two pages could present the user with more concise information, taking into account the information that was displayed in the deleted page. The only risk would be to lose some of the information Bickmore might have extracted otherwise. Conducting an experiment on this to check how much time can be saved and how much information would be lost, would be very interesting. This will be left to future work.

This evaluation is based on a single annotator because the database had only one curator at the time of the study. In [KLS⁺07], the evaluation is also based on a single subject. In [KSL⁺08], two curators took part in the study. In [AGH⁺08], the evaluation is three-fold. While the first part (reported in Section 2.3.6) is based on 4 curators, the other two parts are evaluated with a single curator. This suggests how difficult it is to carry out multi-annotators studies. Indeed, finding additional curators who will commit to spending time on evaluating our research tools can prove very difficult.

5.3 Summary

The NPD Curator System Interface provides the curator with several options. First of all, it gives the user the option of running a document retrieval method that identifies papers that are appropriate for IE. The user can then choose to launch the tool on a particular article (which is also possible directly from the homepage). The tool presents the user with highlighted text, as well as a Table summarising all the different localisation relations found in order of importance.

The interface allows the user to extract 25% more information from full-text papers by spending 16.30% more time than when working on printed-out paper version of abstracts. The trade-off seems interesting enough that the tool, as it is, can be used in day-to-day annotation after this evaluation period.

Chapter 6

Extensibility and maintainability

This chapter explains how extensible the work developed for this thesis is, as well as how easy it will be to maintain the system in order to keep it working. Indeed, for my thesis work to be of continuing value, it is important to first evaluate my methods (see Chapters 3 and 4) and the tool developed (see Chapter 5), as well as show that these methods are extensible, reusable and maintainable.

6.1 Extensibility

This section discusses the degree of extensibility of my methods. Extensibility is studied using a step by step approach.

6.1.1 From the nucleolus to the nucleus and other compartments

In previous chapters, I introduced “rule A”, which looks for compartment adjectives followed by, or in the same sentence as, a protein name (see respectively Section 4.3.1 and Section 3.4.1). Although compartments that have a corresponding adjective might be at an advantage because of this particular rule - not all compartments have a matching adjective - all other rules and computed features can be associated with all the compartments.

The training corpora, both for the MRes tool (IR) and the PhD tool (IE), comprise papers that are primarily about proteins located in the nucleolus. However, as examples below will show, other compartments were inevitably mentioned in those articles too. The tool picks up on any kind of compartments present in my lexicon (see Section 3.4.1 and Appendix C). Because the training data are never treated as strings, but always converted to features and assessed on these features, the nucleolus is “a” compartment and that is all the classifiers will learn from the sentences. Any compartment present in my lexicon can therefore fill a specific pattern and this lexicon can easily be modified if need be. This proves the extensibility of the “compartment

name” lexicon.

The “words of interest” lexicon is the other gazetteer that needs to be proven extensible. Looking at [SWJ⁺00], [DMO00] and [PPRMV04] from the training corpus, I found examples of linguistic patterns (highlighted in red) for different compartments:

- “ *the reason for the **presence of nucleolar proteins such as fibrillarin** in CBs remains uncertain”*
- “ *The GAR Domain of Fibrillarin **Contains a Nucleolar Localization Signal**”*
- “ *fibrillarin is **present in both nucleoli and CBs**”*
- “ *fibrillarin is always **seen in the NDF** and is **present in telophase PNBs**, but it **disappears from them** in early G cells”*
- “ *fibrillarin preferentially **localizes to the DFC**”*
- “ *BIG1 was partially **colocalized with nucleolin** in nuclei but not in cytoplasm, where, especially near the plasma membrane, collections of nucleolin were clearly not **associated with BIG1**.”*
- “ *In cells with little or no nuclear BIG1, nucleoporin p62 is clearly **concentrated at the periphery of most nuclei** (presumably the nuclear envelope), with some **scattered through the cytoplasm** and little to be **seen within the nucleus**.”*
- “ *There is also less nucleoporin p62 in the cytoplasm and more **scattered through the nuclei** in these cells.”*

The penultimate example might suggest that the phrase “scattered through” is a specific linguistic pattern to a mass like the cytoplasm. But the last example shows it is not the case, as the same phrase is used for the nuclei. Besides, my approach is based on a count of NEs rather than specific templates. Therefore, if the lexico-syntactic patterns that authors use to refer to the nucleolus are revealed to be in any way different to the ones used when talking about other compartments, the “words of interest” lexicon could be accordingly altered in order to allow for those to be caught.

In conclusion, although one particular feature and rule was shown biased to compartments that have a matching adjective, the “compartment name” lexicon was proven easily extensible. The other gazetteer of concern is the “words of interest” lexicon that could require further investigation for its extension (studying more example sentences, seeking advice from a domain expert, *etc.*), however, it is entirely extensible. Therefore, my work can be applicable to the whole of the nucleus by modifying two existing gazetteers.

6.1.2 From the nucleus to the cell

The following step is to discuss extensibility from the nucleus to the cell. If the approach is extensible to the whole of the nucleus by modifying the “compartment name” and “word of interest” lexicons, the approach is likely to be extensible to the rest of the cell by adapting those two lexicons to work with localisations outside the nucleus.

The Cell Component Ontology (CCO) is a controlled vocabulary that describes 160 cellular components, as well as the relationships between them. There is one class-subclass relationship and three class-instance relationships (*is-a*, *component-of* and *surrounded-by*). A class represents a type of biological entity (*e.g.*, organelle) whereas an instance represents a particular biological entity (*e.g.*, mitochondrion). Each term in this ontology has a main name, along with possible synonyms for it, a referenced definition and one or more taxonomic classes of organisms to which they apply. The class hierarchy is given in Appendix B. Using the Cell Component Ontology to create a new gazetteer and possibly extending the “words of interest” lexicon too, my approach could be extended to the whole cell. Although superclasses and relationships in CCO might not be of use, all the other terms would constitute a rich lexicon.

In Section 6.1.1, I discussed how all computed features (except for one that favours certain kinds of compartment) can work for any type of localisation terms placed in the “compartment name” lexicon. In Section 6.1.4, I back up this hypothesis with an experiment on my data using a new gazetteer on protein function. Moreover, the next section looks at previous work on domain adaptation. Indeed, changing the domain and extending system coverage to a wider cross-range of domains may have distinct problems. Evaluation of the modified or extended system would be required. If system performance were significantly degraded as a consequence of the change or extension, domain adaptation research methods could be applied to improve results. Section 6.1.3 presents several domain adaptation approaches that have been proven to help.

6.1.3 From inside the cell to outside the cell

Many other lexicons could be used to reuse my approach in order to find sentences relevant to other types of localisation.

- The Open Biomedical Ontologies (**OBO**, <http://obofoundry.org/>, [SAR⁺07]) consist of more than 50 controlled vocabularies that are well structured and developed for shared use across domains such as anatomy, behaviour, phenotype and sequence.
- The Gene Ontology (**GO**, <http://www.geneontology.org/>, [ABB⁺00]) consortium creates and maintains a controlled vocabulary that describes gene products for all organisms under three main categories (cellular component, molecular function and biological process). GO is one of the vocabularies included within OBO.

- The **UMLS** (<http://www.nlm.nih.gov/research/umls/>, [HL93]) is composed of several controlled vocabularies in the biomedical domain. It provides a mapping between these vocabularies, as well as NLP utilities that developers in medical or health informatics can use to make sense of all the different terminology systems.

For example, the Common Anatomy Reference Ontology (CARO, <http://www.obofoundry.org/cgi-bin/detail.cgi?id=caro>) is being developed by Albert Burger, Duncan Davidson and Richard Baldock at the University of Edinburgh to enhance interoperability of anatomy ontologies accross species. Writing a gazetteer based on this ontology, my approach would be extensible to identifying sentences relevant to localisation of certain entities in the anatomy.

While the extensibility discussed in Section 6.1.1 is plausible to a certain extent as the lexicons and corpora used for this thesis work already include all nuclear compartments (rather than nucleolar ones only), one cannot assume that the same approach would smoothly port to other biological topics such as disease, anatomy, or even cellular components. Domain adaptation offers techniques to make it possible to apply approaches developed using domain-specific labeled training data to text from a different domain with no labeled data available.

In [BMP06], the authors use Structural Correspondence Learning (SCL) to model the correlation of different domains with what they call “*pivot features*”. They claim and test (on POS tagging in two different domains, Wall Street Journal and MEDLINE) that it is possible to learn what features are meaningful in the source and target domains based on unlabeled data for both, train a model with the labeled data from the source domain using those features, and obtain results showing the approach generalises well to the target domain. They select *pivot features* based on frequent occurrences and similar behaviour in the unlabeled data of both domains. Those features enable them to generate correspondences between different domains. This is further explained in John Blitzer’s PhD thesis [Bli08].

In [JC06], the authors use a ranking system where they first rank features within each domain and then combine the rankings to select features that were highly ranked in all domains or “*generalisable across domains*”. They train their classifier by focussing on those features and evaluate their method by identifying gene names from 3 distinct species in text. They present 3 experiments where, in each, the training data is composed of two species and the test data of the third species. Their results show that their proposed technique outperformed results obtained by the state of the art NER methods.

More recently, in [KBP08], the authors use a “*multiobjective genetic algorithm*” in order to select the best features out of a set of all the domain-independent features collected. They name their system DINERS, which stands for “Domain Independent Named Entity Recognition System”. The two domains they tested DINERS on were Newswire (Reuters news stories from the CoNLL - conference on Computational Natural Language Learning - 2003 data set) and

judicial data (taken from the Uganda Courts of Judicature’s court cases). Their results show that using domain independent features yields enhanced performance on precision as well as recall across all entities (the training and testing data were alternated in their experiments for both domains).

6.1.4 From localisation to function

Not only does the NPD host information about nuclear protein localisation, it also contains data about their biological and molecular functions. Identifying sentences containing information on protein function involves picking up on phrases such as:

- “the ATRX complex displays ATP-dependent activities”
- “it may remodel chromatin differently”
- “modulates”
- “regulates”
- “mediates”
- “loss of this protein affects the development of ...”
- “this protein has an essential role in the development of ...”
- “this protein may act as a transcription regulator”

In order to demonstrate that the approach I have developed is extensible from detecting sentences relevant to protein localisation to sentences relevant to protein function, I designed a new experiment, which I describe below.

New lexicon on protein function

I added a new lexicon to my set of gazetteers (see Section 3.4.1) containing function related keywords so that my tool would identify them as NEs. I collected 153 terms from the function fields of the NPD, both the “molecular function” field and the “biological process” field, in order to compose this new lexicon. Examples of terms include: “acetylation”, “splicing”, “repair” and “transcription”.

Corpora

All 2638 sentences of the training corpus (see Section 3.3.1) were then tagged with function relations based on NE tags present in them. To complete Table 3.3, the number of such tags in the training corpus is 1041. This gives a density of 0.395 per sentence for this NE. I manually assessed the results obtained on the test set (see Section 3.3.2). 286 NE tags were found, which gives a density per sentence of 0.58. This means the density per sentence of the test set is higher than the one of the training set.

Results obtained using my approach

I did not compute recall, as identifying all the false negatives would have taken too much time. I gathered a precision of 0.559, which is in the same range as the precision obtained on localisation relations (see Table 4.2) rather than function relations.

Results obtained using BioIE

I performed BioIE on my test set selecting the field “Function” (rather than Structure, Diseases and Therapeutic Compounds, Localisation or Familial Relationships) in order to extract sentences that contain templates related to protein functions. BioIE returned 299 sentences, this is to say 60.65% of the test set (which contains 493 sentences). My method returned 96 sentences or 19.47% of the test set.

Manually assessing 299 sentences would unfortunately take a long time. BioIE, in this experiment, might be trading a high recall for a lower precision. However, the reason BioIE returns these 299 sentences is probably because they all happen to carry its predefined templates and rules. Perhaps this simply means my test set describes “Function” of proteins extensively. This certainly coincides with the fact that 58% of the words in the sentences contained in the test set were tagged as a function NE.

Discussion

On April 19th 2009, GO contained 27204 terms (<http://www.geneontology.org/GO.downloads.ontology.shtml>):

- 16330 for the biological process category
- 8547 for the molecular function category
- 2327 for the cellular component category
- (as well as 1392 obsolete terms not included in the above statistics)

Firstly, we notice that GO terms for function are divided into two broad categories: molecular function (*e.g.*, “is an ATPase”) and biological process (*e.g.*, “is involved in regulating splicing”). This indicates the ontology for function is more elaborated than the one for localisation. The two categories for function represent 24877 terms whereas location is covered by 2327 terms only. It means that GO has over ten times more terms for function than location. This suggests how much more there is to say about function compared to the amount there is to say about localisation and implies there must be more different ways to talk about function than localisation at the lexical, syntactic and text level.

A consequence of this content richness and language variability is that describing protein function is more complex than describing protein localisation. Such a description can span over multiple sentences, making it more difficult to identify “functional sentences” as passages might be more suitable in this case. Moreover, as pointed out in [MDK⁺05], terms used to

describe protein function are not specific to it and can be used for other purposes. It is this lack of informative and representative keywords that makes the task so difficult.

In what follows, I present previous work related to extracting protein function from free text, as described in [BLKV05], [MDK⁺05], [KMAM07] and [CO08]. In the first BioCreative competition, the goal of task 2.1 was to “assess tools able to extract text fragments to support the annotation of a given protein-GO term association” ([BLKV05], p. 6). Most participants used full-text articles and based their approach at the sentence level. Most groups used an ML method. Pattern matching and regular expressions were also popular approaches.

The highest precision (0.80) was achieved by Chiang *et al.* [CY04]. Their recall, although not calculated, was low, as their system only predicted 36 annotations correctly. Their approach made use of MeKE [JHHC02], an ontology-based text-mining system which extracts gene products’ function by studying pattern matching.

The two groups that achieved the highest number of correct predictions (TPs) were Krallinger *et al.* [KPV05] and Couto *et al.* [CSC05], with 303 and 301 annotations respectively, although both groups obtained a precision just under 0.29. Krallinger *et al.* calculated sentence scores for each NE of interest (protein names and GO terms) using weight scores based on the entity’s occurrence in Gene Ontology Annotation (GOA) abstracts as well as heuristic estimates. Couto *et al.* used an unsupervised method called FiGO (Finding Genomic Ontology) which, for each GO term, returns a rank-ordered list of sentences likely to contain the term based on its information content.

Section 2.3.4 explained how METIS struggled with extracting information about function too, with BioIE (see Section 2.3.3) achieving a precision of 0.16 for function alone. A paper published at ISMB 2007 [KMAM07] presented an IE system where curators provide relevant and irrelevant sentences to a given topic to start with. The system then deduces IE rules from the training data that users can choose to select or disqualify. Based on those, the system returned a set of relations found in the text on various topics of interest. After gathering a precision of 0.66 and a recall of 0.15 on protein function, the authors warn:

“In a biomedical context, protein structure is about a protein itself and its components, and can be expressed in a simple way in texts, whereas function- and disease-related relations appear to exhibit a higher level of complexity and cannot be so narrowly defined.” ([KMAM07], p. i262)

In 2008, the authors of [CO08] describe GEANN, a “Genomic Entity Annotation System”, whose experiments on associating GO terms to genes yielded better results than previous attempts. They created textual patterns by studying GO concept evidence publications and extracting patterns that appeared frequently without figuring in all abstracts. Their textual patterns were then extended semantically using WordNet [Fe198] (a database of English words) and similarity measures. Moreover, larger patterns were constructed when overlappings were

found in previously created ones. The system was tested on its annotation predictions for genes in GenBank [BKML⁺05], which were already annotated. Results from 114 GO terms (from all 3 categories) generated for 7357 genes (with 6805 evidence abstracts) were evaluated and gave a precision of 0.78 and a recall of 0.61. Furthermore, they claim their approach could generate annotations when run over full text too.

Conclusion

Automatically extracting information about protein function has been proven to be an extremely difficult task. The results of this attempt, performed by adding a new lexicon to my system, can be placed in the same context as the previous efforts. Nevertheless, the results of this experiment demonstrate the extensibility of my approach.

6.1.5 Unsupervised learning for extensibility

If the NPD Curator System Interface did not need any training data, it would be more extensible. In Section 3.6, Infomap produced a rank-ordered output without using any training data. This method could ultimately be used instead of my supervised classifier. One would have to go further down the list though, in order to retrieve all the different types of information, as Table 3.20 shows in chapter 3.

Moreover, as mentioned in Chapter 5, it seems the curator tends to go directly to the page of the tool that displays the summarising Table. This Table is not generated using any training data. Therefore, the most valuable part of the tool for the user is actually extremely extensible. Alternatively, Infomap could also be used to perform a cosine similarity approach to group similar sentences together.

6.1.6 Generic re-usable elements

Other than creating and using new lexicons, my approach is extensible in other ways. Indeed some concepts I used in my PhD work are generic enough to be applicable to any kind of annotation. To this extent, the work achieved in this thesis could potentially be applied to other domains.

- In section 3.4.5, I introduced a new metric $A@n$. $A@n$ stands for at least one instance of each answer. This new metric derives from the known metric “ $a@n$ ”, which itself stands for at least one instance of “the” answer. Using rank-ordered lists of results and $A@n$ made the analysis of the results easier and the discussion more interesting.
- Previous research work has been done on sentence similarity using the BLEU score, word-level edit-distance (see [SZK⁺06]) or vector space models and cosine similarity experiments using, for example, Infomap [WP]. In this thesis work, sentences were

tagged with labels according to what localisation relations were found in them, and were then grouped based on the labels they had in common.

- Finally, using colour coding to highlight sentences is another concept that would be extensible to other work. Table 4.3 in section 4.5 gives results that show that sentences highlighted in red by the tool have indeed a much higher precision than sentences in pink. The evaluation of the tool also confirmed this was useful for the user.

6.2 Maintainability

This section discusses what would need to be done, as well as how often, in order to maintain my system and keep it working to its best potential.

6.2.1 MRes tool

The tool uses the year of publication as one of its features. This particular feature has a set number of values (similarly to values in Table 3.6). Indeed, the classification is more efficient with a set number of values (a value for each recent year and the value “old”) rather than with integer as type of the feature. For example, DT is able to have branches labeled “year: old” and have a specific treatment for less recent articles. Each year, this set of values needs to be updated with the new year coming up, so that papers to appear in the following year will have a value for its year attribute that is recognisable by the system. The threshold for the “old” papers could also be updated by simply deleting older years from the set of values.

The tool also uses the list of all the articles (or more precisely their PMIDs) cited in the NPD in order to compute the number of related articles to a particular paper (under the text categorisation process) already entered in the database. The “related articles” feature, as shown in [Can04] (p. 27), is one the strongest features of the tool, it could therefore be important to update this list.

Other than that, the reason behind updating the system any further would be that the writing style of publications has evolved or, even more likely, that new concepts have emerged in the research field, which authors describe in ways that are not reflected in the current training corpus. Whenever the material contained in the NPD has been modified significantly enough compared to the 2004 version of the database the tool is using at the moment, it would be ideal to update the training data for the tool. (The document retrieval step of the tool is based on the 2004 version of the NPD whilst other parts of the tool are based on the latest version as explained in Chapter 4.)

I updated the data in 2005 and it did not make much difference. In fact, the results were similar and slightly better with the 2004 version of the database so I kept using the latter.

However, after five or ten years, the tool could benefit from an update or it might not perform at its best. The appropriate frequency of updating is an empirical question that will have to be assessed over the next few years. For example, one could produce a learning curve giving results obtained after updating the tool every year. It would then be possible to determine the number of years after which the curve shows significant improvement, and therefore the number of years after which it would be best to update the system.

This task would involve rebuilding models used to compute some features (see [Can04] Figure 7 p. 32), recomputing and updating features. Four sub-tasks can be identified: one would need to

- build a new Rainbow model (Rainbow was introduced in Section 4.1.1).
- build a new MaxEnt model (MaxEnt was introduced in Section 2.2.9). The system uses an Ensemble method which combines results from DT, NB and MaxEnt.
- compute features for all the new PMIDs cited in the NPD in arff format in order to run WEKA on DT and NB, and update the training corpus and its features. WEKA and the arff format were introduced in Section 2.2.6.

6.2.2 Retrieving full text

In order to retrieve full text in HTML, my script goes from one URL to the next following the links to the page that actually displays the full-text paper in HTML format. This process could perhaps fail in the future if the Websites used made any modifications to their HTML format or the way they work. Also if SciXML (discussed in Section 4.1.2) or any other format (see Section 7.2) makes a breakthrough, it will be more and more used for scientific papers, but the Curator System Interface will not deal with this format unless adapted to it.

Chapter 3 covers text preprocessing and explains that different journals may have different formats for their HTML source pages. My text preprocessor is, to an extent, customised to the current standards of the main journals of interest to the curator. Again, if these journals change their way of displaying information, this process could perhaps fail.

6.2.3 Updating the protein names lexicon

For the summarising Table (see Section 5.1.5), in order to give accurate information about what is new or not new to the NPD, the tool checks that it is working with the up-to-date version of the database flat files and if it is not, downloads the latest version uploaded on <ftp://ftp.hgu.mrc.ac.uk/pub/npd/>. The novelty-detection component can then provide the user with the information that is currently contained in the database. The protein names lexicon could be updated with new protein entries to the NPD each time the tool downloads a

new version of the database, or whenever a significant amount of proteins have been added to the database.

6.3 Summary

In this chapter, I showed how extensible the work achieved for this thesis is. I first discussed extensibility from one localisation to a broader one (from the nucleolus to the nucleus, from the nucleus to the cell, from the cell to outside the cell). I then showed, using an experiment, how extensible the approach was from identifying sentences relevant to the localisation of proteins to identifying sentences relevant to the function of proteins. I also discussed how much the tool could rely on unsupervised methods, which do not use any training data and are therefore a lot more extensible. Finally, the last section of this chapter gave suggestions as to how much work needs to be done to keep the tool working at its best.

Chapter 7

Epilogue

The first part of this chapter gives conclusions for this thesis. It revisits the initial claims made in Section 1.5 and offers a discussion on the contribution this PhD thesis brings to the field. The second part of this chapter presents ideas for future work on this project and in the field in general.

7.1 Conclusions

7.1.1 Claims of the thesis revisited

The research I have carried out supports the claim that relevant information that is new to biomedical databases can be found automatically, as one step in minimising the amount of time a human expert will need to spend to keep these databases up to date.

The claim was supported by presenting methods and their results working for a particular database - the NPD - and by demonstrating the extensibility of the approach.

Bickmore could not afford to spend time reading through a whole publication on a printed out paper version, or online. She can now extract 25% extra information from full text articles using the NPD Curator System Interface.

If instead of the 10 minutes that she used to spend on average reading through an abstract, it now takes her 16.30% more time on a full text article, the extra minute(s) mean she can annotate the database with major localisation relations contained in the paper, minor ones that were not mentioned in the abstract, as well as other information, such as protein function.

7.1.2 Contribution to the field

The NPD Curator System Interface is based on extensible methods, as discussed in Section 6.1. In order to obtain good results that are specific to a particular area, one needs to customise a

general tool with appropriate resources. Although the tool is customised to the NPD's needs, it can be adapted to other projects.

I have also contributed to the field by developing annotated corpora (see Section 3.3), as well as gazetteers. Specialised resources are better at recognising NEs in specialised text rather than every-day text. I have, for this purpose, created such resources for several different kinds of entities (see Sections 3.4.1 and 6.1.4).

The software, annotated corpora and gazetteers are all available on request. Please contact me at catherine.canevet@bbsrc.ac.uk.

7.2 Future work

This section presents ideas that could be explored in the future. For example, more information could be extracted from the text. Rather than looking at <protein, localisation> pairs or <protein, function> pairs, work could be done to extract extra information about the conditions of these facts (which the database stores in a field named "Detail"):

- Manner (using keywords such as adverbs or "by", "through", "via", "using", "direction", "*in vitro*", "*in vivo*")
- Instrument (using keywords such as "with", "without the aid of", "by", "through", "via", "using")
- Temporal (using keywords such as "during", "before", "after")
- Condition (using keywords such as "in the presence of", "in response to")

In order to illustrate this issue, a sentence was selected from the training corpus as an example. Manner is highlighted in red while temporal information is highlighted in blue:

By confocal immunofluorescence microscopy, BIG1 was localized with nucleoporin p62 at the nuclear envelope (probably during nucleocytoplasmic transport) and also in nucleoli, clearly visible against the less concentrated overall matrix staining.

Some of the words that key manner also key instrument (*e.g.*, "by", "through", "via", "using"). Therefore, something more than keywords would be needed to distinguish them. Also manner, condition, temporal and, possibly, instrument will probably take more than one word to specify. This is not something I considered in looking for proteins and compartments.

Once this extra information was captured, one could use XML tags to label each passage or sentence with them. In a talk given at the BOTM workshop ("Bridging Ontologies and Text-Mining", <http://www.ebi.ac.uk/Information/events/botm/>, 2007), Jung-Jae Kim showed the use of such tags to annotate text passages with attributes such as polarity (*i.e.*

whether the sentence is positive or negative), source, target, phase, condition of events. <http://www.ebi.ac.uk/~kim/eventannotation/> gives a short introduction and examples of the annotation used. Using XML tags to label passages this way would enable the intra-document novelty detection of my approach (see Section 4.3) to cluster sentences per type of information of a much finer granularity.

As discussed in Section 3.7, the NPD Curator System Interface could take input from its users in order to update the training corpus and improve the tool's results continuously. Moreover, the NPD Curator System Interface could not only provide the curator with more information about sentences (or indeed entire text passages), enhance results on a regular basis using the curator's feedback, it could also let the user specify what protein name they want to work with on a particular article and give them the option of saving that protein name in the lexicon so that the tool can recognise it in the future. [STMA08] demonstrates how users can improve the performance of tools this way.

More generally, as bio-curators do not always concur on annotation, there is a need to get original authors involved. Indeed, it makes sense to get them to add as much metadata as possible before articles enter the publishing process. IE is traditionally run after papers has been published as raw free text. This means that the intent of the author is not clearly known and IE results are not easy to validate. Furthermore, the results produced do not conform to a standard format. The quality of the results could be greatly improved if authors were providing support for future IE. If authors were providing information on their publication whilst they were writing it, the authors' intent would be correctly captured and the information could be stored in a standard format. This would require an agreed format, tools for authoring as well as authors and publishers to comply. It might prove like a difficult stage to get to, however, this seems like a more effective distribution of the labour.

Text-mining could be facilitated and indeed improved if publications were correctly associated with ontology terms as keywords or even in the text every time such a concept is mentioned. Ontologies are well developed, and this would definitely ameliorate text-mining results. Moreover, database entries could be linked directly from the text. This would enrich articles considerably as they would become an integral part of the Semantic Web. It would also reduce the problem of ambiguity in NER tasks.

As it is not possible to expect authors to annotate their papers with such information while they are writing it in their usual text editor, work has recently been undertaken to provide help for authors to enrich their manuscripts semantically. Firstly, in his Masters thesis [Kav08] Silvestras Kavaliauskas developed an online tool called PaperMaker (<http://www.dev.ebi.ac.uk/Rebholz-srv/PaperMaker/>). Authors feed the tool their original paper, PaperMaker then goes through a succession of modules in the following order:

- module 1: spell check

- module 2: acronym resolution
- module 3: NER
- module 4: GO term recognition
- module 5: MeSH term recognition
- module 6: reference check (this module finds authors that were found in the bibliography but not in the body of the publication and vice versa)
- module 7: related work (this module finds related work that is not already mentioned in the bibliography)
- module 8: summary (this module offers a summary of the article from its number of words to its possible keywords, GO and MeSH terms *etc.*)

Finally, authors can download a modified publication or a digital abstract. Either way, this allows them to submit a paper where general readability is improved, the use of domain terminology is consistent, and indexing this article correctly is made easier.

The OKKAM project (<http://www.okkam.org/>) entitled “Enabling the Web of Entities” provides amongst other things an entity editor in Microsoft Word as part of one of their work-packages named “Entity-centric authoring environment”. While this is only a prototype, a Microsoft Word 2007 plugin has been developed at the University of California San Diego in collaboration with Microsoft External Research, as part of the BioLit [FKWB08] project (<http://biolite.ucsd.edu/>). This Microsoft Word 2007 add-in (<http://www.codeplex.com/UCSDBioLit>) allows authors to embed metadata into their manuscript. As they are writing, hovering their mouse over a word will offer them possible mark-up(s). It is then possible for them to ignore the suggestions, view the proposed term in its ontology (or all terms in their different ontologies) or add one of the mark-ups suggested. Once a term has been marked-up, hovering over it will offer authors to apply this mark-up to all other occurrences of this term in the document or to stop recognising this term as such. The ontologies used are OBO (see Section 6.1.3) to which it is possible to add custom metadata to suit specific needs. It will also soon be possible to provide the tool with a url to download another ontology of interest to the authors. In terms of databases identifiers assigned, at the moment those come from GenBank/RefSeq [PTM05], UniProtKB/Swiss-Prot [BBA⁺03] and the Protein Data Bank (PDB [BKW⁺77, BWF⁺00]).

Microsoft Word is not the only writing environment used in the scientific community, LaTeX is also extensively used. SALT (Semantically Annotated LaTeX, [GMH⁺07]) allows LaTeX authors to add information about the claims they are making and their arguments. KonneX [GHMD08] is an infrastructure created in order to support finding claims

and links within Argumentation Discourse Networks (ADNs). It is a Semantic Web application which uses the Resource Description Framework (RDF, <http://www.w3.org/RDF/>) as well as the Linking Open Data (LOD, <http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>). RDF is based on XML but is highly structured. The principle behind LOD is to link data on the Semantic Web. This allows KonneX to offer users a navigation over ADNs for articles that have been written using SALT.

Other recent initiatives have emanated the same flavour. The society for Neuroscience in 2008 launched new strategies for data mining. One of the suggested strategy was to publish with metadata so as to improve IR, another one was to enhance links between databases and journals. The SWAN project (Semantic Web Applications in Neuromedicine) is working on the creation of the SWAN ontology, which is a framework aimed at enriching scientific discourse based on the Semantic Web approach. Having more structure in the data will definitely mean less hard work for new IE software.

Whilst in chemical informatics, the International Chemical Identifier (InChI) has solved the problem of having unique accessions for each concept, in bioinformatics no such thing has arisen and NER for genes and proteins in text still represents a big challenge. Microsoft Office Word 2007 also provides a way for authors to write semantically-rich chemistry information easily with Chem4Word (<http://research.microsoft.com/en-us/projects/chem4word/>).

Another initiative in biochemistry was the Federation of European Biochemical Societies (FEBS) Letters Structured Digital Abstracts (SDAs) experiment launched in April 2008 and implemented in ScienceDirect. Authors can provide SDAs using four-column spreadsheets where the first two columns represent protein names (X and Y), the third one is a verb (V) while the last one is a method Z. For example, the quadruple ($X, Y, interact, Z$) would correspond to a sentence like “protein X interacts with protein Y, by method Z”. This is a nice experiment coming from a journal but reveals the need for an appropriate XML editor or other authoring environment at the very least for SDAs.

Will publishers soon require authors to submit SDAs or even entire publications semantically enriched? Would semantic links then become part of the peer review process? Or will authors publish and contribute to the scientific community using different means? Possibly one step ahead of the game, Robert Hoffmann has created Wikigenes ([Hof08], <http://www.wikigenes.org>), a wiki system about genes, proteins and chemical compounds where authorship is recorded, authors can rate each other and the data is semantically linked. Will such wikis turn out to be the future of scientific publishing? One way or another, it looks as though efforts are undertaken, solutions are on their way and the days of “*raw text-mining*” are being counted.

In fact, in June 2009, the ninth annual meeting of BioLINK (Linking Literature, Information and Knowledge for Biology) Special Interest Group will take place at the conference on Intelligent Systems for Molecular Biology and it will be the first time that BioLINK holds a session on scientific publishing and how it will affect IE methods in the future. Indeed, text miners should not be out of a job but they will have to adapt to new tasks taking into account all the various metadata and links made available. This new trend will hopefully significantly enhance results, which will be very welcome in areas where *raw text-mining* has been struggling, such as detecting information on protein function (see Section 6.1.4).

Appendix A

Abbreviations

ADN: Argumentation Discourse Network
AROC: Area under the ROC curve
BLEU: BiLingual Evaluation Understudy
CARO: Common Anatomy Reference Ontology
CB: Cajal Bodies
C&C tagger: Curran and Clark tagger
CCO: Cell Component Ontology
DFC: Dense Fibrillar Components
DNA: Deoxyribonucleic acid
DT: Decision Tree
EBI: European Bioinformatics Institute
Ensemble: Ensemble Learning method
ER: Endoplasmic Reticulum
FC: Fibrillar Centers
FEBS: Federation of European Biochemical Societies
FN: False Negative
FP: False Positive
FPR: False Positive Rate
GC: Granular Components
GeneRIF: Gene Reference Into Function
GFP: Green Fluorescent Protein
GO: Gene Ontology
GOA: Gene Ontology Annotation
GPs: Gaussian Processes
GS: Gold Standard
GSA: Gold Standard Annotations

GTT: Generic Topic Template
HGMD: Human Genome Mutation Database
HMDB: Human Metabolome Database
HPRD: Human Protein Reference Database
IE: Information Extraction
InChI: International Chemical Identifier
IR: Information Retrieval
KDD: Knowledge Discovery and Data mining
KEGG: Kyoto Encyclopedia of Genes and Genomes
LOD: Linking Open Data
MAP: Mean Average Precision
MaxEnt: Maximum Entropy
MeSH: Medical Subject Headings
ML: Machine Learning
mRNA: messenger RNA
MRR: Mean Reciprocal Rank
MS: Mass Spectrometry
MSA: Multiple Sequence Alignment
MT: Machine Translation
NB: Naive Bayes
NDF: Nucleolus-Derived Foci
NE: Named Entity
NER: Named Entity Recognition
NIST: National Institutes of Standards and Technology
NLM: National Library of Medicine
NLP: Natural Language Processing
NPD: Nuclear Protein Database
OBO: Open Biomedical Ontologies
OCR: Optical Character Recognition
OMIM: Online Mendelian Inheritance in Man
PCA: Principal Component Analysis
PDB: Protein Data Bank
PMID: PubMed Identifier
PML: ProMyelocytic Leukaemia
POS: Part Of Speech
PPI: Protein-Protein Interaction
RDF: Resource Description Framework

rDNA: ribosomal DNA
RER: Rough ER
RN: (Chemical Abstracts Service) Registry Number
RNA: RiboNucleic Acid
RNP: RiboNucleoProtein
ROC: Receiver Operating Characteristic
SALT: Semantically Annotated LaTeX
SCL: Structural Correspondence Learning
SDA: Structured Digital Abstract
SER: Smooth ER
snRNP: small nuclear RNP
SVD: Singular Value Decomposition
SVM: Support Vector Machine
TC: Text Categorisation
TF.IDF: Term Frequency . Inverse Document Frequency
TN: True Negative
TNR: True Negative Rate
TP: True Positive
TPR: True Positive Rate
TREC: Text REtrieval Conference
tRNA: transfer RNA
TTT2: Text Tokenisation Tool 2
UMLS: Unified Medical Language System
VSM: Vector Space Model

Appendix B

The Cell Component Ontology (CCO)

The Cell Component Ontology (CCO) is a controlled vocabulary that describes 160 cellular components as well as the relationships between them. This appendix lists the classes and subclasses in its class hierarchy (without displaying the terms that correspond to the instances of each subclass 6.1.2).

- cell fraction
 - microsome
- cell surface matrix
 - cell wall
 - extracellular matrix (sensu Animalia)
- envelope
 - cell envelope (sensu Bacteria)
 - organellar envelope
- membrane
 - endoplasmic reticulum membrane
 - endosome membrane
 - microbody membrane
 - mitochondrial membrane
 - nuclear membrane
 - plasma membrane
 - plastid membrane

- thylakoid membrane
- Golgi membrane
- lysosomal membrane
- outer membrane (sensu Gram-negative Bacteria)
- vacuolar membrane
- vesicle membrane
- organelle
 - membrane-bound organelle
 - non-membrane-bound organelle
- space
 - endoplasmic reticulum lumen
 - endosome lumen
 - microbody lumen
 - periplasmic space
 - plastid intermembrane space
 - plastid stroma
 - thylakoid lumen
 - cytosol
 - extracellular space
 - Golgi lumen
 - lysosome lumen
 - mitochondrial intermembrane space
 - mitochondrial lumen
 - nuclear lumen
 - perinuclear space
 - vacuolar lumen
 - vesicle lumen
- suborganelle compartment
 - Golgi cisterna

- plastid thylakoid
- super component
 - cytoplasm

Appendix C

The compartment name lexicon

```
<?xml version="1.0"?>
<lexicon name="cn">
  <lex word="nucleus"/>
  <lex word="nuclei"/>
  <lex word="nucleolus"/>
  <lex word="nucleoli"/>
  <lex word="nucleoplasm"/>
  <lex word="speckle"/>
  <lex word="speckles"/>
  <lex word="paraspeckles"/>
  <lex word="NDF"/>
  <lex word="NDFs"/>
  <lex word="Sam68"/>
  <lex word="SLM"/>
  <lex word="SNB"/>
  <lex word="PML"/>
  <lex word="chromatin"/>
  <lex word="heterochromatin"/>
  <lex word="interchromatin"/>
  <lex word="IGC"/>
  <lex word="IGCs"/>
  <lex word="SLMs"/>
  <lex word="SNBs"/>
  <lex word="PMLs"/>
  <lex word="chromatins"/>
  <lex word="heterochromatins"/>

```

```
<lex word="cajal" />
<lex word="coiled" />
<lex word="CB" />
<lex word="CBs" />
<lex word="gem" />
<lex word="gems" />
<lex word="polycomb" />
<lex word="PcG" />
<lex word="Pc-G" />
<lex word="Pc-Gs" />
<lex word="NPC" />
<lex word="OPT" />
<lex word="PcGs" />
<lex word="NPCs" />
<lex word="OPTs" />
<lex word="cytoskeleton" />
<lex word="lamina" />
<lex word="laminas" />
<lex word="INM" />
<lex word="PNC" />
<lex word="INMs" />
<lex word="PNCs" />
<lex word="chromosomal" />
<lex word="chromosome" />
<lex word="chromosomes" />
<lex word="cleavage" />
<lex word="PR" />
<lex word="PRs" />
<lex word="cytoplasm" />
<lex word="granular" />
<lex word="GC" />
<lex word="MIG" />
<lex word="PNB" />
<lex word="DFC" />
<lex word="GCs" />
<lex word="MIGs" />
<lex word="NORs" />
```



```
<lex word="PNBs" />
<lex word="DFCs" />
<lex word="fibrillar" />
<lex word="spindle" />
<lex word="TC" />
<lex word="spindles" />
<lex word="TCs" />
<lex word="Golgi" />
<lex word="synaptonemal" />
<lex word="SC" />
<lex word="SCs" />
<lex word="centrosome" />
<lex word="centrosomes" />
<lex word="centromere" />
<lex word="centromeres" />
<lex word="endoplasmic" />
<lex word="reticulum" />
<lex word="ER" />
<lex word="ERs" />
<lex word="IR" />
<lex word="IRs" />
<lex word="kinetochore" />
<lex word="X" />
<lex word="XY" />
<lex word="telomere" />
<lex word="ribosome" />
<lex word="telomeres" />
<lex word="ribosomes" />
</lexicon>
```


Appendix D

The protein keyword lexicon

```
<?xml version="1.0"?>
```

```
<lexicon name="pk">
```

```
  <lex word="protein"/>
```

```
  <lex word="receptor"/>
```

```
  <lex word="kinase"/>
```

```
  <lex word="enzyme"/>
```

```
  <lex word="histone"/>
```

```
  <lex word="proteins"/>
```

```
  <lex word="receptors"/>
```

```
  <lex word="kinases"/>
```

```
  <lex word="enzymes"/>
```

```
  <lex word="histones"/>
```

```
  <lex word="kDa"/>
```

```
  <lex word="GFP"/>
```

```
</lexicon>
```


Appendix E

The list of stop words

a	accordingly	after	again	against
all	almost	already	also	although
always	among	an	and	another
any	anyone	apparently	are	as
aside	at	away	be	because
been	before	being	between	both
briefly	but	by	can	cannot
certain	certainly	copyright	could	did
different	due	during	do	does
done	each	either	else	enough
especially	et-al	etc	ever	every
following	for	found	from	further
gave	gets	give	given	giving
gone	got	had	has	hardly
have	having	here	her	how
however	if	immediately	importance	important
in	into	is	it	its
itself	just	keep	kept	kg
km	mg	might	knowledge	largely
like	made	mainly	make	many
may	ml	more	most	mostly
much	mug	must	nearly	necessarily
neither	next	no	none	nor
normally	nos	not	now	of
often	on	only	or	other
ought	our	out	owing	particularly
past	perhaps	please	poorly	possible

possibly	potentially	somewhat	regardless	predominantly
present	previously	primarily	probably	prompt
promptly	quickly	quite	rather	readily
really	recently	refs	relatively	respectively
results	upon	several	should	significantly
similar	similarly	since	slightly	so
some	sometimes	somewhat	soon	specifically
strongly	substantially	successfully	such	sufficiently
than	that	the	their	theirs
them	then	there	therefore	these
they	this	those	though	through
throughout	to	too	toward	towards
under	until	upon	usefully	usefulness
usually	various	was	were	what
when	where	whether	which	while
who	whose	why	widely	will
with	within	without	would	yet

Appendix F

Examples of sentence classification

F.1 Third sentence of the abstract of article [RRB⁺03]

Table F.1 displays the four columns for the third sentence of the abstract of article [RRB⁺03]:

“Microscopic analysis of both fixed and live mammalian cells showed that NuSAP is primarily nucleolar in interphase, and localizes prominently to central spindle microtubules during mitosis.”

The resulting feature vector is “pn1, pk0, cn1, ca1, ck1, phn, loc1, int1, ra1, rb1, rc1, 44.44444444, y”.

F.2 Title of article [CSK98]

Table F.2 displays the four columns for the title of article [CSK98]:

“A specific subset of SR proteins shuttles continuously between the nucleus and the cytoplasm.”

The resulting feature vector is “pn1, pk1, cn2, ca0, ck0, ph0, loc0, int1, ra0, rb0, rc1, 50, y”.

F.3 Fourth sentence of the abstract of article [KZCJ02]

Table F.3 displays the four columns for the fourth sentence of the abstract of article [KZCJ02]:

“hCdc14A dynamically localizes to interphase but not mitotic centrosomes, and hCdc14B localizes to the interphase nucleolus.”

The resulting feature vector is “pn2, pk0, cn2, ca0, ck0, phn, locn, int0, ra0, rb1, rc0, 75, y”.

Microscopic	JJ	O	NONE
analysis	NN	O	NONE
of	IN	O	STOP
both	DT	O	STOP
fixed	JJ	O	NONE
and	CC	O	STOP
live	JJ	O	NONE
mammalian	JJ	O	NONE
cells	NNS	O	NONE
showed	VBD	B-INT	INT
that	IN	O	STOP
NuSAP	NNP	B-PN	PN
is	VBZ	O	STOP
primarily	RB	O	STOP
nucleolar	JJ	B-CA	CA
in	IN	O	STOP
interphase	NN	B-PHAS	PHAS
,	,	O	NONE
and	CC	O	STOP
localizes	VBZ	B-LOC	LOC
prominently	RB	O	NONE
to	TO	O	STOP
central	JJ	O	NONE
spindle	NN	B-CN	CN
microtubules	NNS	B-CK	CK
during	IN	O	STOP
mitosis	NN	B-PHAS	PHAS
.	.	O	NONE

Table F.1: Four column-output from the C&C tagger for the third sentence of the abstract of article [RRB⁺03]

A	DT	O	NONE
specific	JJ	O	NONE
subset	NN	O	NONE
of	IN	O	STOP
SR	SYM	B-PN	PN
proteins	NNS	B-PK	PK
shuttles	VBZ	B-INT	INT
continuously	RB	O	NONE
between	IN	O	STOP
the	DT	O	STOP
nucleus	NN	B-CN	CN
and	CC	O	STOP
the	DT	O	STOP
cytoplasm	NN	B-CN	CN
.	.	O	NONE

Table F.2: Four column-output from the C&C tagger for the title of article [CSK98]

hCdc14A	NN	B-PN	PN
dynamically	RB	O	NONE
localizes	VBZ	B-LOC	LOC
to	TO	O	STOP
interphase	VB	B-PHAS	PHAS
but	CC	O	STOP
not	RB	O	STOP
mitotic	JJ	B-PHAS	PHAS
centrosomes	NNS	B-CN	CN
,	,	O	NONE
and	CC	O	STOP
hCdc14B	SYM	B-PN	PN
localizes	VBZ	B-LOC	LOC
to	TO	O	STOP
the	DT	O	STOP
interphase	NN	B-PHAS	PHAS
nucleolus	NN	B-CN	CN
.	.	O	NONE

Table F.3: Four column-output from the C&C tagger for the fourth sentence of the abstract of article [KZCJ02]

Bibliography

- [ABB⁺00] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.
- [AGH⁺08] Beatrice Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. Assisted Curation: Does Text Mining Really Help? In *Proceedings of the Pacific Symposium on Biocomputing*, Kohala Coast, Hawaii, USA, Jan 2008.
- [AHG07] Beatrice Alex, Barry Haddow, and Claire Grover. Recognising nested named entities in biomedical text. In *Proceedings of BioNLP*, pages 65–72, Prague, Czech Republic, June 2007.
- [ALF⁺02] Jens S Andersen, Carol E Lyon, Archa H Fox, Anthony K L Leung, Yun Wah Lam, Hanno Steen, Matthias Mann, and Angus I Lamond. Directed proteomic analysis of the human nucleolus. *Curr Biol*, 12(1):1–11, Jan 2002.
- [Aro01] A. R. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings AMIA Symp*, pages 17–21, La Jolla, California, 2001.
- [Att02] Teresa K Attwood. The PRINTS database: a resource for identification of protein families. *Briefings in Bioinformatics*, 3(3):252–263, Sep 2002.
- [BA96] A. Bairoch and R. Apweiler. The Swiss-Prot protein sequence data bank and its new supplement TrEMBL. *Nucleic Acids Res*, 24(1):21–25, Jan 1996.
- [BBA⁺03] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O’Donovan, Isabelle Phan, Sandrine Pilbout, and Michel Schneider. The Swiss-

- Prot protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res*, 31(1):365–370, Jan 2003.
- [BCW06] Ted Briscoe, John Carroll, and Rebecca Watson. The Second Release of the RASP System. In *Proceedings of the COLING/ACL Interactive Presentation Sessions*, pages 77–80, Sydney, Australia, 2006.
- [BKML⁺05] Dennis A Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and David L Wheeler. GenBank. *Nucleic Acids Res*, 33(Database issue):D34–D38, Jan 2005.
- [BKW⁺77] F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, E.F. Meyer Jr, M.D. Brice, J.R. Rodgers, O. Kennard, and M. Tasumi T. Shimanouchi. The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535–542, 1977.
- [Bli08] John Blitzer. *Domain adaptation of natural language processing systems*. PhD thesis, University of Pennsylvania, 2008.
- [BLJ⁺08] William A Baumgartner, Zhiyong Lu, Helen L Johnson, J. Gregory Caporaso, Jesse Paquette, Anna Lindemann, Elizabeth K White, Olga Medvedeva, K. Bretonnel Cohen, and Lawrence Hunter. Concept recognition for extracting protein interaction relations from biomedical text. *Genome Biol*, 9 Suppl 2:S9, 2008.
- [BLKV05] Christian Blaschke, Eduardo Andres Leon, Martin Krallinger, and Alfonso Valencia. Evaluation of BioCreAtIvE assessment of task 2. *BMC Bioinformatics*, 6 Suppl 1:S16, 2005.
- [BMP06] John Blitzer, Ryan Mcdonald, and Fernando Pereira. Domain Adaptation with Structural Correspondence Learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July 2006.
- [BNSH03] G Bhalotia, P Nakov, A Schwartz, and M Hearst. Biotext team report for the TREC 2003 genomics track. In *The Twelfth Text REtrieval Conference (TREC 2003) Notebook*, pages 612–621, Gaithersburg, Maryland, USA, 2003.
- [Bra97] A. P. Bradley. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30(6):1145–1159, 1997.
- [BvKNL07] Francois-Michel Boisvert, Silvana van Koningsbruggen, Joaquin Navascues, and Angus I Lamond. The multifunctional nucleolus. *Nat Rev Mol Cell Biol*, 8(7):574–585, Jul 2007.

- [BWF⁺00] Helen M. Berman, John Westbrook, Zukang Feng, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucl. Acids Res.*, 28:235–242, 2000.
- [Can04] Catherine Canevet. Towards automating the curation decision for the Nuclear Protein Database (NPD). Master’s thesis, University of Edinburgh, September 2004.
- [CBLJ04] David P A Corney, Bernard F Buxton, William B Langdon, and David T Jones. BioRAT: extracting biological information from full-length papers. *Bioinformatics*, 20(17):3206–3213, Nov 2004.
- [CC03a] Kuo-Chen Chou and Yu-Dong Cai. Prediction and classification of protein sub-cellular location-sequence-order effect and pseudo amino acid composition. *J Cell Biochem*, 90(6):1250–1260, Dec 2003.
- [CC03b] James R. Curran and Stephen Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of HLT-NAACL*, pages 164–167, Edmonton, Canada, 2003.
- [CDG⁺03] Peter Claus, Friederike Doring, Susanne Gringel, Frauke Muller-Ostermeyer, Jutta Fuhlrott, Theresia Kraft, and Claudia Grothe. Differential intranuclear localization of fibroblast growth factor-2 isoforms and specific interaction with the survival of motoneuron protein. *J Biol Chem*, 278(1):479–485, Jan 2003.
- [CKY⁺08] Dean Cheng, Craig Knox, Nelson Young, Paul Stothard, Sambasivarao Damaraju, and David S Wishart. PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Res*, 36(Web Server issue):W399–W405, Jul 2008.
- [CO08] Ali Cakmak and Gultekin Ozsoyoglu. Discovering gene annotations in biomedical text databases. *BMC Bioinformatics*, 9:143, 2008.
- [Con03] FlyBase Consortium. The FlyBase database of the Drosophila genome projects and community literature. *Nucleic Acids Res*, 31(1):172–175, Jan 2003.
- [Con07] UniProt Consortium. The Universal Protein Resource (UniProt). *Nucleic Acids Res*, 35(Database issue):D193–D197, Jan 2007.
- [CSA02] Jeffrey T Chang, Hinrich Schutze, and Russ B Altman. Creating an online dictionary of abbreviations from MEDLINE. *J Am Med Inform Assoc*, 9(6):612–620, 2002.

- [CSC05] Francisco M Couto, Mrio J Silva, and Pedro M Coutinho. Finding genomic ontology terms in text using evidence content. *BMC Bioinformatics*, 6 Suppl 1:S21, 2005.
- [CSK98] J. F. Caceres, G. R. Sreaton, and A. R. Krainer. A specific subset of SR proteins shuttles continuously between the nucleus and the cytoplasm. *Genes Dev*, 12(1):55–66, Jan 1998.
- [CWB08] Catherine Canevet, Bonnie Webber, and Wendy A. Bickmore. The Nuclear Protein Database (NPD) Curator System Interface. In *Proceedings of ISMB BioLINK SIG Meeting: Linking Literature, Information and Knowledge for Biology*, Toronto, Canada, 2008.
- [CY04] Jung H. Chiang and Hsu C. Yu. Extracting functional annotations of proteins based on hybrid text mining approaches. In *Proceedings of BioCreative*, Granada, Spain, 2004.
- [DA05a] Anna Divoli and Teresa K Attwood. BioIE: extracting informative sentences from the biomedical literature. *Bioinformatics*, 21(9):2138–2139, May 2005.
- [DA05b] Anna Divoli and Teresa K Attwood. Protein family databases: text-mining & annotation. In *Proceedings of ISMB*, Detroit, Michigan, USA, June 2005.
- [DFB03] Graham Dellaire, Rachel Farrall, and Wendy A. Bickmore. The Nuclear Protein Database (NPD): sub-nuclear localisation and functional annotation of the nuclear proteome. *Nucleic Acids Res*, 31(1):328–330, Jan 2003.
- [DHS⁺06] Tom P J Dunkley, Svenja Hester, Ian P Shadforth, John Runions, Thilo Weimar, Sally L Hanton, Julian L Griffin, Conrad Bessant, Federica Brandizzi, Chris Hawes, Rod B Watson, Paul Dupree, and Kathryn S Lilley. Mapping the Arabidopsis organelle proteome. *PNAS*, 103(17):6518–6523, Apr 2006.
- [DMdB⁺03] Ian Donaldson, Joel Martin, Berry de Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D Bader, Katerina Michalickova, Tony Pawson, and Christopher W V Hogue. PreBIND and Textomy—mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4:11, Mar 2003.
- [DMO00] M. Dundr, T. Misteli, and M. O. Olson. The dynamics of postmitotic reassembly of the nucleolus. *J Cell Biol*, 150(3):433–446, Aug 2000.

- [DO98] M. Dundr and M. O. Olson. Partially processed pre-rRNA is preserved in association with processing components in nucleolus-derived foci during mitosis. *Mol Biol Cell*, 9(9):2407–2422, Sep 1998.
- [Ent] Entrez cross-database. Available from <http://www.ncbi.nlm.nih.gov/sites/gquery>, May 2009.
- [EYD04] Sergei Egorov, Anton Yuryev, and Nikolai Daraselia. A simple and practical dictionary-based approach for identification of proteins in Medline abstracts. *J Am Med Inform Assoc*, 11(3):174–178, 2004.
- [EZB⁺04] David Eichmann, Yi Zhang, Shannon Bradshaw, Xin Ying Qiu, Li Zhou, Padmini Srinivasan, Aditya Kumar Sehgal, and Hudon Wong. Novelty, Question Answering and Genomics: The University of Iowa Response. In *The 13th Text Retrieval Conference (TREC 2004) Notebook*, pages 71–81, Gaithersburg, Maryland, USA, 2004.
- [FAD⁺06] J. Lynn Fink, Rajith N Aturaliya, Melissa J Davis, Fasheng Zhang, Kelly Hanson, Melvena S Teasdale, Chikatoshi Kai, Jun Kawai, Piero Carninci, Yoshihide Hayashizaki, and Rohan D Teasdale. LOCATE: a mouse protein subcellular localization database. *Nucleic Acids Res*, 34(Database issue):D213–D217, Jan 2006.
- [Fel98] Christiane Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
- [FKWB08] J. Lynn Fink, Sergey Kushch, Parker R. Williams, and Philip E. Bourne. BioLit: integrating biological literature with databases. *Nucl. Acids Res.*, pages gkn317+, 2008.
- [GAET06] Ben N G Giepmans, Stephen R Adams, Mark H Ellisman, and Roger Y Tsien. The fluorescent toolbox for assessing protein location and function. *Science*, 312(5771):217–224, Apr 2006.
- [Gas06] Caroline Gasperin. Semi-supervised anaphora resolution in biomedical texts. In *Proceedings of BioNLP*, pages 96–103, New York City, USA, 2006.
- [GGLZ02] MM Ghanem, Y Guo, H Lodhi, and Y Zhang. Automatic scientific text classification using local patterns: KDD Cup 2002 (task 1). *SIGKDD Explorations newsletter*, 4(2):95–96, 2002.

- [GHJS04] Tao Guo, Sujun Hua, Xinglai Ji, and Zhirong Sun. DBSubLoc: database of protein subcellular localization. *Nucleic Acids Res*, 32(Database issue):D122–D124, Jan 2004.
- [GHMD08] Tudor Groza, Siegfried Handschuh, Knud Möller, and Stefan Decker. KonneXSALT: First Steps Towards a Semantic Claim Federation Infrastructure. In *Proceedings of the 5th European Semantic Web Conference (ESWC)*, volume 5021 of *Lecture Notes in Computer Science*, pages 80–94, Tenerife, Canary Islands, Spain, 2008. Springer.
- [GKRS05] S. Gaudan, H. Kirsch, and D. Rebholz-Schuhmann. Resolving abbreviations to their senses in Medline. *Bioinformatics*, 21(18):3658–3664, Sep 2005.
- [GMH⁺07] Tudor Groza, Knud Möller, Siegfried Handschuh, Diana Trif, and Stefan Decker. SALT: Weaving the Claim Web. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pages 197–210, Busan, South Korea, 2007.
- [GMT06] Claire Grover, Michael Matthews, and Richard Tobin. Tools to Address the Interdependence between Tokenisation and Standoff Annotation. In *Proceedings of NLPXML-2006 (Multi-dimensional Markup in Natural Language Processing)*, pages 19–26, 2006.
- [Har93] Donna Harman. Overview of the first text retrieval conference (TREC). In *Proceedings of SIGIR*, pages 36–47, Pittsburgh, PA, USA, 1993.
- [Har02] Donna Harman. Overview of the TREC 2002 Novelty Track. In *The Eleventh Text REtrieval Conference (TREC 2002) Notebook, NIST Special Publication 500-251*, pages 46–55, Gaithersburg, Maryland, USA, 2002.
- [HB03] William Hersh and Ravi Teja Bhupatiraju. TREC 2003 genomics track overview. In *The Twelfth Text REtrieval Conference (TREC 2003) Notebook*, Gaithersburg, Maryland, USA, 2003.
- [HBR⁺04] W Hersh, R Bhupatiraju, L Ross, P Johnson, A Cohen, and D Kraemer. TREC 2004 genomics track overview. In *The Thirteenth Text REtrieval Conference (TREC 2004) Notebook*, Gaithersburg, Maryland, USA, 2004.
- [HCY⁺05] W Hersh, A Cohen, J Yang, R Bhupatiraju, P Roberts, and M Hearst. TREC 2005 genomics track overview. In *The Fourteenth Text REtrieval Conference (TREC 2005) Notebook*, pages 14–25, Gaithersburg, Maryland, USA, 2005.

- [HDG⁺07] Marti A Hearst, Anna Divoli, Harendra Guturu, Alex Ksikes, Preslav Nakov, Michael A Wooldridge, and Jerry Ye. BioText Search Engine: beyond abstract search. *Bioinformatics*, 23(16):2196–2197, Aug 2007.
- [HL93] B. L. Humphreys and D. A. Lindberg. The UMLS project: making the conceptual connection between users and the information they need. *Bull Med Libr Assoc*, 81(2):170–177, April 1993.
- [HM07] Barry Haddow and Michael Matthews. The Extraction of Enriched Protein-Protein Interactions from Biomedical Text. In *Proceedings of BioNLP*, pages 145–152, Prague, Czech Republic, June 2007.
- [Hof08] Robert Hoffmann. A wiki for the life sciences where authorship matters. *Nat Genet*, 40(9):1047–1051, 2008.
- [HS05] Oliver Hofmann and Dietmar Schomburg. Concept-based annotation of enzyme classes. *Bioinformatics*, 21(9):2059–2066, May 2005.
- [HV04] Robert Hoffmann and Alfonso Valencia. A gene network for navigating the literature. *Nat Genet*, 36(7):664, Jul 2004.
- [HVT⁺07] Joshua L Heazlewood, Robert E Verboom, Julian Tonti-Filippini, Ian Small, and A. Harvey Millar. SUBA: the Arabidopsis Subcellular Database. *Nucleic Acids Res*, 35(Database issue):D213–D218, Jan 2007.
- [HW04] Karsten Hokamp and Kenneth H Wolfe. PubCrawler: keeping up comfortably with PubMed and GenBank. *Nucleic Acids Res*, 32(Web Server issue):W16–W19, Jul 2004.
- [JC06] Jiang Jing and Zhai ChengXiang. Exploiting domain structure for named entity recognition. In *Proceedings of HLT-NAACL*, pages 74–81, New York, New York, 2006.
- [JHHC02] Chiang Jung-Hsien and Yu Hsu-Chun. MeKE: discovering the functions of gene products from biomedical literature via sentence alignment. *Bioinformatics*, 19:1417–1422, 2002.
- [Kav08] Silvestras Kavaliauskas. PaperMaker: Consistency Analysis of Scientific Biomedical Literature. MSc in Life Science Informatics, Rheinische Friedrich-Wilhelms-University of Bonn, The Faculty of Mathematics and Natural Science, December 2008.

- [KBP08] F. Kitoogo, V. Baryamureeba, and G. De Pauw. Towards Domain Independent Named Entity Recognition. *Strengthening the Role of ICT in Development*, pages 38–49, 2008.
- [KID08] Trish E Kaiser, Robert V Intine, and Miroslav Dunder. De novo formation of a subnuclear body. *Science*, 322(5908):1713–1717, Dec 2008.
- [KLRPV08] Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biol*, 9 Suppl 2:S4, 2008.
- [KLS⁺07] Nikiforos Karamanis, Ian Lewin, Ruth Seal, Rachel Drysdale, and Ted Briscoe. Integrating Natural Language Processing with FlyBase Curation. In *Pacific Symposium on Biocomputing*, pages 245–256, Grand Wailea, Maui, Hawaii, 2007.
- [KMAM07] Jee-Hyub Kim, Alex Mitchell, Teresa K Attwood, and Hilario Melanie. Learning to extract relations for protein annotation. In *Proceedings of ISMB/ECCB*, volume 23, pages i256–i263, Vienna, Austria, 2007.
- [KPV05] Martin Krallinger, Maria Padron, and Alfonso Valencia. A sentence sliding window approach to extract protein annotations from biomedical articles. *BMC Bioinformatics*, 6 Suppl 1:S19, 2005.
- [KSL⁺08] Nikiforos Karamanis, Ruth Seal, Ian Lewin, Peter McQuilton, Andreas Vlachos, Caroline Gasperin, Rachel Drysdale, and Ted Briscoe. Natural language processing in aid of FlyBase curators. *BMC Bioinformatics*, 9:193, 2008.
- [KZCJ02] Brett K Kaiser, Zachary A Zimmerman, Harry Charbonneau, and Peter K Jackson. Disruption of centrosome structure, chromosome segregation, and cytokinesis by misexpression of human Cdc14A phosphatase. *Mol Biol Cell*, 13(7):2289–2300, Jul 2002.
- [Le03] Zhang Le. MaxEnt toolkit. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html, September 2003.
- [LEG⁺09] Eduardo Andres Leon, Iakes Ezkurdia, Beatriz Garca, Alfonso Valencia, and David Juan. EcID. A database for the inference of functional interactions in *E. coli*. *Nucleic Acids Res*, 37(Database issue):D629–D635, Jan 2009.
- [Lew07] Ian Lewin. Using hand-crafted rules and machine learning to infer SciXML document structure. In *Proceedings of 7th E-Science All Hands Meeting*, Nottingham, UK, 2007.

- [LHZW06] Hongfang Liu, Zhang-Zhi Hu, Jian Zhang, and Cathy Wu. BioThesaurus: a Web-based thesaurus of protein and gene names. *Bioinformatics*, 22(1):103–105, Jan 2006.
- [LJN⁺04] Fang Liu, Tor-Kristian Jenssen, Vegard Nygaard, John Sack, and Eivind Hovig. FigSearch: a figure legend indexing and classification system. *Bioinformatics*, 20(16):2880–2882, Nov 2004.
- [LS02] Ding-Yen Lin and Hsiu-Ming Shih. Essential role of the 58-kDa microspherule protein in the modulation of Daxx-dependent transcriptional repression as revealed by nucleolar sequestration. *J Biol Chem*, 277(28):25446–25456, Jul 2002.
- [LTML05] Yun Wah Lam, Laura Trinkle-Mulcahy, and Angus I Lamond. The nucleolus. *J Cell Sci*, 118(Pt 7):1335–1337, Apr 2005.
- [MDK⁺05] A. L. Mitchell, A. Divoli, J-H. Kim, M. Hilario, I. Selimas, and T. K. Attwood. METIS: multiple extraction techniques for informative sentences. *Bioinformatics*, 21(22):4196–4197, Nov 2005.
- [med] MEDLINE. Available from <http://medline.cos.com/>, May 2009.
- [MKC⁺05] Karim Mekhail, Mireille Khacho, Amanda Carrigan, Robert R J Hache, Lakshman Gunaratnam, and Stephen Lee. Regulation of ubiquitin ligase dynamics by the nucleolus. *J Cell Biol*, 170(5):733–744, Aug 2005.
- [MLW⁺08] Alexander A Morgan, Zhiyong Lu, Xinglong Wang, Aaron M Cohen, Juliane Fluck, Patrick Ruch, Anna Divoli, Katrin Fundel, Robert Leaman, Jrg Hakenberg, Chengjie Sun, Heng hui Liu, Rafael Torres, Michael Krauthammer, William W Lau, Hongfang Liu, Chun-Nan Hsu, Martijn Schuemie, K. Bretonnel Cohen, and Lynette Hirschman. Overview of BioCreative II gene normalization. *Genome Biol*, 9 Suppl 2:S3, 2008.
- [MNM⁺00] K. Misawa, T. Nosaka, S. Morita, A. Kaneko, T. Nakahata, S. Asano, and T. Kitamura. A method to identify cDNAs based on localization of green fluorescent protein fusion products. *PNAS*, 97(7):3062–3066, Mar 2000.
- [MRA03] A. L. Mitchell, J. R. Reich, and T. K. Attwood. PRECIS: protein reports engineered from concise information in Swiss-Prot. *Bioinformatics*, 19(13):1664–1671, Sep 2003.
- [MSK⁺06] Gopa R Mishra, M. Suresh, K. Kumaran, N. Kannabiran, Shubha Suresh, P. Bala, K. Shivakumar, N. Anuradha, Raghunath Reddy, T. Madhan Raghavan, Shalini

- Menon, G. Hanumanthu, Malvika Gupta, Sapna Upendran, Shweta Gupta, M. Mahesh, Bincy Jacob, Pinky Mathew, Pritam Chatterjee, K. S. Arun, Salil Sharma, K. N. Chandrika, Nandan Deshpande, Kshitish Palvankar, R. Raghav-nath, R. Krishnakanth, Hiren Karathia, B. Rekha, Rashmi Nayak, G. Vish-nupriya, H. G Mohan Kumar, M. Nagini, G. S Sameer Kumar, Rojan Jose, P. Deepthi, S. Sujatha Mohan, T. K B Gandhi, H. C. Harsha, Krishna S Desh-pande, Malabika Sarker, T. S Keshava Prasad, and Akhilesh Pandey. Hu-man protein reference database–2006 update. *Nucleic Acids Res*, 34(Database issue):D411–D414, Jan 2006.
- [Nie06] Leif Arda Nielsen. Extracting protein-protein interactions using simple contex-tual features. In *Proceedings of BioNLP*, pages 120–121, New York City, USA, June 2006.
- [NLM99] Kamal Nigam, John Lafferty, and Andrew McCallum. Using Maximum Entropy for Text Classification. In *IJCAI Workshop on Machine Learning for Information Filtering*, pages 61–67, Stockholm, Sweden, 1999.
- [OA06] Naoaki Okazaki and Sophia Ananiadou. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, October 2006.
- [OGS⁺99] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*, 27(1):29–34, Jan 1999.
- [OMI] OMIM: Online Mendelian Inheritance in Man. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD). Available from <http://www.ncbi.nlm.nih.gov/omim/>, May 2009.
- [OSWG02] Martin Offterdinger, Christian Schofer, Klara Weipoltshammer, and Thomas W Grunt. c-erbB-3: a nuclear protein in mammary epithelial cells. *J Cell Biol*, 157(6):929–939, Jun 2002.
- [PCC⁺01] J. Pustejovski, J. Castano, M. Cochran, M. Kotecki, M. Morrell, and A. Rumshisky. Linguistic Knowledge Extraction from Medicine: Automatic Construction of an Acronym Database. *Medinfo*, 2001.
- [PDF00] V. Pogaci, F. Dragon, and W. Filipowicz. Human H/ACA small nucleolar RNPs and telomerase share evolutionarily conserved proteins NHP2 and NOP10. *Mol Cell Biol*, 20(23):9028–9040, Dec 2000.

- [PM01] Kim D. Pruitt and Donna R. Maglott. RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Res*, 29(1):137–140, Jan 2001.
- [pmc] PubMed Central. Available from <http://www.pubmedcentral.nih.gov/>, May 2009.
- [PPRMV04] Philip Ian Padilla, Gustavo Pacheco-Rodriguez, Joel Moss, and Martha Vaughan. Nuclear localization and molecular partners of BIG1, a brefeldin A-inhibited guanine nucleotide-exchange protein for ADP-ribosylation factors. *PNAS*, 101(9):2752–2757, Mar 2004.
- [PTM05] Kim D. Pruitt, Tatiana Tatusova, and Donna R. Maglott. NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucl. Acids Res.*, 33:501–504, 2005.
- [Pub] PubMed. Available from <http://www.pubmed.gov/>, May 2009.
- [Qui93] Ross J. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.
- [RAG⁺05] Sebastien Rey, Michael Acab, Jennifer L Gardy, Matthew R Laird, Katalin de-Fays, Christophe Lambert, and Fiona S L Brinkman. PSORTdb: a protein subcellular localization database for bacteria. *Nucleic Acids Res*, 33(Database issue):D164–D168, Jan 2005.
- [RAI] Rainbow: statistical text classification. Available from <http://www-2.cs.cmu.edu/~mccallum/bow/rainbow/>, May 2009.
- [RFLF02] Yizhar Regev, Michal Finkelstein-Landeau, and Ronen Feldman. Rule-based extraction of experimental evidence in the biomedical domain—the KDD Cup 2002 (task 1). *SIGKDD Explorations newsletter*, 4(2):90–92, 2002.
- [RJ88] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. *Document retrieval systems*, pages 143–160, 1988.
- [RRB⁺03] Tim Raemaekers, Katharina Ribbeck, Joel Beaudouin, Wim Annaert, Mark Van Camp, Ingrid Stockmans, Nico Smets, Roger Bouillon, Jan Ellenberg, and Geert Carmeliet. NuSAP, a novel microtubule-associated protein involved in mitotic spindle organization. *J Cell Biol*, 162(6):1017–1029, Sep 2003.
- [RRP80] S. Robertson, C. Rijsbergen, and M. Porter. Probabilistic models of indexing and searching. In *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56, Cambridge, England, 1980.

- [RSKA⁺06] Dietrich Rebholz-Schuhmann, Harald Kirsch, Miguel Arregui, Sylvain Gaudan, Mark Rynbeek, and Peter Stoehr. Protein annotation by EBIMed. *Nat Biotechnol*, 24(8):902–903, Aug 2006.
- [RSKA⁺07] Dietrich Rebholz-Schuhmann, Harald Kirsch, Miguel Arregui, Sylvain Gaudan, Mark Riethoven, and Peter Stoehr. EBIMed–text crunching to gather facts for proteins from Medline. *Bioinformatics*, 23(2):e237–e244, Jan 2007.
- [RSKC05] Dietrich Rebholz-Schuhmann, Harald Kirsch, and Francisco Couto. Facts from text–is text mining ready to deliver? *PLoS Biol*, 3(2):e65, Feb 2005.
- [SAR⁺07] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J. Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Nigam Shah, Patricia L. Whetzel, and Suzanna Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotech*, 25(11):1251–1255, November 2007.
- [SCD⁺02] Alexander Scherl, Yohann Cout, Catherine Don, Aleth Call, Karine Kindbeiter, Jean-Charles Sanchez, Anna Greco, Denis Hochstrasser, and Jean-Jacques Diaz. Functional proteomic analysis of human nucleolus. *Mol Biol Cell*, 13(11):4100–4109, Nov 2002.
- [SEM⁺02] M Shi, DS Edwin, R Menon, L Shen, JYK Lim, and HT Loh. A machine learning approach for the curation decision of biomedical literature–KDD Cup 2002 (task 1). *SIGKDD Explorations newsletter*, 4(2):93–94, 2002.
- [SH03] Ian Soboroff and Donna Harman. Overview of the TREC 2003 novelty track. In *The Twelfth Text Retrieval Conference (TREC 2003) Notebook*, pages 38–53, Gaithersburg, Maryland, USA, 2003.
- [SMB⁺09] Peter Stenson, Matthew Mort, Edward Ball, Katy Howells, Andrew Phillips, Nick Thomas, and David Cooper. The Human Gene Mutation Database: 2008 update. *Genome Med*, 1(1):13, Jan 2009.
- [SMBP98] J. Schultz, F. Milpetz, P. Bork, and C. P. Ponting. SMART, a simple modular architecture research tool: identification of signaling domains. *PNAS*, 95(11):5857–5864, May 1998.
- [Sob04] Ian Soboroff. Overview of the TREC 2004 novelty track. In *The Thirteenth Text Retrieval Conference (TREC 2004) Notebook*, pages 57–70, Gaithersburg, Maryland, USA, 2004.

- [SPL00] Z. Strezoska, D. G. Pestov, and L. F. Lau. Bop1 is a mouse WD40 repeat nucleolar protein involved in 28S and 5.8S rRNA processing and 60S ribosome biogenesis. *Mol Cell Biol*, 20(15):5516–5528, Aug 2000.
- [SSE⁺98] A. H. Stegh, O. Schickling, A. Ehret, C. Scaffidi, C. Peterhansel, T. G. Hofmann, I. Grummt, P. H. Krammer, and M. E. Peter. DEDD, a novel death effector domain-containing protein, targeted to the nucleolus. *EMBO J*, 17(20):5974–5986, Oct 1998.
- [SSS⁺05] Keiko Shimono, Yohei Shimono, Kaoru Shimokata, Naoki Ishiguro, and Masahide Takahashi. Microspherule protein 1, Mi-2beta, and RET finger protein associate in the nucleolus and up-regulate ribosomal gene transcription. *J Biol Chem*, 280(47):39436–39447, Nov 2005.
- [STMA08] Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught, and Sophia Ananiadou. How to make the most of NE dictionaries in statistical NER. *BMC Bioinformatics*, 9 Suppl 11:S5, 2008.
- [STnA⁺08] Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A Struble, Richard J Povinelli, Andreas Vlachos, William A Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maa-Lpez, Jacinto Mata, and W. John Wilbur. Overview of BioCreative II gene mention recognition. *Genome Biol*, 9 Suppl 2:S2, 2008.
- [SWJ⁺00] S. Snaar, K. Wiesmeijer, A. G. Jochemsen, H. J. Tanke, and R. W. Dirks. Mutational analysis of fibrillarin and its mobility in living human cells. *J Cell Biol*, 151(3):653–662, Oct 2000.
- [SWS⁺04] M. J. Schuemie, M. Weeber, B. J A Schijvenaars, E. M. van Mulligen, C. C. van der Eijk, R. Jelier, B. Mons, and J. A. Kors. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, 20(16):2597–2604, Nov 2004.
- [SZK⁺06] Yuan Shen, Gabriel Zaccak, Boris Katz, Yuan Luo, and Ozlem Uzuner. Duplicate Removal for Candidate Answer Sentences. *Proceedings of the First CSAIL Student Workshop (CSW)*, September 2006.

- [TML07] Laura Trinkle-Mulcahy and Angus I Lamond. Toward a high-resolution view of nuclear dynamics. *Science*, 318(5855):1402–1407, Nov 2007.
- [TTA08] Yoshimasa Tsuruoka, Jun’ichi Tsujii, and Sophia Ananiadou. FACTA: a text search engine for finding associated biomedical concepts. *Bioinformatics*, 24(21):2559–2560, Nov 2008.
- [Vla07] A. Vlachos. Evaluating and combining biomedical named entity recognition systems. In *Proceedings of BioNLP*, pages 199–206, Prague, Czech Republic, 2007.
- [WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, second edition, June 2005.
- [WKG⁺06] David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res*, 34(Database issue):D668–D672, Jan 2006.
- [WLDP02] Hester M. Wain, Michael Lush, Fabrice Ducluzeau, and Sue Povey. Genew: the human gene nomenclature database. *Nucleic Acids Res*, 30(1):169–171, Jan 2002.
- [WP] Dominic Widdows and Stanley Peters. Infomap. Available from <http://infomap-nlp.sourceforge.net/>, May 2009.
- [WTK⁺07] David S Wishart, Dan Tzur, Craig Knox, Roman Eisner, An Chi Guo, Nelson Young, Dean Cheng, Kevin Jewell, David Arndt, Summit Sawhney, Chris Fung, Lisa Nikolai, Mike Lewis, Marie-Aude Coutouly, Ian Forsythe, Peter Tang, Savita Shrivastava, Kevin Jeroncic, Paul Stothard, Godwin Amegbey, David Block, David D Hau, James Wagner, Jessica Miniaci, Melisa Clements, Mulu Gebremedhin, Natalie Guo, Ying Zhang, Gavin E Duggan, Glen D Macinnis, Alim M Weljie, Reza Dowlatabadi, Fiona Bamforth, Derrick Clive, Russ Greiner, Liang Li, Tom Marrie, Brian D Sykes, Hans J Vogel, and Lori Querengesser. HMDB: the Human Metabolome Database. *Nucleic Acids Res*, 35(Database issue):D521–D526, Jan 2007.
- [XML05] Huang X, Zhong M, and Si L. York university at TREC 2005: Genomics track. In *The Fourteenth Text REtrieval Conference (TREC 2005) Notebook*, Gaithersburg, Maryland, USA, 2005.

- [YHM03] Alexander S Yeh, Lynette Hirschman, and Alexander A Morgan. Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup. *Bioinformatics*, 19 Suppl 1:i331–i339, 2003.