EPFL semester project
Extraction of Protein Concentration from Scientific Literature

Philémon Favrod

Supervisors: Jean-Cédric Chappelier, Renaud Richardet

January 10, 2014

Contents

Chapter 1

Introduction

1.1 Motivation

A usual start for a paper in the field of Information Extraction (IE) is the assessment of the increasing difficulty to access data hidden, for instance, in texts or, more generally, in unstructured form. The growth of scientific publication is exponential. For instance, the world-known biomedical citation database, Pubmed, contains over 20-millions (2010) citations while it contained less than 8-millions citations in the mid-eighties[1]. That is, looking for very specific data is a problem that can only become harder. For instance, the use of automated techniques such as the ones coming from the natural language processing (NLP) are needed to keep the access time to specific data realistic. The Blue Brain project[2] (BBP) where simulation of biological models requires a lot of very precise data is not a departure from that norm. In the latter context, the researchers' need to access information concerning protein concentration at different scale in the brain is the main motivation of this project. In the point of view of the Computer scientist, as described later, it is also a trial to extract very specific pieces of data using simple extraction model. Therefore, the results of this approach is valuable for future research in that field even though its very specialized character limits the significance of generalizing to other fields.

1.2 Project description

The project falls within the line of successive works in natural language processing at the Blue Brain project. For two years, IE tools have been developed for different purposes, mainly to achieve named entity recognition (NER) for neuroscience extending the Apache UIMA framework[3] and giving birth to the Bluima toolkit. Henceforth, using these tools to perform Relation Extraction (RE) and

[1] `http://database.oxfordjournals.org/content/2011/baq036.full`
[2] `http://bluebrain.epfl.ch`
[3] `http://uima.apache.org`

construct accessible databases for researchers is the ultimate purpose in which this project take place. Taking advantage of building upon this past work, this semester work focuses on building an analysis chain to put in relation mentions of proteins, concentrations and their local context such as brain regions, cell types or subcells. In more practical terms, this project is about building a ready-to-use UIMA analysis chain to extract triplets containing a protein mention, its associated concentration and the biological location of those two. In the next chapters, the term "triplet" will be used to refer to this kind of piece of data.

In order to more precisely describe the problem, we will consider an example. Consider the following sentence [PMID: 15659591]:

> Using this calibration procedure, we find that mature granule cells (doublecortin-) contain approximately 40 μM, and newborn granule cells (doublecortin+) contain 0-20 μM calbindin-D28k.

The latter describes a strong relation in the sense that it establishes how concentrated are "calbindin-D28k" proteins in "granule cells". Nevertheless, this relation is mentionned in an unstructured form. This information would be more accessible if it was in a more structured format such as:

Protein	**Concentration**	**Location**
calbindin-D28k	40 μM	mature granule cells (doublecortin-)
calbindin-D28k	0-20 μM	newborn granule cells (doublecortin+)

This transformation (from unstructured form to structured form) of such a relation is the main goal of this project. We will refer to this kind of transformation by calling it an extraction, as it is usually done in IE. The columns (protein, concentration and location) of the table are the named entities taking part in the relation. The above type of relation, describing how concentrated a protein is in a brain location, will be referred to as the targeted relation to distinguish it from other kind of relations that will appear in our results and because it is what the researchers are mainly looking for.

Chapter 2 will focus on the methodological aspect of this project by describing how the tool aiming to perform the above transformation was developed and how it works. Then, the results of large-scale extractions are analyzed in chapter 3. Finally, a conclusion can be found in chapter 5.

Chapter 2

The process: building the UIMA chain

This chapter intends to explain the design decisions which leads to the final tool developed during this project. First, a little digression is made in order to describe the main challenge(s) we had to face. Then, the method used to extract the text from PDF documents is explained. Finally, the entire UIMA analysis chain will be described component by component.

2.1 Facing the absence of any annotated corpus

The greatest obstacle this project has to overcome from the start is the absence of a reference gold standard, i.e. the absence of an already-annotated corpus. Annotated corpora for biomedical entity exist and some key components this project is built upon are themselves built on machine learning model and trained on such corpora (see sections 2.4, 2.6 or 2.7). However, it appears that no annotated corpus includes information regarding the targeted relation (see section 1.2). Furthermore, no annotated corpus seems to exist even for the sub-relation between a protein mention and a concentration mention. The work of annotating such entities in a corpus is outside the field of NLP and Computer science; it implies the involvement of individual(s) specialized in the field in question — (neuro)biologists in the present case. Due to the lack of annotated data, supervised machine learning cannot be considered as a design option. That is, the relation extraction processes designed along this project could only be rule-based heuristics or unsupervised methods.

Note that the absence of any such corpus also significantly affects the evaluation method. That is, even if we tried to be the more quantitative we could in our decision path, some decision had to be taken only on a totally qualitative basis. Another problem the lack of corpus induces in evaluation is the absence of idea concerning the recall of our heuristics since the denominator of the recall

formula remains unknown. Therefore, the chosen approach in this project was to incrementally improve the precision of our tools while qualitatively judging whether the decision taken were too specific or not.

2.2 Extracting the full text from PDF

Some considerations must be taken into account regarding the corpora that can be used and how they can. The NLP tasks conducted in the BBP are currently performed on either the abstracts of scientific publications or on the full-text PDFs of the scientific publications[1]. Preliminary analysis of abstracts quickly leads to the conclusion that a data as specific as the concentration of proteins hardly appears in abstracts. We therefore choose to run our tool on the full-text PDFs. Thus, "PDF reader" converting PDFs to raw text are used in this project. For that matter, we used the toolkit developed during another semester project [8] which is a version of PDFTextStream[2] specialized in the extraction of scientific content.

2.3 A quick overview of the analysis chain

Before describing each component of the UIMA chain, we will just look at the big picture of the chain shown in figure 2.1. As one could expect, the design of our chain follows the usual pattern used in relation extraction:

- the text is preprocessed simplifying it enough to make it suitable for the rest of the chain; this includes tokenization, POS tagging, lemmatization and chunking;
- the NERs are applied to annotate the underlying instance of the relation in question (here: protein, concentration and a brain location);
- the relation extraction algorithm is applied to those named entities;
- finally, some filtering is applied to clean up the results.

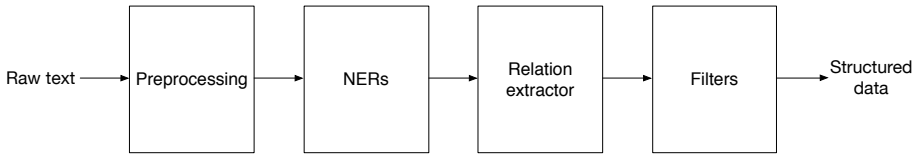


Figure 2.1: The big picture of the processing chain

[1] More precisely, the abstracts and PDFs from PubMed. See `http://www.ncbi.nlm.nih.gov/pubmed`.

[2] See `http://snowtide.com/`

2.4 Preprocessing

The first stage of the UIMA chain is to treat the text such that it can be used by usual NLP tools (see Figure 2.2). First, the text is sliced into sentences and words are tokenized. Then, each token is assigned its corresponding part-of-speech. Then, the sentences are syntactically chunked. All these tasks are achieved using the JulieLab NLP tool suite[3] which consists of UIMA wrappers for the Apache OpenNLP library[4]. Finally, the token are lemmatized using BioLemmatizer [7].

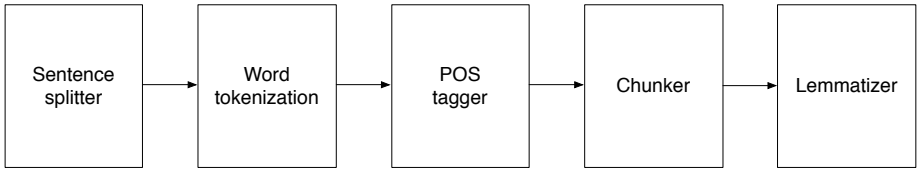
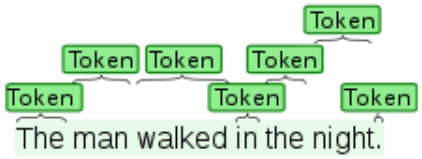


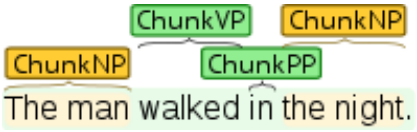
Figure 2.2: The different components of the preprocessing pipeline.

For example, given the sentence "The man walked in the night.", it is first recognized as a sentence by the JulieLab-OpenNLP Sentence Splitter. Then, its words are tokenized using again a JulieLab-OpenNLP tool. In our example, the sentence is simplified as a sequence of tokens as follows:



Then, each token is assigned its presumed grammatical role in the sentence (a part-of-speech tag) by the JulieLab-OpenNLP POS Tagger. For instance,

where `DT` stands for determinant, `NN` for a singular noun, `VBD` for a verb (past tense) and `IN` for a preposition. Then, this POS tags are grouped into grammatical chunks:



[3] `http://www.julielab.de/`
[4] `http://opennlp.apache.org/`

Finally, the BioLemmatizer [7] sets the lemmas of the token's instances. In the above example, it has the only effect to assign the lemma "walk" to the token "walked", the lemma of the other tokens being themselves.

2.5 The Named Entity Recognizers

The second stage of the UIMA pipeline consists in detecting the occurrence of the named entities which matter in the context of the project. Since the approach is very different depending on the named entity in question, each subsection below describes the model used to annotate a particular named entity.

2.5.1 Annotating the Concentrations

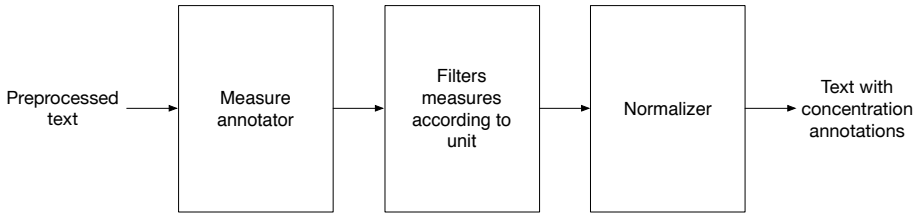


Figure 2.3: The overall pipeline for annotating concentrations

Concentrations constitutes obviously a class of named entity of great interest in this project. Such a class is a subset of a larger class, the measures for which the Bluima toolkit [10] provides a regexp-based annotator developed during another semester project [4] at the BBP to detect the mentions of any kind of measurements. It extracts the measures as a pair whose first element is the value and the second one is the unit. For instance, "1 μg/ml" is extracted as {1, "μg/ml"}.

The concentration NER is built upon this tools by filtering out the measures which have no unit or a unit which does not correspond to a concentration measurement. The unit in question includes measures of molarity, ratios between a measure of weight and a measure of volume and other specific concentration units.

2.5.2 Normalizing the Concentrations

The first results reveals the need to normalize the unit of concentration. Otherwise, for instance, "0.005 mg/mL insulin" would be considered a different concentration of insulin than "5 mg/L insulin" while, physically speaking,

$$0.005[\text{mg/mL}] = 5[\text{mg/L}].$$

To explain such a need, note that, the model of relation extraction being based on co-occurrences, the frequency of a co-occurrence is a valuable information

but, since measures of concentration are not necessarily normalized in the text, some concentrations may be considered as different while they are the same. To solve that matter, we consider three classes of concentrations units, namely

1. the units expressed in terms of molarity (molar, mol/m3 , etc.),
2. the units expressed as a ration of a weight and a volume (like mg/ml, kg/l, etc.) and
3. the other specialized units (like ppm, ppb, etc.)

After having characterized as concentration each measure whose unit is a member of one of those classes. The ones that can be normalized, namely classes 1 and 2, are normalized using a regexp-based algorithm. Every units in the first category can be converted to a similar unit. The most logical choice is the standard unit used among researchers, namely molar[5]. Note that the second class of units cannot be converted into the first one without any knowledge of the context, since it requires the knowledge of the molar mass to perform such a conversion[6]. However, such a class of unit can be normalized inside itself. Therefore, kg/m^3 was chosen as targeted unit when normalizing the second class of units.

The normalization is performed using a regex-based approach. First, each SI prefix (such as d-, nano- or μ-) is mapped to the power of 10 it corresponds to so that we can get standardized unit. For instance, the value of concentration whose unit are prefixed by μ- are multiplied by 10^6. Then, for the second class, we convert volume units (i.e. their denominator). In spite of its simplicity, this approach seems to result in good qualitative results. Table 2.1 shows some examples of unit transformation performed by the normalizer.

Before normalization	**After normalization**
10 nM	10^{-8} molar
10 mM	0.00001 molar
163 g/m^3	0.163 kg/m^3
245.6 ng/ml	0.0002456 kg/m^3

Table 2.1: Examples of unit normalization for concentrations

2.6 Annotating the Proteins

Another significant existing work this project need and is built upon is a protein NER. This section describes how we selected it among the two protein NERs provided by the Bluima toolkit [10] and how we adapted it to our needs.

[5] i.e. mol/L, sometimes abbreviated by M. See `http://en.wikipedia.org/wiki/Molar_concentration` for more details.

[6] It is outside the scope of this project, but this is not an unrealistic goal given an appropriate database to look up the molar masses.

2.6.1 BANNER

We first considered BANNER [9], a protein NER that has been adapted for UIMA. It is based on a machine learning model (Conditional Random Field).

BANNER tokenization weakness

First experimental results suggest that BANNER is likely to mistokenize protein mentions in presence of measure mentions. To illustrate the problem, consider the following sentence [PMID: 19535906]:

> Cells were lysed in a buffer containing 50 mM Tris at pH 7.5, 1% Triton X-100, 150 mM NaCl, 5 mM EDTA, 1 mg/ml pepstatin A, 1 μg/ml leupeptin, 2 μg/ml aprotinin, 1 mM PMSF, 0.1 mg/ml benz- 21. amidine and 8 g/ml each of calpain I and calpain II.

The result of BANNER annotation is:

> [...], 1 mg/ml $\underbrace{\text{pepstatin A, 1 } \mu\text{g}}_{\text{PROTEIN}}/\underbrace{\text{ml leupeptin}}_{\text{PROTEIN}}$, 2 μg/ml aprotinin, 1 mM PMSF, [...]

As one can see, the bounds of the annotations returned by BANNER exceeds the protein mentions. In other words, such annotations cover some part of the surrounding measures in the text. Note that some other protein mention such as "aprotinin" are not detected by BANNER.

2.6.2 Benchmark of BANNER against Gimli

The above-mentioned problem in BANNER leads us to study one of its competitor, Gimli [1], another machine-learning-based NER for the recognition of protein mentions. The quality of its output appeared qualitatively better than the one of BANNER; since no similar tokenization problem were found. However, it runs remarkably slowly compared to BANNER and it is only partially integrated with UIMA. A major argument in favor of Gimli was that its ability to do syntactical parsing, which could have been reused later. However, because of its incomplete UIMA integration, these data could not be trivially accessed and reused. Moreover, the incomplete integration of Gimli into UIMA seems buggy: it enters a dead lock after annotating about ten PDFs.

The boxplots on figure 2.4 compare BANNER and Gimli performances in terms of the number of documents processed by second. This data have been collected by profiling both annotators while they were analyzing PDFs articles. As one can see, according to this dataset, BANNER ($M = 0.094$ doc/s) is on average about three times quicker than Gimli ($M = 0.027$ doc/s). That is, BANNER analyzes about 338 documents in one hour on average while Gimli analyzes only about 97 of them. Considering the decrease in the runtime involved by the use of Gimli while running on a huge number of documents and the time spent to debug and to fully integrate it to UIMA leads us to keep BANNER as protein NER and to improve its results.

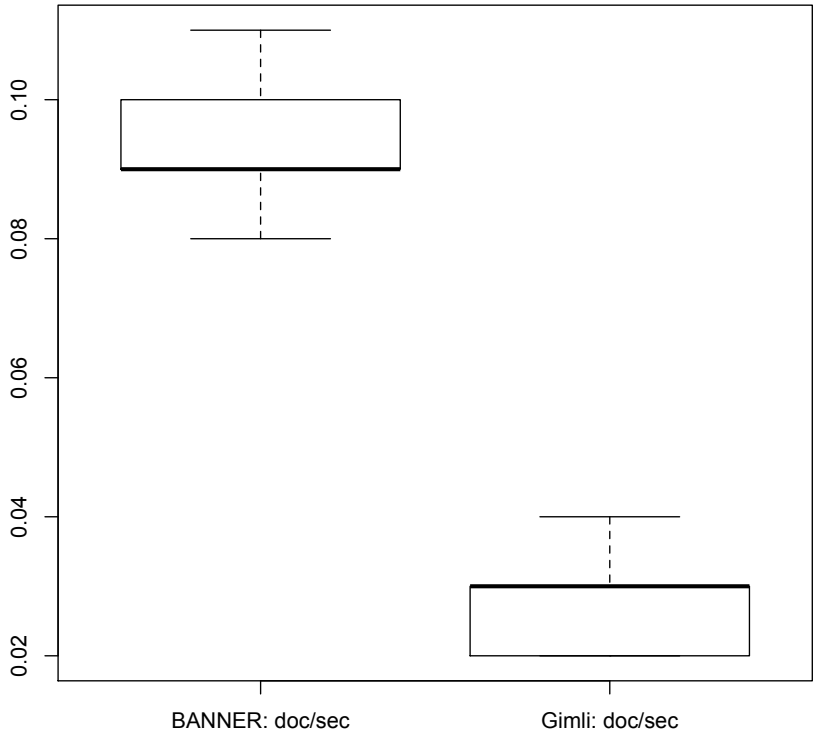


Figure 2.4: Comparative boxplot of speed performance between Banner and Gimli

2.6.3 BannerM

To overcome the problem of tokenization of protein mentions with BANNER, a wrapper was developed so that BANNER is fed with a cleaned input. That is, all measurement mentions detected in the text are removed[7] before BANNER analyzes the text. This tool will be referred to as BannerM where the "M" stands for "modified".

The surprising fact is that, once the measurements filtered out of the input, BANNER seems to recognize more protein. For instance, in the sentence mentionned above, the "aproptinin" mention is recognized as one. This is also

[7]More precisely, they are replaced by spaces.

reflected on larger datasets: when run on 238 PDFs, BannerM found 22613 distinct protein mentions while BANNER found only 22090. Moreover, they both recognize 19380 distinct common entities, but, for 18599 of them, BannerM found more occurrences. Finally, considering the tail of the distribution of the length of the protein mentions, BannerM appears to have a closer distribution to the one of GENIA [6], a reference protein-annotated corpus, (see section 2.6.4 for more information about GENIA) than BANNER which still have a lot of very long protein mentions as one can see on figure 2.5.

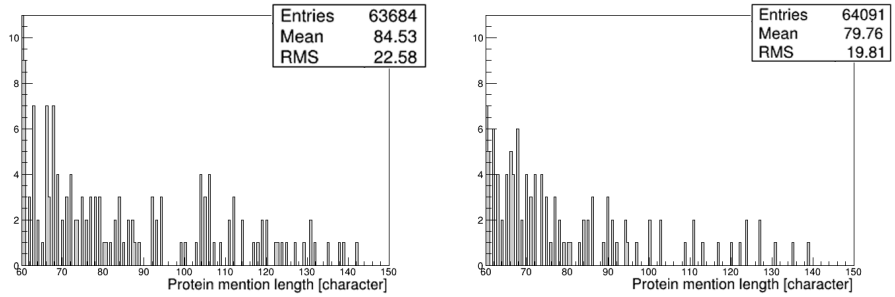


Figure 2.5: Histograms of the distribution of length of protein mentions extracted by BANNER (top) and BannerM(botom)

Using BannerM justifies the position of the protein NER in the pipeline — after the measurement and concentration annotator. Therefore, it gets already annotated text with mentions of measurements. Thus, BannerM can easily remove them before passing the text to the wrapped original BANNER [9] version.

2.6.4 Filtering on the length of Proteins

The first results of protein annotations has shown that when the sentence is not an English sentence — this appends when the extraction of the full-text is not valid — machine-learning-based annotators, especially BANNER, becomes unpredictable. A qualitative analysis leads to the fact that proteins with the longest length tends to be false positives. To find an upper bound for protein name length, the already-annotated mentions of proteins have been extracted from the GENIA corpus [6]. An histogram summarizing protein name length can be found on figure 2.6. As the histogram tail shows, the most long protein mention found in the GENIA corpus has a length of 118 characters. The dataset being large enough to make decision using it, we set our cutoff point a bit higher to allow some variation, i.e. 150 characters. In other words, BannerM implements a feature to filter out proteins estimated mentions whose length are greater than 150 characters.

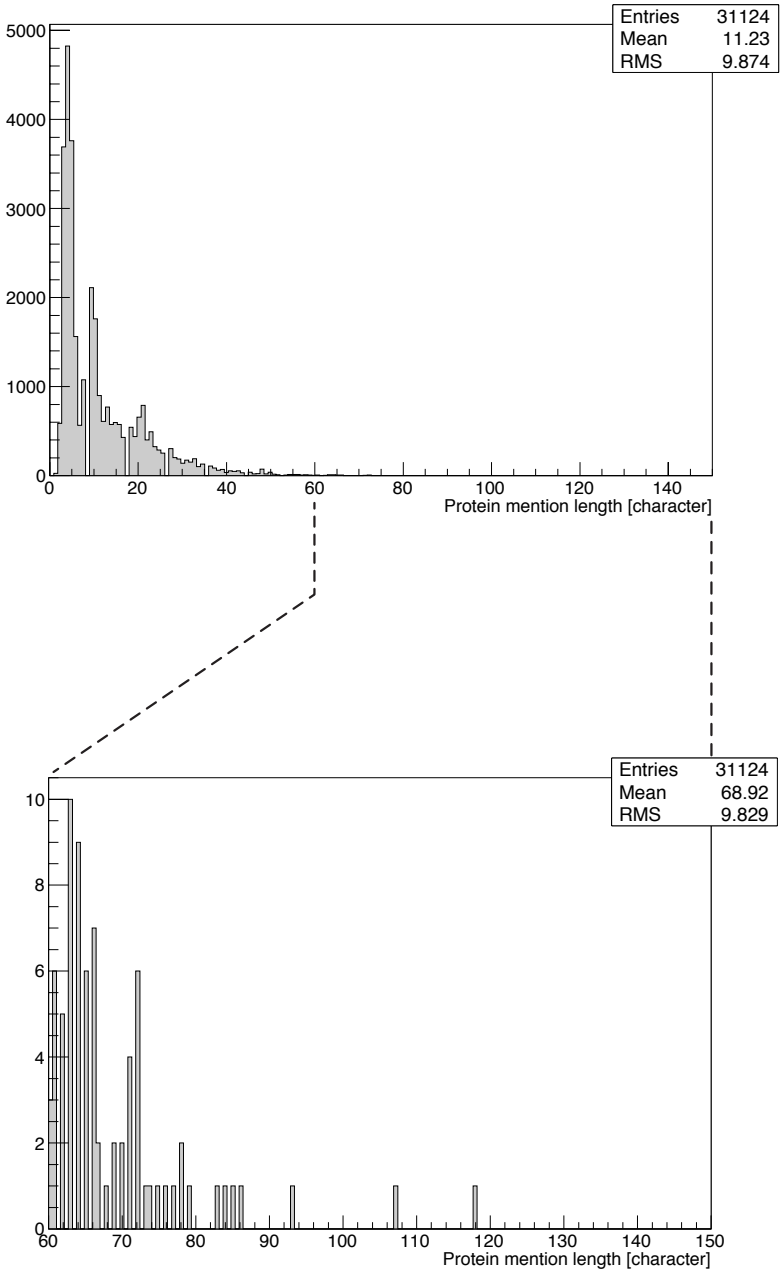


Figure 2.6: Histograms summarizing the length of protein found in the GENIA corpus. Top: overall distribution of the length of protein mentions in the GENIA coups. Bottom: the tail of that distribution.

2.7 Annotating the Brain Regions, Cell types and Subcells

The annotator developed to detect the brain regions is a machine-learning-based annotator (CRF) [3]. It is now a component of the Bluima analysis tools [10]. It has been retrained for the purpose of this project. The annotators and subcells and cell types are list-based annotators and, thus, should produce more primitive results. That is, a big part of the project focused on the triplets {protein, concentration, brain region}.

However, one should also consider cell types and subcells given the high level of precision induced by protein concentration. The concentration of a protein makes more sense in the context of a subcell or a cell type than in the context of a brain region. The lack of targeted relations found in the extraction of the latter form suggests that this intuition is correct.

2.8 The Relation Extraction Stage

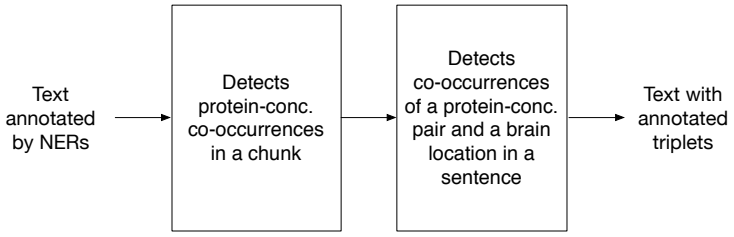


Figure 2.7: The overall model for relation extraction

Given the absence of any reference annotated corpora, the time constraints and the lack of biomedical experience to annotate a corpus ourselves, the approach used to extract relations is quite simple and based on a co-occurrence model. Two named entities co-occurrs in a given enclosing scope (e.g. sentence) if there is at least one occurrence of the considered scope in the text containing occurrences of both named entities. Note that this project is about extracting triplets, not simple binary relation. Since the latter is more easy to conceive as well as to implement, the extraction of these tuples is considered as two subsequent co-occurrences' extractions. In the first stage, co-occurrences between a protein's named entity and a concentration's named entity (sometimes denoted as pc-cooccurrences) are detected. Then, co-occurrences between these pc-cooccurrences and brain region's named entities are then extracted.

To illustrate this approach, consider the following sentence [PMID: 21843466]:

> After a further 3h in the absence (A) or presence (B) of 25 ng/ml VEGF, individual cells started to migrate into the wound.

Then, it is annotated by the NERs (concentration mention in blue, protein mention in red and location in green):

> After a further 3h in the absence (A) or presence (B) of 25 ng/ml VEGF, individual cells started to migrate into the wound.

Since the protein and the concentration appear in the same chunk, the complement of the "or", they are considered as related:

> After a further 3h in the absence (A) or presence (B) of 25 ng/ml VEGF, individual cells started to migrate into the wound.

Since the underlined group, the pc-cooccurrence, co-occurs with a brain location in a sentence, this sentence is extracted.

Choosing the enclosing scope is the trickiest part of this kind of relation extractor. Indeed, the lack of reference annotated material forces to use a qualitative approach. One strategy (that we tried) is to use sentences. However, qualitative analysis reveals the existence of a large number of sentences with high density of concentration named entities. Indeed, biomedical literature usually contains a section that precisely defines an experiment in a step-by-step way. Such recipes are prone to contain a lot of details involving concentrations (the "Materials & Methods" section).

Let $C_{pc}(x)$ be the list of co-occurrences (protein-concentration) in a sentence x and $T_{pc}(x)$ be the list of true positives among $C_{pc}(x)$, being the co-occurrences linking a protein and a concentration which are truly related semantically speaking. Consider the following examples (the text colored in red are proteins while the text colored in blue are concentrations) which bears witness to the weakness of the pc-cooccurrences among sentences.

Sentence s	$\lvert C_{pc}(s)\rvert$	$\lvert T_{pc}(s)\rvert$
The follicular membranes were removed by digestion in Ca2-free OR2 solution (82.5 mM NaCl, 2.5 mM KCl, 1 mM MgCl , 5 mM HEPES, 2 pH 7.6) containing 2 mg/ml collagenase (Sigma). [PMID: 15016809]	5	1
Oocytes were freed from ovarian lobes by gentle mechanical agitation in Ca2-free ND96 solution (96 mM NaCl, 2 mM KCl, 1 mM MgCl2, and 5 mM HEPES, with pH adjusted to 7.5 with NaOH) containing 1.5 mg/ml collagenase (type 1A, Sigma Chemical) for 4560 min. [PMID: 12890647]	5	1
Phosphorylation of ERK-1/2 and Elk-1 was increased by N/OFQ (100 nM) and the PKC activators PDBu (100 nM) and IDB (100 nM) in acutely isolated rat cerebral parietal cortical neurons. [PMID: 20331962]	6	0

This leads us to the conclusion that a sentence is a too wide enclosing scope. Choosing chunks as enclosing scope seems to be a promising way since it fits with all the above examples and since intuition tells us that the probability of co-occurrence in the same chunk of a protein and a concentration which are not in relation should be quite low. Nevertheless, the other side of the coin is that it could decrease the recall. However, the lack of control on the recall inherent in this project leads us to choose the precision increase as a sufficient argument.

Chapter 3

Results

This chapter groups together the results collected during the large-scale analysis on the BBP clusters. It also aims to explain for each extraction how it influences the UIMA chain and, more generally, the decision path along the project.

3.1 Analysis of the first large-scale Dataset

To test the chain, more than one million full-text PDFs (1'339'555) coming from the Pubmed corpora has been processed. The state of the pipeline was as described in chapter 2, but, since the experiment conducted on GENIA corpus described in section 2.6.4 was not finished at the time of the run, the cutoff point for protein name length was arbitrarily set to 200 characters. To get an idea of the spread of some visible common errors over the data, an extract of records, namely tuples of the form

{pubmed id, protein, brain region, concentration, enclosing sentence},

were randomly extracted from the dataset. Note that the complete dataset extracted during this first large-scale extraction consists of 191'996 of such records.

3.1.1 The 300-records Experiment

As mentionned above, the purpose of experiment described here is to understand the spread of some error types over the results. Using a Pareto-like approach, the ultimate goal is to select the problem that will make the more difference once fixed.

The process of this experiment is the following:

1. 300 records were extracted from the dataset,
2. I annotate by hand as belonging or not to some class of error (presented below).

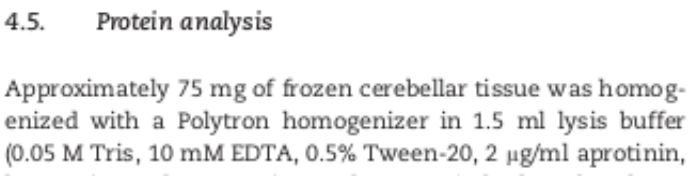
We will discuss each of these classes one after the other in the next subsections.

Bad Sentence Tokenization and Table Error

Some text recognized as sentence by our sentence splitter (see section 2.4) are not so in the English sense of the term. This includes simple mistokenization of sentence. For instance, consider the following sentence [PMID: 17045977]:

> Protein analysisApproximately 75 mg of frozen cerebellar tissue was homogenized with a Polytron homogenizer in 1.5 ml lysis buffer (0.05 M Tris, 10 mM EDTA, 0.5% Tween-20 [...]

where screenshot in PDF is



As one can see, the beginning of the sentence consists in an overlap of the title of the section and the paragraph following it.

Another highly represented class of such false sentences is the result of the PDF reader reading the content of data tables in the PDFs. Here is an example [PMID: 23022093]:

> Template network Anatomical center of the difference-cluster Coordinates Volume Effect (ml) XYZNOI1 (Medial visual network) NOI2 (Lateral visual network)NOI3 (Somatosensory/auditory network)NOI4 (Sensorimotor network)NOI5 (Default mode network)NOI6 (executive/attention/salience network)NOI7 (Left dorsolateral visual stream/working memory)NOI8 (Right dorsolateral visual stream, working memory)Frontal pole 36 91 48 0.51 M ¿ P Hypothalamus 45 59 30 0.49 M ¿ P Cuneus 38 24 50 0.58 M b P Dorsocaudal anterior cingulate 45 66 60 1.32 M b P Cerebellum 55 29 24 1.31 M b P 33 33 23 1.10 M b P Amygdala 33 65 23 1.16 M b P Cuneus 44 26 52 0.85 M b P Precuneus 37 26 59 0.62 M b P Frontal pole 59 86 51 0.47 M b P Supramarginal 74 50 58 0.45 M b P Occipital pole 55 18 32 1.06 M ¿ P Cerebellum 45 42 35 52.67 M b P White matter 53 67 45 3.26 M b P Supramarginal gyrus Right 16 43 51 2.30 M b P Left 75 42 54 0.50 M b P Insula 21 69 36 1.38 M b P Ventricle 42 47 42 0.98 M b P Premotor 38 65 68 0.97 M b P Frontal pole 59 87 48 0.60 M b P Posterior cingulate cortex 45 37 49 8.58 M b P Rostral anterior cingulate 46 90 34 8.52 M b P Frontal pole Right 29 80 59 2.79 M b P Left 58 81 58 2.49 M b P Brainstem 45 60 29 1.94 M b P Fusiform Right 36 26 30 1.58 M b P Left 53 23 28 0.84 M b P Cerebellum 66 27 14 0.97 M b P 54 42 12 0.75 M b P Lateral parietal lobule 65 28 60 0.86 M b P Retrosplenium 39 41 35 0.54 M b P Middle frontal gyrus 71 71 56 0.48 M b P Left caudate

44 63 41 1.01 M b P Dorsal anterior cingulate 45 76 46 0.70 M b P Cerebellum 57 27 17 0.46 M b P Frontal pole, right 30 90 48 0.42 M b P Insula Left 63 71 35 0.41 M b P Right 28 73 29 0.38 M b P Dorsal paracingulate 47 91 44 1.18 M ¿ P Cerebellum 34 23 18 0.74 M ¿ P Frontal pole Left 56 86 53 0.57 M ¿ P Right 24 85 47 1.63 M b P Middle frontal gyrus 32 73 64 1.84 M b P Frontal gyrus Left 65 79 47 16.62 M b P Right 19 78 46 1.98 M b P Left inferior temporal gyrus 72 37 30 2.38 M b P Precuneus 45 26 60 1.46 M b P Superior parietal lobule 68 38 62 0.99 M b P Occipital fusiform 41 22 32 0.73 M b P Left hippocampus 54 55 28 0.67 M b P Retrosplenial cortex 49 42 38 0.41 M b PNOI templates are obtained from independent component analysis as described by Beckmann et al. (2005).

while in original PDF it looks like:

Template network	Anatomical center of the difference-cluster	Coordinates			Volume (ml)	Effect
		X	Y	Z		
NOI1 (Medial visual network)	Frontal pole	36	91	48	0.51	M>P
NOI2 (Lateral visual network)	Hypothalamus	45	59	30	0.49	M>P
	Cuneus	38	24	50	0.58	M<P
NOI3 (Somatosensory/auditory network)	Dorsocaudal anterior cingulate	45	66	60	1.32	M<P
	Cerebellum	55	29	24	1.31	M<P
		33	33	23	1.10	M<P
	Amygdala	33	65	23	1.16	M<P
	Cuneus	44	26	52	0.85	M<P
	Precuneus	37	26	59	0.62	M<P
	Frontal pole	59	86	51	0.47	M<P
	Supramarginal	74	50	58	0.45	M<P
NOI4 (Sensorimotor network)	Occipital pole	55	18	32	1.06	M>P
	Cerebellum	45	42	35	52.67	M<P
	White matter	53	67	45	3.26	M<P
	Supramarginal gyrus					
	Right	16	43	51	2.30	M<P
	Left	75	42	54	0.50	M<P
	Insula	21	69	36	1.38	M<P
	Ventricle	42	47	42	0.98	M<P
	Premotor	38	65	68	0.97	M<P
	Frontal pole	59	87	48	0.60	M<P
NOI5 ("Default mode" network)	Posterior cingulate cortex	45	37	49	8.58	M<P
	Rostral anterior cingulate	46	90	34	8.52	M<P
	Frontal pole					
	Right	29	80	59	2.79	M<P
	Left	58	81	58	2.49	M<P
	Brainstem	45	60	29	1.94	M<P
	Fusiform					
	Right	36	26	30	1.58	M<P
	Left	53	23	28	0.84	M<P
	Cerebellum	66	27	14	0.97	M<P
		54	42	12	0.75	M<P

Note that the above example is a good example of the hardness to treat full-text table extracted from PDF documents. Note that these two classes of error (sentence tokenization problem and table error) are highly correlated ($r = 0.86$). It can be explained by the fact that a table error is a special instance of a sentence tokenization problem. 93% of the cases where a sentence presents a tokenization problem, the sentence appears to be spanning a data table. Nevertheless, a quick fix can be deduced from the fact that sentences spanning table content appear to be quite longer than normal sentence. More precisely, the indicator telling whether a sentence has a length greater than 1000 characters has a 0.768 correlation with the one indicating that a sentence has a table form. Note that the problems of sentence tokenization is a major problem for our extraction. since 71.33% of the sentences suffers from it. That is, filtering out sentences whose length is greater than 1000 charcters appears to be a good workaround.

Bad Chunking

We talk of bad chunking when a chunk contains non related (syntactically speaking) protein and concentration. A current pattern in the syntax of scientific literature is the use of enumeration. The co-occurrences of a protein and a concentration do not escape this rule. Consider the co-occurrences between biomedical entities and concentration in the following sentence [PMID: 22197517]:

> Tissues were sonicated in 1% SDS in TE (pH = 7.4) containing 1 protease inhibitor cocktail (1 mM AEBSF, 0.08 mM aprotinin, 21 mM leupeptin, 36 mM bestatin, 15 mM pepstatin A, and 14 mM E-64).

Note that the number of co-occurrences grows exponentially with the length of the enumeration. Let B be the set of biomedical entity appearing in the above enumeration. That is,

$$B = \{\text{AEBSF, aprotinin, leupeptin, bestatin, pepstatin, E-64}\}$$

and C the set of all the concentration appearing in the example. That is,

$$C = \{\text{1 mM, 0.08 mM, 21 mM, 36 mM, 15 mM, 14 mM}\}.$$

Then, the total number of co-occurrences in the enumeration is $|C \times B|$ when following a naive approach. To solve this misconception, one can argue that, in the model described before, the co-occurrence of a protein and a concentration is considered as being a relation if the protein and the concentration appears in the same syntaxical chunk. However, in practice, the definition of chunk provided by the OpenNLP chunker [2] contains too much irregularities to assume that any enumeration would be separated on the commas.

The problem of bad chunking seems to be present in 21% of our results. After excluding the sentences being badly tokenized from the calculations since it is hard to decide whether a piece of text is properly chunked in those since the concept of chunk only makes sense in a syntactically correct context, the resulting sample contains 65.1% of cases with chunking problem. That is, we decided to add some properties to the definition of chunk, namely

- a item of an coma-separated enumeration is always one chunk, and
- pattern of the form '⟨a protein mention⟩ (⟨a concentration mention⟩)' is always in one chunk.

Then, to avoid the combinatorial expansion of the number of co-occurrences, only the high-confidence co-occurrences are kept, namely the co-occurrences with the smallest distance between their members. Reconsider one of the examples from section 2.6.3 [PMID: 15016809]:

> The follicular membranes were removed by digestion in Ca2-free OR2 solution (82.5 mM NaCl, 2.5 mM KCl, 1 mM MgCl , 5 mM HEPES, 2 pH 7.6) containing 2 mg/ml collagenase (Sigma).

If one consider all the 5 possibles pc-cooccurrences enclosed by the sentence, it gets 4 false positives. However, keeping only the nearest-neighbours co-occurrence which contains "collagenase" means keeping only the co-occurrence between it and the closer concentration in the sentence, i.e. "2 mg/ml". One could argue that this approach can be used among sentences and free us from using chunks. However, notice that a protein and a concentration mentions at both ends of a sentence intuitively tends to be unrelated but could be a nearest-neighbouts co-occurrence provided no other mention of such kinds exists in the sentence. Therefore, we keep our approach to consider pc-cooccurrences among chunk, but, in the case where more than one concentration or protein occurs in a chunk, we use a nearest-neighbours policy.

Bad Protein Mentions

Qualitative analysis of protein estimated mentions reveals two big problems: (1) when BannerM is run on table, it becomes unpredictable as revealed by these examples of extracted protein estimated mentions:

- GR 3 19 F 5 Anisocoria GR 4 16 M 7 GR 5 35 M 4 Anisocoria GR 6 48 F 12 Anisocoria GR 7 28 M 3 GR
- C9 87 M 17 IV
- UI 16 2 79 F 76 3 GD
- LB 86 F W 16 Str
- NrA Normal control C-2 80 M 12 NrA Normal control C-3 57 F 8 NrA Normal control C-4 70 M 7 NrA
- P Middle frontal gyrus 32 73 64 1.84 M b P Frontal gyrus Left 65 79 47 16.62 M b P Right 19 78 46 1.98 M b P

and (2) chemical elements are interpreted as protein mentions such as "NaCL" or "KHPO4".

The first problem should be overcome by filtering out the text coming from table content using the length of sentence as explained in section 3.1.1. To limit the number of some obviously non-protein chemical elements BANNER considers as protein mentions, a list-based filter was built based on the more frequent of such confusions. A more long-term efficient approach would have been to use a chemical NER such as OSCAR4 [5] to differentiate a protein mentions from chemical ones, but this would have increased the runtime too much considering the time constraint of the project.

3.2 Large-scale Dataset after Filtering

Memoization of the outputs of machine-learning-based annotators using BinaryCasWriter and BinaryCasReader of the Bluima toolkit [10] during the produc-

tion of the last large-scale dataset allows the quick production of a new dataset after inserting different filters in the pipeline as described in the last chapter.

3.2.1 Filtering summary

To sum up what have been described above, this new large-scale run is performed with new elements in the pipeline, i.e.

- a chunk adapter as described in last section,
- a list-based filter to limit number of the chemical entities in the collected protein estimated mentions,
- a sentence-length filter which filters out sentences with length higher than 1000 characters, and
- a nearest neighbours co-occurrences annotator which only keeps co-occurrences enclosing the closest entities — in case of race when two possible co-occurrences enclose entities at the same distance of each other, the first detected co-occurrence is kept.

3.2.2 Reconducting the 300-records Experiment

Reconducting the 300-records experiment, the experiment on error distribution conducted on the last large-scale extraction (see 3.1.1), leads to promising result regarding the syntactical quality of the data. Indeed, the filtering applied during this extraction remarkably reduced the impact of the sentence mistokenization problem, namely only 8% of the sentences appear to be mistokenized in this dataset and also 8% are in a table form. However, chunking problems are still present in 37% of the records but not in the form of combinatorial expansion of co-occurrences. Some doubts can be emitted regarding the efficacity of the chunk adapter. Note that some little bugs have been diagnosed and corrected in the chunk adapter since; thus, the percentage of chunk error must be reconsidered. No new data is available at the moment. Despite the bug, qualitative analysis of the results seems to indicate that the quality of non-buggy chunks are better than in the last run. Therefore, the nearest-neighbour co-occurrences seems to produce good results.

More generally, about 60% appears to be error-free from the point of this experiment. This is a slightly good result considered relatively to the first dataset which has only about 8% of its results presenting none of the considered errors.

3.2.3 The apparent Lack of True Positives

In addition to make the data cleaner, one of the aim of this second large-scale data collection was to diminish the persistent hardness to find instances of the

Pubmed ID	Sentence
9483526	The rat hippocampus was homogenized (5% w/v) in cold PBS containing 1 mg/ml of leupeptin, 100 mM phenylmethylsulphonyl fluoride, and 1 unit/ml aprotinin, and centrifuged at 10,000 g for 30 min.
20829391	Rat cortical neurons were serum starved for 3 d and then stimulated with 0.5 nM insulin.
21784010	Hippocampus was homogenated by sonication in 250 l of lysis buffer (PBS, 1% Nonidet P-40, 0.1% SDS, 0.5% sodium deoxycholate, 1 mM phenylmethylsulphonyl fluoride, 10 g/ml aprotinin, 1 g/ml leupeptin, 2 g/ml sodium orthovanadate) and centrifuged at 13,000 rpm for 15 min 4 °C.
15896913	Dissected hippocampus was homogenized in lysis buffer (18 l/mg tissue) containing 137 mM NaCl, 20 mM Tris?HCl (pH 8.0), 1% NP40, 10% glycerol, 1 mM PMSF, leupeptin (1 g/ml), sodium vanadate (0.5 mM), AEBSF (100 mg/ml).

Table 3.1: Some example of the "methodological pollution"

targeted relations[1] among results.

The apparent majority of the results, in spite of their clearly better quality than in the first large-scale run, comes from the *Materials and methods* section or are, at least, methodological data which is not the main concern of BBP researchers. That is, they describe experimental process which is not the most wanted kind of relations in the perspective of this project. Table 3.1 shows some examples of such sentences. The rarity of true positives could also indicate that our focus on brain region as the third entity of our co-occurrences is wrong regarding to the desired information. That is, some focus on the cell types and subcells data is important at this stage. However, a qualitative inspection of the records for the cellular scale reveals that the "methodological polution" of our data is still there. For instance, this extraction counting 286100 records, filtering out the sentences containing lemmas that are typically methodological, namely "incubat", "wash", "culture" or "buffer", results in a set of 181122 records. That is, 104978 have been filtered out by such an approach. Note that this kind of filter can be damageable without a corpus to evaluate them. Therefore, we tried to apply a more conservative approach. First, the section title are annotated using a regexp approach. Then, a confidence flag is added to each record having a higher value if it corresponds to a sentence in a section like *Results* or *Conclusions* and a lower one if it corresponds to a sentence in the methodological section(s). If it is not possible to determine the section enclosing a sentence, then a neutral confidence flag is set. The first qualitative look at the resulting materials (after section-confidence annotation) seems to show that

[1]Reminder: the relation describing how a given protein is concentrated in a given location (all scale included: cellular, subcellular, regional). See section 1.2 for more details.

- indeed, the resulting data is more result-oriented and, that is, more useful to researchers,
- it seems that a considerable number of methodological mentions still appears in the sections presenting results in the papers and, therefore, does not solve the problem, and
- the targeted relation still appears to be hidden.

3.3 Results overview

The summary of our final results are shown in table 3.2. It intends to give an overview of the results obtained for the three considered scales (brain region, cell and subcell). The first row shows the number of records extracted for each scale. The next rows display the size of some interesting subsets of our final dataset. Finally, the two last rows respectively shows the percentage of the data extracted from a section presenting results (*Results*, *Discussions*, *Conclusions*) and from the *Materials and Methods* section. Note that the fact this two rows do not sum up to 100% is due to the cases where it is not possible to establish the section from which they were extracted.

As a conclusion for this chapter, it is interesting to see that, even if the methodological pollution is effective in our data qualitatively speaking, our extractions comes more from the section about results than from the methodological sections in general. This seems to confirm the qualitative conclusion of last section concerning the presence of methodological content in the sections about results and, therefore, challenges the chosen approach to classify sentences as being methodological or not. Finally, a web application has been developed to make the result avaible and is described in next section.

Scale	**Brain region**	**Cell**	**Subcell**
# records	41'938	286'100	2'255
# distinct triplets	37'212	166'371	1'844
# distinct proteins	18'431	75'101	1'295
# distinct regions/cells/subcells	7'808	488	57
% from RESULTS	48.60%	48.50%	57.80%
% from MM	28.50%	32.40%	17.90%

Table 3.2: Final result summary

Chapter 4

Visualizing the results

This chapter summarizes the visualization tool developed to present the results of this work to BBP researchers. A server which makes the collected data avaible in JSON form was developed in Python using the Tornado framework[1] and follows a REST-full philosophy. In addition, three views were developed in HTML/CSS/JavaScript using Bootstrap framework[2] to visualize the collected data. The next sections intend to give an overview of each of those.

4.1 The Top Table View

The Top Table View displayed in figure 4.1 intends to allow the researchers to quickly access the details concerning the most frequent triplets through a color-coded table. It confronts the most frequent proteins and the most frequent brain regions (it can easily be adapted for cells or subcells too) in a table and displays in the cells how many relations exists between them according to our dataset. The cells are color-coded following the principles: the darker, the more. Once set up on a Tornado server, the Table View is accessible by GETting /static/table.html.

4.2 The Search View

As one can see on figure 4.2, the Search View intends to allows a more specialized access to the data by letting the user enters keywords for the protein or brain region (exact match or SQL regexp) according to its interest. The search results are then displayed in a list mode and are sorted according to the amount of data avaible for each of them. To illustrate the different keyword matching: searching for protein "insulin" in exact match will returns only relations involving insulin while searching for "%insulin" in SQL regexp (LIKE syntax[3]) returns relations

[1] See `http://www.tornadoweb.org/en/stable/`.
[2] See `http://getbootstrap.com/`.
[3] See `http://dev.mysql.com/doc/refman/5.0/en/pattern-matching.html` for references.

Summary

	cortical	Ca2	Spinal cord	pituitary	hippocampus	striatal	cerebrospinal fluid	cerebellar	striatum	cerebellum	cortex	CT	retinal	cerebral cortex	Hypothalamic
trypsin	14	19	13	4	15	3	4	4	3	12	14	3	4	3	0
insulin	24	8	2	43	2	1	4	8	0	0	2	1	9	2	8
leupeptin	11	0	24	3	35	3	1	1	10	11	12	5	1	8	1
bovine serum albumin	9	7	9	6	8	5	14	3	2	13	5	3	4	0	2
Fig	17	5	6	1	1	5	2	1	3	4	3	5	1	6	3
Aprotinin	8	1	6	3	31	8	4	0	6	5	7	2	3	9	2
EGF	14	2	4	58	2	2	0	2	5	4	1	7	10	1	2
PMSF	6	0	18	2	23	4	0	1	5	7	7	4	1	5	4
NGF	17	1	13	2	2	1	0	0	1	2	6	2	0	0	1
leptin	1	1	0	10	3	0	1	0	0	1	1	0	0	0	19
Epidermal Growth Factor	5	0	0	117	0	0	0	0	1	0	1	0	1	0	0
ANG II	1	4	1	2	0	0	1	3	1	12	1	1	1	0	6
IgG	3	3	7	0	1	0	29	1	0	3	1	0	0	0	0
TNF	15	2	1	0	0	0	1	0	0	0	0	2	0	0	2
NMDA receptor	7	1	7	0	7	3	3	6	0	2	2	1	0	1	0
VEGF	5	4	17	0	0	0	1	0	0	0	1	1	16	3	0
bFGF	10	0	5	5	3	3	0	0	1	0	2	0	4	1	0
NaF	2	1	9	1	14	5	0	3	4	5	4	1	0	3	2

Figure 4.1: A screen shot of the Top Table View

involving “insulin”, “bovine insulin” and “plasma insulin”. The Search View is accessible by GETting /static/search.html.

4.3 The Detailed View

The detailed view can be accessed either from the Top Table View or the Search View. It displays the collected data in plots for normalized concentrations and in tables for unnormalized information as displayed on figure 4.3. It also provides a quick way to access the original document by providing a link to its PubMed citation after the user has clicked on a data point as shown in figure . The plots are drawn using the AmCharts JavaScript library[4] and displays, for a given concentration on the x axis, the number of such concentration for the relation in question on the y axis. For instance, the Detailed View for the leupeptin protein and hippocampus brain region displays the data point $(1, 3)$ in the plot for kg/m^3-normalized units meaning that our databases contains three relations linking 1 kg/m^3 of leupeptin and hippocampus.

[4]See `http://www.amcharts.com/`.

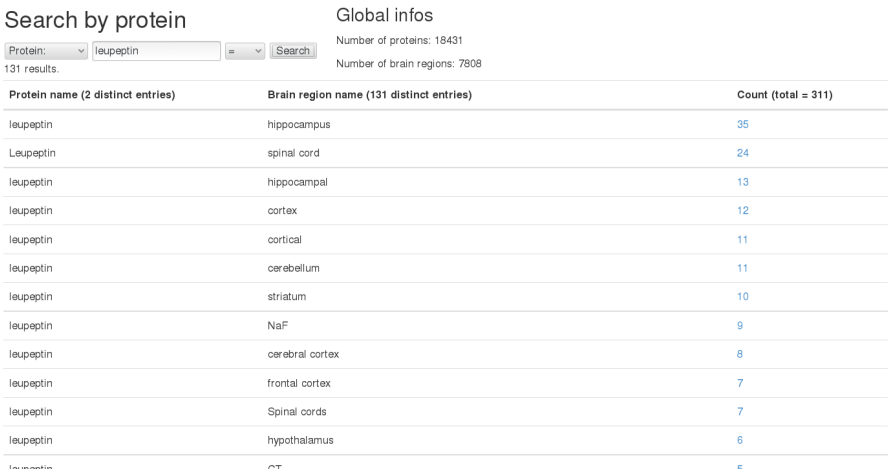


Figure 4.2: A screen shot of the Search View

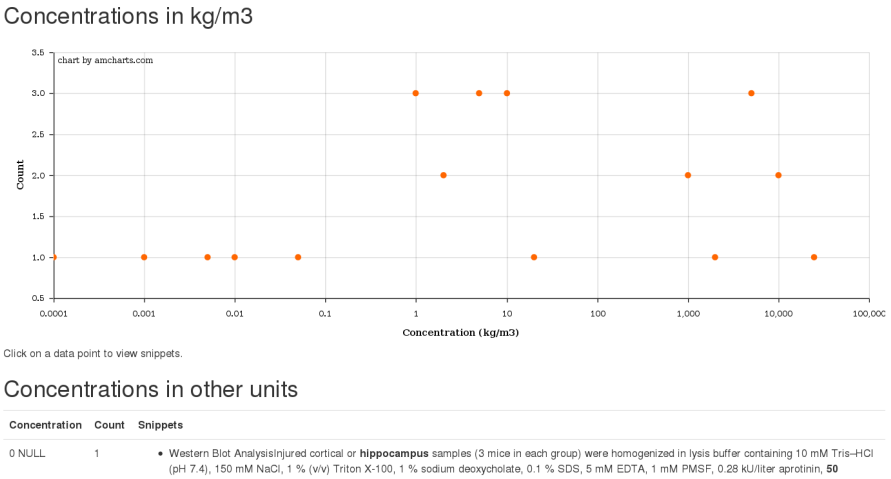
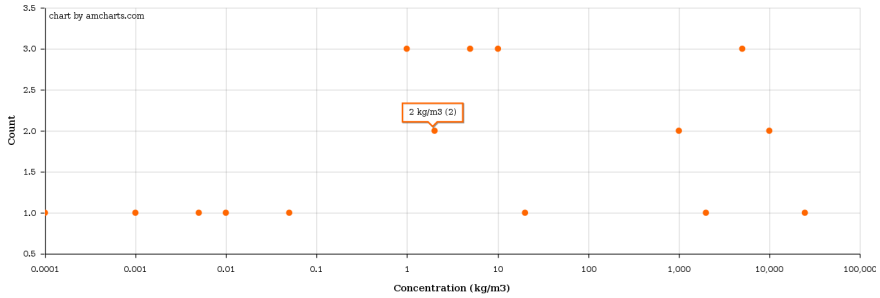


Figure 4.3: A screen shot of the Detailed View



- Fifty milligrams of the **hippocampus** were homogenized with 0.5 ml of ice-cold buffer containing 50 mM 3-[N-morpholino]propane–sulfonic acid–HCl (Tris–HCl, pH 7.4), 1 mM sodium orthovanadate, 150 mM NaCl, 1% Triton X, 5 mM EDTA, 0.5% sodium deoxycholate, 0.1% sodium dodecyl sulphate (SDS), **2 mg/ml leupeptin**, 1 mM phenylmethylsulfonyl fluoride, and 1 mM MgCl2. (Pubmed)
- Microdiscs (1 mm diameter) were potentially lead to phosphorylation-dependent activapunched out from the different structures and immetion of two major cAMP-regulated proteins involved diately homogenized in ice-cold sample preparation in the regulation of protein synthesis—the transcripbuffer (50 mM Tris–HCl pH 7.4, 5 mM EDTA, 1 mM tion factor CREB and DARPP-32, and D1 receptors EGTA, 0.1% Triton X-100, 1 mM PMSF, 200 nM calyare necessary for **hippocampus** to prefrontal cortex culin A, **2 mg/ml leupeptin**, 5 mg/ml aprotinin, 1 mM LTP—we chose to investigate CREB and DARPP-32 microcystin, and 5 mM NaF), and centrifuged at phosphorylation state following the induction of pre- 14,000g for 5 min at 4°C. (Pubmed)

Concentrations in other units

Figure 4.4: A screen shot of the Detailed View once clicking on a data point.

Chapter 5

Conclusion

During this work, we have seen that

- the extraction of relations linking a protein, its concentration and a brain location can be performed using simple heuristics associated with powerful existing NERs despite the very specific nature of this material
- but attempting to select the targeted relation kind (how concentrated is a given protein in a given location) among the results seems to remain hard.

The heuristics used in this project seems not to be enough efficient to overcome the lack of reference annotated corpus. But only future researchers' feedback will be able to establish the usability of the current dataset. To overcome the difficulty to classify the strength of the collected relations, this project could constitute a basis to construct a reference annotated corpus through, for instance, an active learning tool for making the classification of the relations more precise. In this perspective, a visualization displaying the results has been developed and will be made available to researchers for feed-back and, then, for evaluation.

The question of where the targeted relations are is still open at this stage. The rarity in the results of our extractions of the very specific relations characterizing the concentration of a protein in a location of the brain (all scale included: regional, cellular, subcellular, ...) challenges an assumption implicitly supported along this project that such relations can be extracted from, or even found in the body of scientific publications. This guess has been made not merely because of the absence of corpus to evaluate such assumption but because of the lack of tools to properly treat other form of data presentations in paper in PDF format including charts and tables[1], development of whose was outside the scope of this project. Therefore, future work attempting to extract highly specialized data could focus on the treatment of such data formats. One

[1] the example given in section 3.1.1 bears witness of the hardness to use full-text table using the current version of the tools

could also ask, more generally, if the very targeted relations really exists in scientific literature. Another future work could be to use the pseudo corpus we produce (by annotating sentence as belonging to the methodological section or to the section presenting results) to bootstrap a reference corpus regarding the classification of sentence as being semantically methodological or not.

Finally, one of the successfully reached goal of this project is its contribution to the improvement of the Bluima toolkit [10] especially regarding the addition of tools dedicated to relation extraction (in particular improved co-occurrence extractors), some of which are currently already in use in other project(s) supported by the BBP.

Acknowledgment

This project was realized in the context of the Blue Brain Project.

I would like to express the deepest appreciation for my supervisors: Jean-Cédric Chappelier for sharing its scientific rigour, its experienced advices and support as well as its availability for constant follow-up, and Renaud Richardet for its advices, its impressive availability, its participation to all the steps of this project and its technical support.

I also place on record my sincere gratitude to Martin Telefont and Daniel Keller for their participation and feedback.

Bibliography

[1] Oliveira J.L. Campos D., Matos S. Gimli: open source and high-performance biomedical name recognition. *BMC Bioinformatics 14*, 54, Feb 2013.

[2] Buyko E. JULIE Lab UIMA Wrapper for OpenNLP Chunker. Avaible at `http://www.julielab.de/coling_multimedia/de/downloads/NLP+Tool+Suite/Documentation/UIMA+Analysis+Engine/jules_opennlp_chunk_ae.pdf`.

[3] Xu L. French L., Lane S. and Pavlidis P. Automated recognition of brain region mentions in neuroscience literature. *Front Neuroinformatics*, 3, Sept 2009.

[4] Portmann J. BioNLP for Extraction of Protein Concentration in Cells out of Scientific Literature. Avaible at `https://bbpteam.epfl.ch/project/spaces/pages/viewpage.action?pageId=5277383`., 2012.

[5] et al. Jessop, D. Oscar4: a flexible architecture for chemical text-mining. *Journal of Cheminformatics*, Oct 2011.

[6] Tateisi Y. Kim J.D., Ohta T. and Tsujii J. GENIA corpus–a semantically anno- tated corpus for bio-textmining. *Bioinformatics 19*, pages i180–i182, Jul 2003.

[7] Baumgartner Jr. W. A. Liu H., Christiansen T. and Verspoor K. Biolemmatizer: a lemmatization tool for morphological processing of biomedical text, 2012.

[8] Rollier O. Content Extraction from PDF Scientific Articles. Avaible at `http://psb.stanford.edu/psb-online/proceedings/psb08/leaman.pdf`., 2013.

[9] Leaman R. and Gonzalez G. Banner: An executable survey of advances in biomedical named entity recognition. *Pacific Symposium on Biocomputing 13*, pages 652–663, 2008.

[10] Chappelier J.-C. Richardet R. and Telefont M. Bluima: a UIMA-based NLP Toolkit for Neuroscience. Avaible at `http://ceur-ws.org/Vol-1038/paper_7.pdf`.