Agile In-Litero Experiments; How can semi-automated information extraction from neuroscientific literature help neuroscience model building?

THÈSE N^{O} 6809 (2016)

PRÉSENTÉE LE 10 FÉVRIER 2016

À LA FACULTÉ DES SCIENCES DE LA VIE
PROJET BLUEBRAIN
PROGRAMME DOCTORAL EN NEUROSCIENCES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Renaud Luc RICHARDET

acceptée sur proposition du jury:

Prof. W. Gerstner, président du jury
Prof. H. Markram, Dr J.-C. Chappelier, directeurs de thèse
Prof. S. Ananiadou, rapporteuse
Prof. M. Martone, rapporteuse
Prof. J.-Ph. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

Agile In-Litero Experiments

How can agile information extraction from neuroscientific literature help neuroscience model building?

Renaud Luc Richardet

September 14th, 2015

Renaud Luc Richardet
Agile In-Litero Experiments
PhD Thesis, September 14th, 2015
Reviewers: Prof. Sophia Ananiadou and Prof. Maryann E. Martone
Supervisors: Prof. Henry Markram and Dr. Jean-Cédric Chappelier

Ecole Polytechnique Fédérale de Lausanne
Blue Brain Project (BBP)
Brain Mind Institute
School of Life Sciences
CH-1015 Lausanne

Abstract

In neuroscience, as in many other scientific domains, the primary form of knowledge dissemination is through published articles in peer-reviewed journals. One challenge for modern neuroinformatics is to design methods to make the knowledge from the tremendous backlog of publications accessible for search, analysis and its integration into computational models.

in litero

In this thesis, we introduce novel natural language processing (NLP) models and systems to mine the neuroscientific literature. In addition to in vivo, in vitro or in silico experiments, we coin the NLP methods developed in this thesis as *in litero* experiments, aiming at analyzing and making accessible the extended body of neuroscientific literature. In particular, we focus on two important neuroscientific entities: brain regions and neural cells.

braiNER:
brain region
connectivity

An integrated NLP model is designed to automatically extract *braiNER: brain region connectivity* statements from very large corpora. This system is applied to a large corpus of 25M PubMed abstracts and 600K full-text articles. Central to this system is the creation of a searchable *database* of brain region connectivity statements, allowing neuroscientists to gain an overview of all brain regions connected to a given region of interest. More importantly, the database enables researcher to provide feedback on connectivity results and links back to the original article sentence to provide the relevant context. The database is evaluated by neuroanatomists on real connectomics tasks (targets of Nucleus Accumbens) and results in significant effort reduction in comparison to previous manual methods (from 1 week to 2h).

neuroNER:
identify
neurons

Subsequently, we introduce *neuroNER* to identify, normalize and compare instances of *neurons* in the scientific literature. Our method relies on identifying and analyzing each of the domain features used to annotate a specific neuron mention, like the morphological term “basket” or brain region “hippocampus”. We apply our method to the same corpus of 25M PubMed abstracts and 600K full-text articles and find over 500K unique neuron type mentions. To demonstrate the utility of our approach, we also apply our method towards cross-comparing the NeuroLex and Human Brain Project (HBP) cell type ontologies. By decoupling a neuron mention’s identity into its specific compositional features, our method can successfully identify specific neuron types even if they are not explicitly listed within a predefined neuron type lexicon, thus greatly facilitating cross-laboratory studies.

bluima:
large-scale
NLP

In order to build such large databases, several tools and infrastructures were developed: a robust pipeline to preprocess full-text PDF articles, as well as *bluima*, an NLP processing pipeline specialized on neuroscience to perform text-mining at PubMed scale.

During the development of those two NLP systems, we acknowledged the need for agile large-scale NLP approaches to rapidly develop custom text mining solutions. This led to the formalization of the *agile text mining* methodology to improve the communication and collaboration between subject matter experts and text miners. Agile text mining is characterized by short development cycles, frequent tasks redefinition and continuous performance monitoring through integration tests. To support our approach, we developed , an NLP framework designed for the development of agile text mining applications.

agile text-mining

Sherlok

Keywords: natural language processing, neuroinformatics, agile data science, information extraction, big data.

Résumé

En neurosciences, comme dans de nombreux autres domaines scientifiques, la principale forme de diffusion des connaissances se fait à travers des articles publiés dans des revues scientifiques. Un grand défi pour la neuroinformatique est de concevoir des méthodes rendant accessible ce large recueil de connaissances, ceci afin de permettre la recherche, l'analyse et l'intégration de ces informations dans des modèles neuroinformatiques.

Dans le cadre de cette thèse, nous introduisons des modèles et systèmes de traitement automatique du langage naturel (TALN) afin d'exploiter les données non-structurées de la littérature neuroscientifique. Tout comme les méthode in vivo, in vitro ou in silico, nous concevons les méthodes de TANL développés dans cette thèse comme expériences *in litero*, visant à analyser et à rendre accessible le vaste corpus de littérature neuroscientifique. En particulier, nous nous concentrons sur deux entités neuroscientifiques importantes: les régions cérébrales et les cellules neuronales.

in litero

Un modèle intégré de TALN est conçu pour extraire automatiquement et à très large échelle des phrases soutenant une connexion entre deux régions du cerveau. Ce système est appliqué à un vaste corpus de 25 millions de résumés issus de PubMed et de 600'000 articles neuroscientifiques intégral. Au cœur de ce système se trouve la création d'une base de données indexée, contenant des connexions entre des régions du cerveau et permettant aux neuroscientifiques d'obtenir un aperçu de toutes les régions connectées à une région particulière. La base de données est évaluée par des neuroanatomistes sur trois tâches (projections afférentes et efférentes au noyau accumbens), résultant en un gain de temps significatif par rapport aux recherches manuelles (temps réduit de 1 semaine à 2h).

connexions entre régions du cerveau

identifications de neurones

Par la suite, nous introduisons un second système, *neuroNER*, pour identifier, normaliser et comparer des instances de neurones dans la littérature neuroscientifique. Notre méthode repose sur la décomposition, l'identification et l'analyse de chacune des propriétés utilisées pour caractériser une mention de neurone, comme par exemple le terme morphologique "cellule pyramidale" ou la région cérébrale "hippocampe". Nous appliquons notre méthode au même corpus et trouvons plus de 500'000 types de neurone différents. Pour démontrer l'utilité de notre approche, nous effectuons une analyse comparative entre NeuroLex et l'ontologie cellulaire du Human Brain Project (HBP). En découplant l'identité d'une mention de neurone dans ses fonctions de composition spécifiques, notre méthode réussit à identifier les types spécifiques de neurones, même si ceux-ci ne figurent pas explicitement dans un lexique, ce qui facilite grandement les comparaisons inter-laboratoires.

TANL à large échelle

Afin de construire ces grandes bases de données, plusieurs infrastructures ont été développées: un outil pour prétraiter les articles intégraux en format PDF, ainsi que *bluima*, une plateforme de TALN spécialisée sur l'analyse des textes neuroscientifiques permettant d'effectuer des fouilles de textes à l'échelle de PubMed.

méthodologie agile de fouille de textes

Lors de l'élaboration de ces deux systèmes TALN (régions du cerveau et neurones), nous avons reconnu la nécessité de proposer de nouvelles approches méthodologiques afin de développer rapidement des solutions personnalisées de fouille de textes. Cela a conduit à la formalisation de la *méthodologie agile de fouille de textes* (AFT, agile text mining) visant à améliorer la communication et la collaboration entre les experts du domaine et les experts en fouille de textes. La méthodologie AFT est caractérisée par des cycles de développement courts, une fréquente réadaptation des objectifs, et le monitoring continu de la performance grâce à des tests d'intégration. Pour soutenir notre approche, nous avons développé *Sherlok*, une plateforme TALN conçue pour le développement d'applications d'AFT.

Mots-clés: traitement automatique du language naturel, neuroinformatique,
agile data science, fouille de texte, big data.

Acknowledgments

> *If I have seen further it is by standing on the shoulders of Giants.*
>
> — **Isaac Newton**
> 1676

I would like to wholeheartedly thank my thesis co-director Jean-Cédric Chappelier for infusing me with his passion for natural language processing, for his honesty & straightness and for the numerous hours he spent coaching me. I am also deeply indebted to my thesis director, Henry Markram, for betting on me and giving me the unique opportunity to join and contribute to the Blue Brain Project. It has been a tremendous experience! A warm thanks to my supervisors at the Blue Brain Project: Catherine Zwahlen for her leadership and availability; Sean Hill for his great feedback that always helped me to move the ball further, and Martin Telefont for introducing me to the field of neuroscience and helping me navigating through its complexities.

Thanks to the greater Blue Brain Project members: Carlos Aguado, Dace Stiebrina, Daniel Keller, Eilif Muller, Emily Clark, Daphne Rondelli, Fabian Delalondre, Felix Schürmann, Iurii Katkov, Jean-Denis Courcol, Jeff Muller, Julian Shillcock, Guy Kahou, Katrien Van Look, Marc-Oliver Gewaltig, Martin Ouellet, Michael Reimann, Mohameth Sy, Yihwa Kim, Rafael Nogueira, Ranjan Rajnish (LNMC), Samuel Kerrien, Srikanth Ramaswamy, Tsolmongerel Papilloud, Vincent Delattre (LNMC), Werner Van Geit and the whole BBP DevOps team (for bearing with me even as I continuousely flooded the cluster). Many thanks as well to my co-authors Xavier Vasquez and Laura Cif. And a very special thanks to Shreejoy Tripathy from the University of British Columbia, my partner in crime for the neuroNER project & my agile text mining alter ego. I also would like to thank the EPFL student that I had the privilege to supervise during my thesis: Joëlle Portmann, Marc Zimmermann, Orianne Rollier, Luca La Spada, Samuel Kimoto, Philémon Favrod, Erick Cobos Tandazo and Marco Antognini.

I'm very grateful to my wonderful jury committee: Maryann Martone, Sophia Ananiadou and Wulfram Gerstner. Thanks as well to Jean-Philippe Thiran, my mentor at EPFL, for being sharp and keeping me on track and in good spirits.

More broadly, I want to thank the whole community of researchers for sharing their research, and the open-source community, in particular the Apache UIMA community.

At last, I want to wholeheartedly thank my family for their support. In particular Elodie, my wife, for believing in me and in us ♡

Genève, le 14 septembre 2015

Contents

1 Introduction

Accessing the vast amounts of data and knowledge embedded in the previous decades of neuroscience publications is essential for modern neuroinformatics. Making these data and knowledge accessible can help scientists maintain a state-of-the-field perspective and improve efficiency of the neuroscientific process by reducing repeated experiments and identifying priorities for new experiments. In order to build models of neural circuitry reflecting the current knowledge, data resulting from many years of prior research must be integrated in the model building process.

In this introductory chapter, we lay the context of this thesis by introducing neuroinformatics (Section 1.1.1) and natural language processing (Section 1.1.2). We continue with an introductory story to lay the context of this thesis (Section 1.2). Then, two research questions are formulated: large scale natural language processing for neuroscience and agile data mining (Section 1.3). A reader's guide to this thesis concludes this chapter (Section 1.4).

1.1 Background

Since the seminal discoveries of Ramón y Cajal in the early twentieth century, modern neuroscience has evolved into a myriad of subdomains, integrating theories and methods from other fields like genetics, microbiology, computer science or physics. This evolution, together with the challenges and importance of brain diseases spawned an unprecedented amount of research, resulting in a vast corpus of scientific knowledge. That knowledge has been mainly published and disseminated through natural written language in scientific articles, so that as of today, a query on Google Scholar for "neuroscience" yields over 2 million results. This exponential flux of information is far too large for individual researchers to ingest. There is thus a vital need to develop tools and methods to stay on top of that growing flow of information.

Searching scientific articles is often performed manually by searching[1], filtering, and manually curating scientific articles. This approach yields high quality information, but is very time-consuming, lacks scalability and might miss relevant articles (because of the lack of semantic information, e.g. synonyms[2] and taxonomies). There is thus a need for systems to perform scalable and precise extraction of neuroscientific information for whole-brain modeling.

This doctoral thesis lays at the boundaries between NLP and neuroinformatics and has been dedicated to creating useful and living links between these two disciplines. The

[1] E.g. using the search engine at Pubmed (http://www.ncbi.nlm.nih.gov), or using Google Scholar (http://scholar.google.com).

[2] For simplicity, `synonyms` is used as a synonym for `surface forms`, and `entities` for `synset`.

objective was neither to develop novel machine learning algorithms nor to discover groundbreaking neuroscience principles. It was instead to push the state of the art in developing and applying NLP models and methodologies onto large corpora of neuroscientific reports.

In addition to in vivo, in vitro or in silico [Mar06] experiments, we coin the NLP methods developed in this thesis as *in litero* experiments, aiming at analyzing and making accessible the extended body of neuroscientific literature. Also, this work is striving to be interoperable with other international efforts.

in litero

1.1.1 Neuroinformatics

Neuroinformatics is a multidisciplinary field combining neuroscience and computer science. Its objective is to develop computational tools to further our understanding of the brain and to structure the large amount of information that neuroscience generates.

One central area of neuroinformatics is concerned with simulating the brain at various granularities and various scales. Such brain simulations are based on a detailed modelization of neurons and synapses, subsequently integrated into models of microcircuits, brain networks and eventually into whole brain systems [D'A+13]. Unlike a top-down theoretical model, a realistic brain simulation is a bottom-up approach based on solid biophysical principles and experimental biological constraints. These constraints are expressed in the form of model parameters. So, simulating the brain requires assigning numerical values to a colossal number of model parameters.

brain simulations

Another central area of neuroinformatics is dedicated to the *integration* of all available neuroscientific data. The launch of the Human Brain Project (HBP) and the growing use of high-throughput methodologies is expected to further accelerate the pace of neuroscientific data production and thus exacerbates the need for neuroinformatics' based integration efforts.

data integration

Schematically, parameters for a brain model can be acquired or integrated from different *sources*. At the simplest, it can be produced in *internal experiments*, e.g. a patch-clamping experiment in one's own laboratory. In such case, one has total control over experiment settings. However, considerably more data is required to construct a brain model that what is possible to create in a single laboratory. Hence the need to integrate experimental results from external laboratories.

internal experiments

There are international efforts to organize and publish neuroscientific data in structured databases and repositories in order to be able to integrate it in brain simulations. For example, the International Neuroinformatics Coordinating Facility (INCF)[3] develops and maintains database and computational infrastructure for neuroscientists. Alternatively, the Neuroscience Information Framework (NIF, [Aki+11; LM13])[4] is a dynamic inventory of web-based neuroscience resources, annotated and indexed with a unified system of biomedical terminology.

organize and integrate external data

One additional source of information is through *manually curated knowledge bases*. Several

manually curated knowledge bases

[3] http://www.incf.org/

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

such initiatives have been created by teams of domain experts manually curating the scientific literature (e.g. BAMS or CoCoMac, see Brain Atlases in Section 2.1). Although these initiatives create extremely valuable information, their creation and maintenance is very time-intensive.

Our idea is to create semi-automated systems that, whenever structured data is not available, will mine the vast amount of unstructured textual data contained in the scientific literature.

1.1.2 Natural language processing (NLP)[5]

Natural language processing (NLP) is a sub-discipline of computer science aiming at developing models and algorithms that allow computers to process and understand human languages. The need for natural language arises by the important fact that natural languages, unlike formal languages, are highly ambiguous, with a lot of undeclared information (implicit shared knowledge). We humans are often not consciously aware of the complexity and ambiguity inherent to natural language. Still, computational models often fail at understanding mildly complex sentence constructs, and great engineering effort has to be deployed to tackle NLP tasks that a 5 years old child would solve at ease. Nonetheless, notable achievements in NLP include machine translation[6], question answering systems [Fer+10], and email spam filtering [And+00].

Whereas early NLP systems relied on a large set of handwritten rules and attempted to formalize natural language (e.g. [Mul+04]), the introduction of statistical machine learning-based approach in the 1980's paved the way to corpus linguistics and to more flexible approaches (e.g. [Cam+12]). Most recent NLP systems are hybrid, relying both on automated machine learning and resources hand-developed by human experts (e.g. [Ric+15]). It is also worth mentioning the trend towards unsupervised methods (e.g. [Mik+13]) to leverage regularities in language. These promising methods are applied on large-scale corpora and can successfully improve NLP systems by providing semantic embeddings [Kae+14].

NLP models have been developed to deal with the different linguistic levels, among which the *morpho-lexical* level (how do languages form words?), *syntactic* level (how do languages form sentences?) and *semantic* level (how do languages convey meaning in sentences?). At the morpho-lexical level, NLP models perform tasks such as sentence and word tokenization, or spelling error correction. At the syntactic level, NLP models have been developed to assign *part-of-speech* tags to each word (e.g., adjective, verb, determinant) in a given sentence as well as inferring the syntactic structure (*grammar*) of a sentence. Syntax is also concerned with the various relationships between words (e.g., subject, object and other modifiers). At the semantic level, NLP models deal with labeling entities like person or proteins, clustering tokens that refer to the same entity (coreference resolution), relation and knowledge extraction (e.g. is-a relationships or protein-protein interaction). There are strong interdependences between the different linguistic levels (recognition is conditioned by structuring, structuring guided by the meaning and the context). There are many other important NLP tasks like speech processing and segmentation, sentiment analysis,

morphology

syntax

semantic

[5] For a thorough introduction to NLP, see [MS99] or [Raj07].

[6] http://translate.google.com/

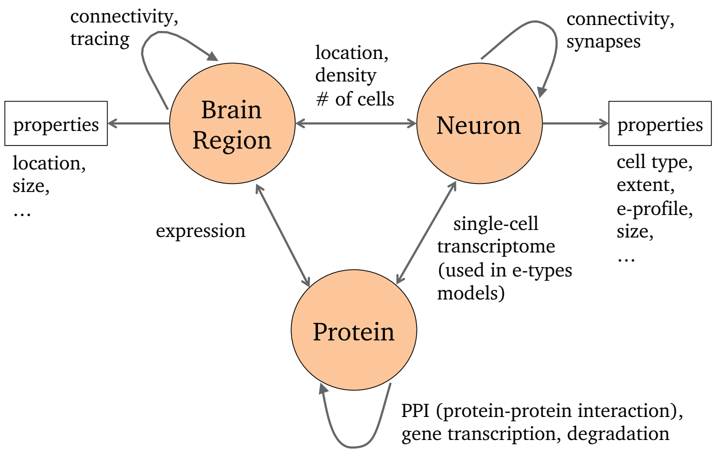


Fig. 1.1: Three major neuroscientific entities of interest and their relationships. Circles represents the main entities of interest in neuroscientific NLP within the context of this thesis. Arrows represents relationships of interest between these entities. Relationship studied in this thesis include brain region connectivity (see Section 4.2) and neuron properties (see Section 4.3).

natural language generation, optical character recognition, machine translation, or automatic summarization which are out of the scope of this thesis.

bioNLP

bioNLP emerged as a sub-discipline of NLP in order to focus on biomedical entities and events, and to address specific challenges in that domain. For example, models have been developed to recognize diseases and brain regions, or systems to identify protein-protein interactions or brain-region connectivity (see Section 2 for an extended list of bioNLP models and systems). bioNLP as a research community has been shaped by shared tasks like BioCreative [Lu+11] and BioNLP [KP13]. One of the oldest and seminal task of bioNLP was the creation of named entity recognizers for proteins and genes, based on the Genia annotated corpus [Kim+03].

neuroNLP

In the context of neuroscience, NLP focuses on entities like brain regions, neurons, proteins and the interaction between them (see Figure 1.1). For example protein-protein interaction [Bjo+11], electrophisiological properties of neurons [Tri+14] or brain-region connectivity (Section 4.2). See [AC12; Bur+08] for a review of text-mining approaches in neuroscience.

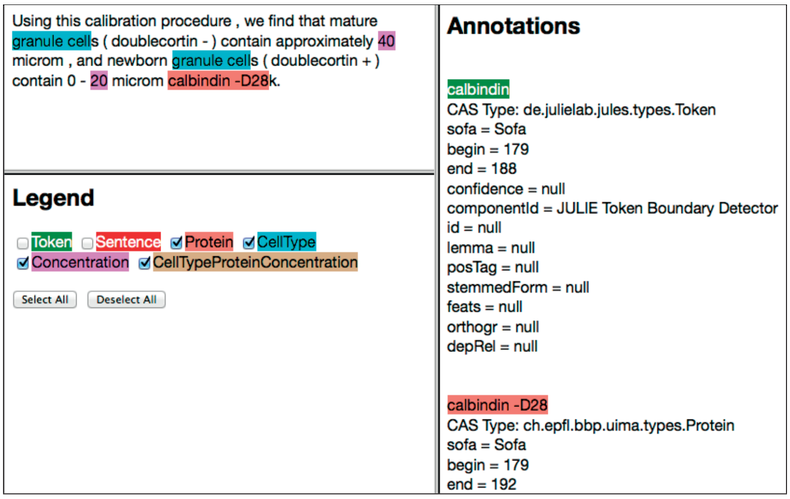
Note that although it is framed in a neuroscientific research environment, this thesis is neither focusing on psycholinguistics (how do people learn and process language) nor on neurolinguistics (where in the brain is language located). Rather, it aims at developing efficient NLP systems to facilitate knowledge extraction from the neuroscientific literature.

1.2 Introductory Story

context of my research

In this section, I explain the context of my research within the Blue Brain Project (BBP) and describe the first *in-litero* experiment we were asked to perform to extract protein

Fig. 1.2: Early prototype of a UIMA-based graphical user interface for the extraction of protein concentrations in cell types. The sentence in the top left panel is annotated for instances of proteins, cell types and concentrations. The right panel provides detailed information on individual entities.



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

Legend

□ Token □ Sentence ☑ Protein ☑ CellType
☑ Concentration ☑ CellTypeProteinConcentration

Annotations

calbindin
CAS Type: de.julielab.jules.types.Token
sofa = Sofa
begin = 179
end = 188
confidence = null
componentId = JULIE Token Boundary Detector
id = null
lemma = null
posTag = null
stemmedForm = null
feats = null
orthogr = null
depRel = null

calbindin -D28
CAS Type: ch.epfl.bbp.uima.types.Protein
sofa = Sofa
begin = 179
end = 192

concentration in cell types from neuroscientific articles. I describe the evolution of my understanding of NLP methodologies, and of the neuroscience field that led to the development of the agile text mining methodology, the core of this thesis. The first experiment we were asked to perform was to support BBP's modelization effort[7] by providing neuroscientists with a searchable database of protein concentrations, classified by cell types. These protein concentrations were to be extracted from scientific articles published in peer-reviewed papers related to neuroscience. Figure 1.2 shows an early prototype to automatically annotate sentences for proteins, concentrations, and cell types. Our experiment shall provide answers to the following questions:

- In which cell type is a given protein present[8]?
- At which concentration?
- From which papers/data sources does this information come from?

manual search

There had been ongoing efforts at BBP to curate scientific articles for protein concentrations. The procedure was to resort to Google Scholar and perform a manual search for a cell region, e.g. `"pyramidal cell"` (mind the quotes). Researchers would then manually analyze the resulting list of abstracts (between 10 and 150 abstracts per query). In each abstract, they would search for the keyword `concentration` and additionally scan tables and headers. The extracted protein concentration would be reported in a spreadsheet or database. The above procedure has proved to be tedious and very time consuming. As of August 2015, the database contains 206 records from 6 distinct publications[9].

challenges

Instead of serving as an initial success-story in neuroscientific information extraction, this first experiment proved more difficult than expected and opened up several challenges related to biomedical information extraction.

[7] In this case, sub-cellular modeling.
[8] Note that it is also of interest to find out that a protein is *not* present in a given cell type.
[9] PMIDs 21874189, 17243894, 22764236, 17110340, 15548210 and 16027175.

concentrations

First, information about *concentrations* is reported in various forms and units in scientific articles, for example `1.25 g/l` or `35 ± 2 µg/m3`. Often, the concentration is stated as a number of copies, e.g. Table 2 of [PMID 17110340] reports the presence of `10.3 Rab3A copies per analyzed vesicle`. In other cases, the only available information is the presence (or absence) of a protein, for example the statement "Additionally, high concentrations of AA can lead to PKC translocation and activation in the model" in abstract [PMID 22764236]. To address this issue and enable the extraction of concentrations and other measures, we developed an NLP module that recognizes and normalizes most units and measures used in scientific paper (see Section 3.2.3 for the aforementioned module and for more examples of units).

cell types

Second, the identification of *cell types* also proved to be challenging because the definition of cell types is disputed within the neuroscience community [MA13]. Despite some major effort to structure the naming of cells (see e.g. [Asc+08]), the definition of neural cell types by and large represents an open issue in the neuroscience community. This led to the development of *neuroNER*, a general approach for identifying and normalizing mentions of specific neuron types from the biomedical literature (see Section 4.3).

proteins

Third, identification and normalization of *proteins* mentions is a non-trivial task. Over 20,000 different proteins have been discovered for humans and although there exists some naming for human proteins[10], these are not always followed. Our initial approach was to use UniProt, a publicly available knowledge base of protein sequence and functional information [Con+08]. However, UniProt was not primarily designed to serve as a lexical resource for NLP. In particular, it lacks extensive synonyms and lexical variations, resulting in low recall[11]. Our approach was the manual creation of a lexical-based NER, starting with a limited list of proteins that were most important from a neuroscience point of view. This raw list was compiled by two BBP researchers. It contained names of proteins and genes, together with abbreviations. Some entries were extremely specific (e.g. "Plasma membrane calcium transporting ATPase 2"), while others entries potentially exhibited high polysemy (e.g. "ras" or "ga"). Additionally, surface forms were not linked to entities. It was soon realized that this list could not be used for a lexical-based NER in such a raw form. Eventually, a more compact and consistent list of approximately 300 proteins and genes was manually compiled by a neuroscientist, drawing from terms and synonyms lists from NCBI and UniProt (see Section 4.3).

extremely low recall

One more challenge was the unrealistic expectation in terms of information extraction. This became obvious, as it was difficult for domain experts themselves to come up with a significant number of protein concentration mentions from the literature (*extremely low recall*). These unrealistic expectations also became apparent during a first proof-of-concept, which consisted of a simplified experiment to extract concentrations occurrences of a single protein (cell types were not extracted). Even for that simplified experiment, it was challenging to find relevant sentences among PubMed abstracts. For example, we started by focusing on `SNAP-25`, an important protein for neuroscience involved in mediating vesicle docking and fusion with the presynaptic membrane in neurons. Our initial system was based on a PubMed query for abstracts containing "Synaptosomal-Associated Protein 25[mesh]" or "SNAP-25" (1711 results in total as of October 2011). We then identified concentrations with

[10] `http://www.genenames.org/`

[11] *Recall* is the ratio of the number of relevant records retrieved to the total number of relevant records. In this example, missing synonyms will result in relevant records not being retrieved.

Tab. 1.1: PubMed abstracts containing SNAP-25 and a concentration. Selected through a PubMed search for “Synaptosomal-Associated Protein 25[mesh]” or “SNAP-25”, subsequently filtered for the presence of a concentration annotation in the same sentence. Concentrations are underlined.

Incubation periods of 24 h and 48 h in 50 mM KCl increased *SNAP-25* levels in hippocampal explants and PC12 cells, but not on cerebellar explants (Sepúlveda CM et al., 1998)

Otherwise, a 24 h incubation with 10 microM AA increased *SNAP-25* expression only in hippocampal explants, although 100 ng/ml phorbol 12-myristate 13-acetate (PMA) did not have effect (Sepúlveda CM et al., 1998)

In intact cells exposed to 66 nM BoNT/A, virtually all of the *SNAP-25* was truncated, accompanied by a near-complete inhibition of exocytosis; however, after their permeabilization a significant level of secretion was recorded upon Ca2+-stimulation. (Lawrence GW et al., 1997)

In HIT cells, a concentration of 30-40 nM BoNT/E gave maximal inhibition of stimulated insulin secretion of approximately 60%, coinciding with essentially complete cleavage of *SNAP-25*. (Sadoul K et al., 1995)

However, during the secretion of insulin stimulated with glucose (15.6 mM), 1 microM arsenite decreased the activity of calpain-10, measured as *SNAP-25* proteolysis. (Díaz-Villaseñor A et al., 2008)

SNAP-25 was mainly distributed in the plasma membrane at 2.8 mM glucose, whereas the syntaxin 1A distribution in the plasma membrane, as compared to the cytosolic fraction, was highest at 8.3 mM glucose. (Andersson SA et al., 2011)

Here we describe Ca(2+)-dependent interaction of this site with syntaxin and *SNAP25* which has a biphasic dependence on Ca2+, with maximal binding at 20 microM free Ca2+, near the threshold for transmitter release. (Sheng ZH et al, 1996)

the above-described module, using simple collocation of concentrations and measures at the sentence level. Not only was the number of returned abstracts very small (7 PubMed abstracts), but also none of the extracted concentrations were effectively related to SNAP[12] (see Table 1.1).

limited quantities of raw textual data

A fifth challenge is that raw textual data is available in limited quantities. While all PubMed abstracts can be licensed for text mining purposes, full-text papers impose much stricter access policies[13]. This turned out to be a serious drawback, since full-text articles contained significantly more relevant neuroscientific information than abstracts (see Section 3.1.1). In order to process large amounts of data, several large corpora of abstracts and full-text articles related to neuroscience were developed during the course of this thesis (see section 3.1). Acquiring information from full-text articles is further complicated by the fact that most of them are available only in PDF. PDF is a presentational format and various preprocessing tools had to be developed to provide precise text extraction (see Section 3.1.2). In addition, a significant amount of relevant data is located in the tables of articles. Identifying those tables and extracting their content proved to be a difficult task.

[12] That is, the concentrations were related to other entities, but were returned in the search results because they collocated in the same sentence as SNAP.

[13] There are however signs that publishers are opening up to the possibility of allowing text and data mining. See for example the roadmap signed by several prominent publishers to enable text and data mining for non commercial scientific research in the European Union [@Stm]

context

A last but important challenge is that even when it would be possible to extract events with high precision[14], the context surrounding these events is necessary to interpret and correctly understand the event. In our case, providing neuroscientists with a database of protein concentrations is not sufficient. Neuroscientists need to know the conditions in which these measures were generated. This contextual information is usually provided in the “materials and methods” section of an article. The necessity to provide context led to the development of user interfaces enabling researcher to quickly navigate to the original article (see Section 4.2).

requirements for further experiments

Based on this initial request to automate information extraction at the BBP, the focus of my research has shifted toward experiments exhibiting the following properties:

- realistic expectations from domain experts, both in terms of precision and recall; the former ensures that expected results can be reasonably automatically extracted; the latter ensures that expected results actually exist in the literature,
- strong commitments of both neuroscience researcher and NLP researcher regarding collaboration and communication,
- sufficient amounts of accessible raw textual data,
- means to evaluate a tasks’ performance (or willingness to create evaluation data),
- availability of NLP models like NERs (or possibility to create them).

1.3 Research Framework and Contributions

This section lays down the three central research questions of this thesis. The first two ones investigate the benefits and implications of using very large corpora for neuroscientific NLP (1.3.1). The third inquires the most effective methodology to develop specialized text mining solutions and to facilitate the communication and collaboration among stakeholders during that development (1.3.2). Subsequently, the contributions of this thesis to the aforementioned research questions are stated (1.3.3).

1.3.1 Large Scale NLP for neuroscience

> ” *It never pays to think until you’ve run out of data.*
>
> — **Eric Brill**
> [BB01]

In the foreseeable future, it is very unlikely if not impossible that NLP systems will come close to the level of sophistication of humans when it comes to understanding and analyzing a single scientific article. There is simply too much implicit knowledge and too much subtleties and ambiguities in written language for an NLP system to make sense of a single article as well as a human would do[15]. At the same time, we have to acknowledge that it is impossible to expect neuroscientists to read and keep up with the growing amount of published articles.

[14] *Precision* is the fraction of retrieved records that are relevant.

[15] For example, tasks like anaphora resolution or the detection of irony or humor are still challenging for NLP.

Thus, we need to stop thinking about humans and machines competing against each other for the task of understanding textual resources. Instead, it pays to think how to design systems leveraging the strengths of both humans and machines. Regarding that matter, one undisputed strength of computer systems is that they can be deployed on large amounts of data, leading to the first research question:

RQ1: How can we capture all the relevant neuroscientific textual data and how can we process it at scale?

One simple way to double the throughput of an NLP system is to simply double the amount of input data[16]. Although this procedure seems trivial, there are at least three barriers to have access to a larger amount of neuroscientific text. First, the data is often locked in scientific journals requiring hefty subscription fees. Second, most articles are only available in the PDF format, making the accurate extraction of raw text, tables and figures challenging. Third, processing such massive amounts of text requires an adequate computing infrastructure. Going back to the above quote from Brill, how are we to avoid running out of data in the first place?

RQ2: How can search and analysis results become more robust and useful as we analyze more data?

Very often, the usability of search interfaces deteriorates as more input data is incorporated. In other words: how can we really benefit from processing more data? The reason is that typically precision decreases as we try to increase recall. This can be observed in typical plots of precision versus recall curves. Thus, the challenge here is to not to downgrade precision as more input is provided to the system.

1.3.2 Agile Text Mining

> *We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*
>
> - *individuals and interactions over processes and tools*
> - *working software over comprehensive documentation*
> - *customer collaboration over contract negotiation*
> - *responding to change over following a plan*
>
> *That is, while there is value in the items on the right, we value the items on the left more*
>
> — **The Agile Manifesto**

Over the years, there has been a steady trend towards developing highly specialized text mining applications (TMAs) addressing very specific use cases. For examples, there exists several specialized TMAs to extract information from scientific papers about protein-protein interactions [Ana+10], brain-region interactions [Ric+15], or neuron-specific electrophysiology [Tri+14]. These TMAs deliver high-precision results by focusing on a very specific task within a narrow domain and single language. However, these specialized tools require

[16] With the hypothesis that the additional data contains the same amount of relevant information than the existing data.

the creation of specific resources (e.g. stop-word lists, training corpus for machine learning models, ontologies) and the development of custom TMAs for each new application. Hence the necessity of cost-effective methodologies for developing specialized TMAs, leading to the following research question:

RQ3: What methodologies are required to efficiently and collaboratively develop custom TMAs?

1.3.3 Contributions

In this section, we list our contribution to the above stated research questions.

RQ1: How can we capture most of the relevant neuroscientific textual data and how can we process it at scale?

We develop tools and methods to accurately handle and preprocess full-text PDF articles (3.1.2). These tools have demonstrated to greatly improve the text representation of PDFs articles and thus the information extraction quality. Subsequently, we introduce *bluima*, an NLP pipeline for information extraction of neuroscientific content at PubMed scale (3.2). bluima is specifically dedicated to processing neuroscientific corpora. E.g. it includes numerous named entity recognizers for neuroscientific entities. Moreover, it was capable of ingesting very large corpora (in the order of magnitude of several billion tokens) by deploying it on a computer cluster. Using these tools, we create very *large corpora of neuroscientific literature*. To the best of our knowledge, no neuroscientific text-mining experiments were performed with corpora of a comparable size.

bluima

large neuroscientific corpora

RQ2: How can search and analysis results become more robust and useful as we analyze more data?

Our approach is to provide neuroscientists with tools allowing them to properly understand the extracted results. This is achieved by generating meaningful aggregations of results that do not swamp them with information, but rather normalize data and improve their understanding of it (see e.g. Figure 4.16, page 78). We also always provide ways of accessing the original article from where information was extracted and provide feedback, resulting in an interface that improves as more researchers use it (e.g. Figure 4.17 page 79). In the case of neuroNER, the extraction rules themselves are written in a scripting language that is quite understandable (see Figure 3.3). It means that, even though a domain expert might not be able to write these rules, he or she can certainly easily understand them, and possibly discuss them with the text mining expert in order to improve them. We believe that it is a critical benefit for a user to have simple ways to understand an analytics system, instead of being left with a black box.

put the domain expert in the middle

RQ3: What methodologies are required to efficiently and collaboratively develop custom TMAs?

The development of specialized TMAs requires the close collaboration of two main stakeholders: subject matter experts (SME) and text mining practitioners (TM). The quality of that collaboration is key to the performance of the TMA. SMEs are for example sports goods marketers who want to enrich their content with semantic information, or biologists seeking

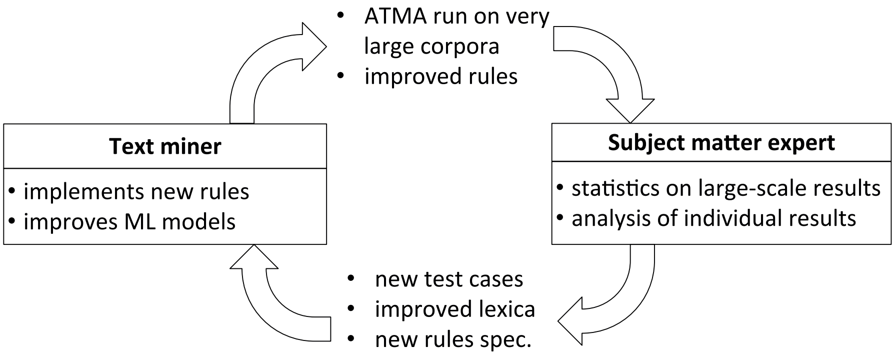


Fig. 1.3: Iterative development cycle of an agile text mining application.

to extract gene mentions and interactions from the scientific literature[17]. SMEs know *what* information can be extracted, how it is structured and possibly which knowledge bases are available (e.g. in the forms of ontologies or lexica). TMs, on the other side, know *how* to extract information. They are familiar with a wide array of natural language processing (NLP) algorithms (e.g. tokenizers, lemmatizers, named entity recognizers, topic models) and their usage (e.g. parametrization, scoring models). TMs know how to manage the large scale deployment of a TMA to process millions of documents.

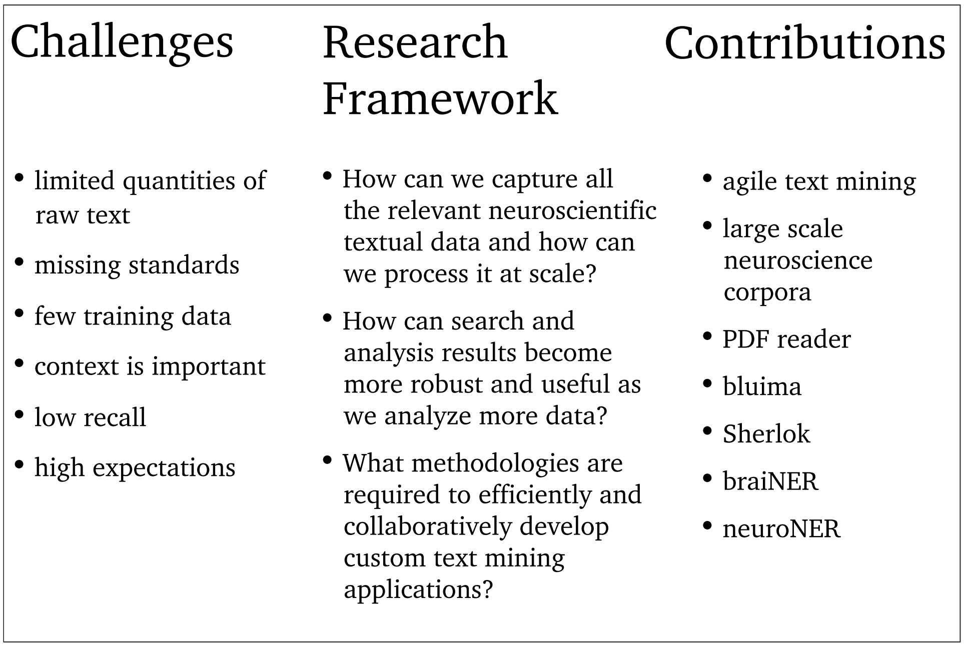
In order to develop TMAs effectively, SMEs ought to have simple means to access the results of an analysis in order to evaluate the performance and provide rapid feedback. Furthermore, for reproducibility all models and resources must be versioned and well tested to ensure continuous improvement. TMs, on the other side, must be able to incrementally improve their models and seamlessly release them so that SMEs can continuously evaluate them. The TMA must provide them with a domain-specific language (DSL) to efficiently write and compose NLP components. Both SME and TM should have a high-level overview of understanding of each other's work (no black box).

agile text mining

This third research question led to the formulation of agile text mining, a new methodology to support the development of efficient TMAs. Agile text mining copes with the unpredictable realities of creating text-mining applications. It is inspired by the *Agile Manifesto* [HD14]. Agile text mining applications (ATMA) are developed during *short iteration cycles*, lasting from a few hours to a few days (see Figure 1.3). Short iteration cycles allow for frequent redefinition of priorities and for rapid adaptation to changing requirements. Each iteration starts with the selection of the most valuable features to implement during that cycle. The first iteration cycle is deliberately short: the goal is not to deliver a perfect system at first, but to get started with a very basic proof of concept. Usually, that first iteration involves combining existing NLP components and resources into a minimal system. The output of every iteration is a complete, working system that is continuously deployed on an annotated corpus or on a medium-size corpus to evaluate it performance.

SMEs have constant access to the latest analysis artifacts that allow them to write new functional tests to communicate how the system should perform. These tests also guarantee that newer development will not break previous development and results. Additionally, SMEs improve and develop new linguistic resources (e.g. lexica, ontologies or annotated

[17] We stress the fact that in this thesis, we define SMEs as experts in the domain targeted by the TMA, not in text mining.



Challenges

- limited quantities of raw text
- missing standards
- few training data
- context is important
- low recall
- high expectations

Research Framework

- How can we capture all the relevant neuroscientific textual data and how can we process it at scale?
- How can search and analysis results become more robust and useful as we analyze more data?
- What methodologies are required to efficiently and collaboratively develop custom text mining applications?

Contributions

- agile text mining
- large scale neuroscience corpora
- PDF reader
- bluima
- Sherlok
- braiNER
- neuroNER

Fig. 1.4: Overview of challenges (see section 1.2), research framework (see section 1.3) and contributions (see section 1.3.3).

corpora for supervised ML model training) and new rules for information extraction. Based on these, TMs implement new rules and models to validate these new test cases. An ATMA allows TMs to perform a scale out on very large corpora without substantial modification of the analysis system and with minimal deployment effort. Thus, an ATMA should be designed to scale horizontally, that is: grow sub-linearly in terms of costs and complexity as data size grows. All models and resources of the ATMA shall be versioned to ensure reproducible analytics at any point in time.

In section 3.3, we introduce Sherlok, a system to design *agile text mining applications* (ATMA), that implements the above requirements.

Sherlok

Figure **??** gives an overview of this thesis' challenges, research framework and contributions.

1.4 Reader's guide

The structure and dependencies in this thesis are presentend in Figure 1.5 below. The present Chapter 1 provided some background into neuroscience and natural language processing (1.1), as well as an introductory story of the context surrounding this thesis (1.2). In (1.3), research questions and contributions were stated: large scale NLP for neuroscience advocating for the use of massive corpora for NLP (1.3.1) and agile text mining, a methodology to efficiently develop text mining applications (1.3.2).

Chapter 1
Introduction

Chapter 2 reviews the available textual resources for biomedical NLP (2.1) and the state-of-the-art methods used throughout this thesis: lexical and machine learning named entity recognizers (NERs, 2.3). Information extraction is reviewed and introduced as a way to

Chapter 2
Methods

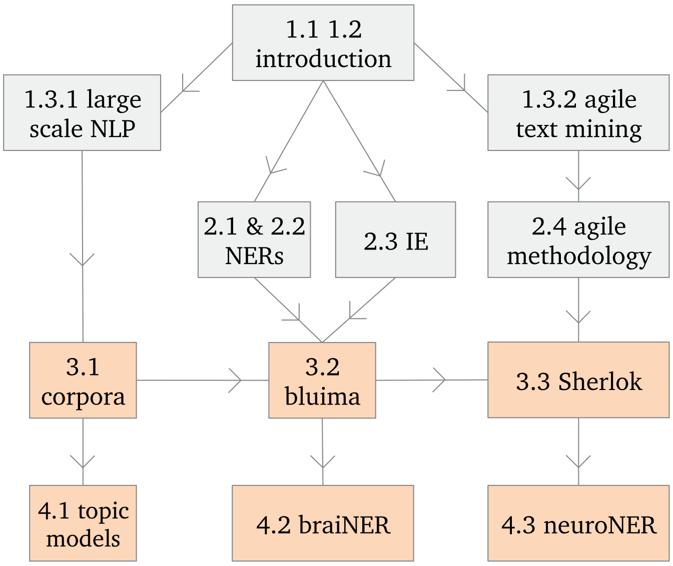


Fig. 1.5: A reader's guide to the structure and dependencies in this thesis. Orange blocks represent contributions while gray blocks represent

improve manual literature search (2.4). Agile methods for text mining are reviewed in Section 2.5.

Chapter 3
Achievements
and tools

Chapter 3 presents the *tools* developed during this thesis. The first section (3.1) describes the *corpora* that will be used in subsequent experiments, including their preprocessing. The second section (3.2) introduces *bluima*, a natural language processing (NLP) pipeline focusing on the extraction of neuroscientific content. The last section (3.3) introduces *Sherlok*, an NLP system supporting the development of agile text mining applications.

Chapter 4
Experiments

Chapter 4 presents the *experiments* conducted during this thesis. The first section is dedicated to *topic models* (4.1), a type of unsupervised machine learning models for discovering the hidden thematic structure in document collections. Section 4.2 presents text-mining models to extract and aggregate *brain regions connectivity* results from a large corpus of 8 billion words. We demonstrate the usefulness of these models through evaluations against in-vivo connectivity data and against manual review of the literature. The last section (4.3) introduces *neuroNER*, an NLP model to perform automated identification and normalization of neuron type mentions in the neuroscientific literature. This kind of decomposition and normalization is essential for cross-laboratory studies, since neuroscience currently lacks consistent terminologies or nomenclatures for describing neuron types.

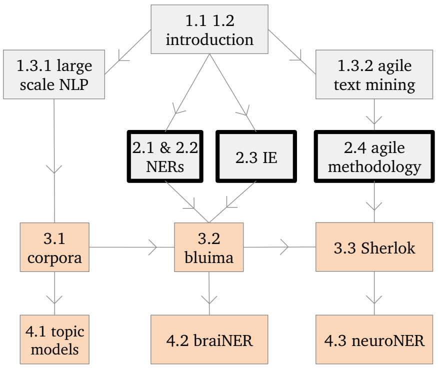
Chapter 5
Synthesis

Chapter 5 concludes with a synthesis and future research directions.

Methods

2

This chapter explains and discusses the methods relevant to this thesis[1]. In the first section (2.1), textual resources available for bioNLP are presented, for example ontologies, taxonomies and annotated corpora. The second part of this chapter (2.3) is devoted to named entity recognition (NERs), in particular lexical-based and machine learning-based approaches. Information extraction is reviewed and introduced in Section 2.4 as a way to improve manual literature search. Agile methods and NLP framework supporting agile development of text mining solutions are reviewed in Section 2.5.



Reader's guide (Section 1.4 p. 12).

Criterion for the selection of these methods where the followings:

- The methods are applicable and suitable to neuroscientific or biomedical corpora. For example, we only considered tokenizers and syntactic parsers that have been specifically trained on (and for) biomedical corpora. Domain-independent methods are also considered.
- Outputs (e.g. entities, events) are relevant and valuable in the context of neuroinformatics. This includes entities like neurons, brain regions and neurological diseases, but also more general entities like proteins, genes and species (compare with Figure 1.1 on page 4).
- Methods and systems have been thoroughly evaluated, including proper cross-validation and inter-annotator agreement metrics.
- Systems are publicly available and/or open-source, whenever possible.

2.1 Textual resources for biomedical NLP (bioNLP)

A growing number of resources are available for biomedical NLP (bioNLP). In this section, we list annotated corpora, lexica, ontologies and brain atlases.

Annotated corpora

Annotated corpora are lexical resources that have been manually annotated with entities of interest. They represent highly valuable textual resources for bioNLP, enabling to evaluate

[1] Methods specific to a single section are described in that particular section.

an NLP system and train machine learning models on it. Many of these corpora were specifically generated for a workshop's shared task. For example, a corpus of 20,000 annotated sentences annotated for gene names was created for the BioCreative II shared task [Wil+07]. This corpus was extended for the subsequent BioCreative III gene normalization shared task [Lu+11] to include full-text articles that were species non-specific and thus moving closer to a real literature curation task. Other shared task like BioNLP-ST [Néd+13; Kim+12] have produced annotations that have been centralized in PubAnnotation[2], a repository of biomedical text annotations [KW12]. PubAnnotation represents a new collaborative and open way of publishing text annotations using recent web technologies. The CRAFT corpus [Bad+12] is a large annotated corpora consisting of 97 full-text biomedical articles annotated for chemicals, cells, genes, species, proteins and sequences. CALBC was the first large-scale silver standard corpus that, unlike a gold-standard corpus, contains annotation resulting of the harmonization of an ensemble of NERs [RS+10]. The goal was to avoid the production of manually-annotated gold standard corpora that are time-consuming and costly. It contains a large number of annotations (1,121,705) from 100,000 Medline abstracts, annotated for proteins, genes, diseases and species The NCBI disease corpus is built as a gold-standard resource for disease recognition [Doğ+14]. It contains 795 PubMed abstracts annotated at the mention and concept level.

gene names

CRAFT

CALBC

disease

More focused on neuroscience, the WhiteText corpus contains annotations about *brain regions* and brain connectivity. 3205 Medline abstracts were manually annotated with 17,585 brain region mentions and 5,208 connectivity statements [Fre+15]. Burns et al. manually annotated 21 Medline abstracts about *tract-tracing* experiments with annotations about brain regions, injection location, labeling location and tracer chemical [Bur+08].

brain regions

tract-tracing

Lexica and Ontologies

An ontology is a formal representation of knowledge in a domain. An *ontology* goes beyond a taxonomy or a controlled vocabulary by the richness and expressiveness of relationships between entities. It provides a consistent abstraction to link experimental data with concepts. While the goal of some ontologies is to formalize as much knowledge as possible, most biomedical ontologies take a more *pragmatic* approach and attempt to create a structure enabling clear communication about *existing* experimental data [LM09].

ontology

The *Gene Ontology* (GO) is a major bioinformatics initiative to standardize the representation of gene attributes across species. At the moment, it contains over 40,000 biological concepts used to annotate gene functions based on over 100,000 scientific papers. The GO is part of a larger effort called the Open Biomedical Ontologies (*OBO*) [Smi+07] federating and coordinating the development of biomedical ontologies. OBO is a growing collection of ontologies designed to be interoperable and logically well formed. For example, OBO contains an ontology of cell types covering the prokaryotic, fungal, animal and plant worlds and consisting of over 680 cell types classified under several generic categories [Bar+05].

Gene Ontology

OBO

UniProtKB/Swiss-Prot[3] is a database of high-quality, manually annotated, non-redundant protein sequences [Dim+11]. *Swiss-Prot* does not provide a direct taxonomy or ontology,

Swiss-Prot

[2] `http://www.pubannotation.org`

[3] http://www.uniprot.org/help/uniprotkb

but provides links to the GO through the UniProt-GO annotation database. UniProtKB consists of two sections: Swiss-Prot containing manually annotated and evaluated records and TrEMBL consisting of computationally analyzed records awaiting full manual annotation.

SAO

SAO is an ontology of subcellular neuroanatomy ("mesoscale"), encompassing cellular and subcellular structure, supracellular domains, and macromolecules [Lar+07]. Its goal is to provide the knowledge necessary to integrate data acquired across multiple scales in neuroscience.

Medical Subject Headings

Medical Subject Headings[4] (MeSH) is a thesaurus used for subject headings. It has a low granularity but has the great advantage to be available for most PubMed articles. See section 4.1.4 for a detailed description.

BioLexicon

The *BioLexicon*[5] is a large-scale English terminological resource developed to facilitate biomedical text mining. It contains over 2.2M lexical entries, over 3.3M semantic relations, and information on over 1.8M variants and on over 2M synonymy relations [Tho+11; Sas+08]. Sasaki et al. [Sas+09] present three applications of the BioLexicon: a dictionary-based POS tagger, a syntactic parser, and query processing for biomedical information retrieval.

NIFSTD

The NIF standardized ontology (*NIFSTD*) consists of common neuroscience domain terminologies structured into a unified representation of the biomedical domains typically used to describe neuroscience data (e.g., anatomy, cell types, techniques), as well as digital resources (tools, databases) being created throughout the neuroscience community. [Ima+12; Ima+11; Bug+08]

NeuroLex

Acknowledging that curation efforts is still very manual, highly technical, and therefore costly, NIF developed *NeuroLex*[6], a semantic wiki to interface with the NIFSTD and enable community-driven curation of neuroscientific terms [LM13]. Additionally, NeuroLex offers machine-readable knowledge representation through application public interfaces (APIs). As of today, NeuroLex is tracking almost 35,000 unique neurobiological entities (e.g. experimental techniques, anatomical nomenclature, genes, proteins and molecules).

It must be noted that while ontologies, taxonomies or lexica are essential resources for bioNLP, their primary purpose is often not to act as such (see discussion in Section 2.3 below).

Brain Atlases

We review selected available brain atlases and lexica, as they are a central resource for neuroscientific NLP.

NeuroNames

NeuroNames[7] was one of the first popular neuroanatomical terminologies in the field. It consists of more than 15,000 neuroanatomical terms. It partitions the brain in about 550 primary structures to which all other structures, names, and synonyms are related [BD03].

[4] http://www.ncbi.nlm.nih.gov/mesh

[5] http://www.ebi.ac.uk/Rebholz-srv/BioLexicon/biolexicon.html

[6] http://neurolex.org/

[7] http://braininfo.rprc.washington.edu/Nnont.aspx

CoCoMac

CoCoMac[8] (Collations of Connectivity data on the Macaque brain) is an manual curation approach to produce a systematic record of the known wiring of the primate brain [Bak+12b; Ste+01]. The database has become by far the largest of its kind (primate), with data manually extracted from more than four hundred published tracing studies.

BAMS

The Brain Architecture Management System (*BAMS*) is a large inventory of data and metadata collated from original literature [Bot+12; BS08][9]. Neuroscientists from the BAMS project have manually curated over 600 scientific articles. They analyzed each article (including tables, images and supplementary materials) and assessed the quality of the experiment. Finally, they normalized brain region mentions to the BAMS ontology, and recorded the connectivity data into a structured database (including directionality and strength).

Allen Brain Atlas

The *Allen Brain Atlas*[10] seek to combine genomics with neuroanatomy, with the goal to advance the research on neurobiological diseases such as Parkinson's, Alzheimer's, and Autism with their mapping of gene expression throughout the brain. Its nomenclature was adapted from [Swa04], [Hof+00], BAMS [Bot+12], and BrainInfo[11], as described in [@All]. It identifies 1,000 anatomical sites in the human brain, backed by more than 100 million data points that indicate the particular gene expression and underlying biochemistry of each site. Also available to the public is the Brain Explorer 3D viewer[12].

BigBrain

BigBrain is a ultrahigh-resolution 3D digital atlas of the human brain [Amu+13]. It has a nearly cellular resolution (20 μm) and consists of 7404 histological sections acquired through MRI and subsequently processed by neuroscientists to remove artifacts and by software to align them.

2.2 Text Preprocessing

Text preprocessing, like segmentation of sentences or part-of-speech tagging, is a very important step in an NLP pipeline. It has been extensively researched and there exists several components that have been trained specifically for bioNLP and that generally deliver very good precision and recall. Below we describe some NLP models for efficient preprocessing of biomedical text.

tokenization

The NLP tool suite[13] from the Jena University provides preprocessing tools like *sentence splitters, word tokenizers and POS taggers*. The ClearTK project also provides similar tools, some trained on biomedical corpora [Ogr+08].

lemmatization

Biolemmatizer [Liu+12] is a lemmatizer trained for English biomedical literature, achieving a state-of-the-art accuracy of .99 on a sampled set of the CRAFT corpus.

anaphora resolution

Several papers with various performance (e.g. [GB08]) have been published on *anaphora resolution* for the biomedical domain, but none published their models.

abbreviation recognition

BioAdi performs *abbreviation recognition* using a trained CRF model [Kuo+09]. On their

[8] http://cocomac.g-node.org/

[9] http://brancusi.usc.edu/

[10] http://www.brain-map.org/

[11] `http://www.braininfo.org`

[12] http://mouse.brain-map.org/static/brainexplorer

[13] `http://www.julielab.de/Resources/Software/NLP_Tools.html`

annotated corpus of 1200 PubMed abstracts, their system achieved a state-of-the-art F-score of .86 with .93 precision at .80 recall. [Oka+10] developed a supervised approach for clustering expanded forms, achieving a 0.984 accuracy and 0.986 F1 score on an experiment of abbreviation disambiguation.

chunkers

Kang [Kan+11a] compares six chunkers for biomedical text, of which OpenNLP performed best (F-scores .89 for noun-phrase chunking and .95 for verb-phrase chunking, on the GENIA Treebank corpus).

syntactic parsers

Several syntactic parsers have been trained on biomedical corpus, e.g. Wang [Wan+10b] trained the ENJU [MT08] and Stanford parsers [DM+06] on the GENIA dataset []. Miyao [Miy+09] compares several parsers, and evaluates them on protein-protein interaction (PPI) extraction from biomedical papers.

2.3 Named Entity Recognition (NER)

Several named entity recognizers (NERs) are publicly available to identify neuroscience-relevant entities like proteins or brain regions.

lexical-based NERs

To build a NER, the first and simplest approach is to match entities from a list of surface forms. These are called *lexical-based NERs*. As reviewed above, several biomedical ontologies and taxonomies are available and these can be used to build a lexical-based NER. However, most have been designed to *structure and organize* a domain, but not to serve as a NER resource. Typically, they lack appropriate synonyms and can be too specific, resulting in low recall (for example, "Entorhinal area, lateral part, layer 6a" is a brain region from the ABA ontology that is highly unlikely to be found in any scientific article). Another issue with lexical-based NERs is their lack of context. For example, there is a gene synonym named “for” [14] that would be often confused by a lexical-based NER with the preposition of the same name. Despite these disadvantages, lexical-based NERs are simple to create and several such NERs were created during this thesis (see Table 3.7 on page 34).

machine learning-based NERs

A second and more sophisticated approach to building a NER is to train a *machine learning* (ML) model on annotated corpora providing examples of entities that are to be recognized. The model relies on so-called *features* to take a decision on whether a group of words represent an entity. Features can be, for example, that a word starts with a capital letter, whether the word belongs to a lexicon or whether the previous word is a verb. A model can includes several hundreds different features and model training consists in learning which combinations of features are most likely to identify an entity. Once a model has been trained on an annotated corpus, it can be used to identify entities on new, unseen text. The advantage is that the model will match complex entities, even if they are not present in any lexica. For example the brain region names “contralateral prepositus hypoglossal nucleus” or “distal parts of the inferior anterior cerebellar cortex” would be correctly identified even though they never appear in a corpus. However, a drawback of supervised ML is that corpus annotations, required to train the model, are very time-consuming and require domain expert knowledge. Another drawback is that unlike lexical-based NERs that commonly associate an entity with a unique identifier, ML NERs require additional steps to normalize and associate a recognized entity with a unique identifier. One last drawback of ML NERs is that their

[14] `http://www.genenames.org/cgi-bin/gene_symbol_report?hgnc_id=12799`

performance can potentially degrade if they are not applied to the same kind of corpus than the one they have been trained on. For example, when training a NER on PubMed abstracts, it will not necessarily perform well on full-text articles.

proteins and genes

Several NERs have been published to identify for *proteins and genes*. State-of-the-art performance is high: Ando et al. [And07] achieved an F-score of .87 on the BioCreative2 task, using a semi-supervised approach that incorporates large amounts of unlabeled data. However, their model has not been published. An open source alternative is BANNER[15] is open source and achieves a near-state-of-the-art F-score of .84 on the BioCreative2 task. GeNo, an open-source system for gene name recognition and normalization achieved an F-measure performance of .86 on the BioCreAtIvE-II test set [Wer+09]. Gimli is a state-of-the-art protein and gene NER, achieving an average F-score of .87 on the BioCreative2 task [Cam+13].

chemicals

OSCAR[16] is a mature NER for *chemical entities* and chemical reactions [Jes+11].

anatomical entities

AnatomyTagger is an open-source machine learning-based NER for *anatomical entities* ranging from subcellular structures to organ systems [PA14]. AnatomyTagger has been trained on the AnatEM corpus consisting 13,000 annotations of anatomical entities grouped in 12 types such as Cellular component, Tissue and Organ.

brain regions

For *brain regions*, NER models have been published by Burns et al. [Bur+08] and French et al. [Fre+15]. They both rely on linear chain conditional random fields , with model features based on morphological, lexical, syntactic and contextual information. French's model achieves a state-of-the-art performance of 86% recall and 92% precision on a training corpus of 1,377 abstracts with 18,242 brain region annotations [Fre+12].

species

Linnaeus [Ger+10] is a *species* NER that uses a dictionary-based approach and a set of heuristics to resolve ambiguous mentions about species (97% of all mentions in PubMed Central full-text documents resolved to unambiguous NCBI taxonomy identifiers). Wang et al. leverage syntactic parse to assign NCBI Taxonomy identifiers to gene mention in biomedical literature [Wan+10c]. This is of particular advantage in species non-specific tasks like BioCreative III.

disease

UTU is a NER for recognizing and normalizing *disease* and symptom mentions in electronic medical records [Kae+14]. Interestingly, it includes word2vec-based vector representations [Mik+13] to solve the normalization task. DNorm is another machine learning-based system for disease name identification and normalization [Lea+13].

concentrations

Some NLP models exist to identify *concentrations* (or more generically, measure entities). For example, [Wan+09b] allows to extract pharmacokinetics numerical data from PubMed abstracts.

It must be noted that no robust NER for neuron cell types or for development stage recognition could be found (whereas a rule-based NER should be able to cover most of the expressions of developmental stages).

[15] http://cbioc.eas.asu.edu/banner/

[16] Open-Source Chemistry Analysis Routines, https://bitbucket.org/wwmm/oscar4/wiki/Home

2.4 Information Extraction

One way to improve manual literature search is to make use of automated information extraction (IE) methods. IE aims at extracting structured information from unstructured text. For example, in the case of brain region connectivity, it facilitates the manual search of connectivity data by analyzing very large numbers of scientific articles and proposing to the neuroscientist a list of brain regions potentially connected (see Section 4.2).

methodology for IE systems

The *methodology for IE systems* starts with a targeted text preprocessing, aimed at refining raw text by either removing noise (e.g. stop words) or enhancing signal (e.g. by adding part-of-speech, chunking or syntactic parsing). The next steps involve the identification of target entities (e.g. using NERs described above), the identification of potential events[17], and the identification of the relevant events (out of all possible events). Each individual step of an IE systems can be implemented with machine learning, rule-based or hybrid methods.

Typical IE tasks include protein-protein interaction (PPI) and drug-drug interaction. State-of-the-art techniques for PPI rely on multi-class SVM [Bjo+11]. Other techniques build on simplified syntactic parses [JG10].

system simplification

Recent efforts in IE systems have been directed towards *simplifying* the systems while maintaining high performance. For example, Bui et al. [Bui+13; BS11] presented a simple biomedical IE system for the BioNLP 2013 event extraction task [Kim+13]. Their system relies on simple syntactic patterns and consists of a learning phase during which a dictionary and patterns are automatically generated from annotated events. During the second phase (extraction), the said dictionary and patterns are applied to infer events from new, unseen text. The system delivered the best performance on strict matching and the third best on approximate matching (F-scores of .48 and .50, respectively).

rule-based IE

Similarily, Kilicoglu et al. [KB09] designed a *rule-based* methodology for event extraction, leveraging dependency parse representations. They reached the 2nd place on the BioNLP event shared-task [Kim+12]. One possible reason for the good performance of the rule-based approach is that it is in general not as much aggressive as ML approaches in optimizing against training data.

unsupervised approach

Alternatively, Vlachos et al. [Vla+09] present an almost *unsupervised approach* for biomedical event extraction based on the output of a syntactic parser and standard linguistic processing, augmented by rules acquired from the annotated development corpus. The system is designed to be as domain-independent and unsupervised as possible, requiring only a dictionary of verbs and a set of argument extraction rules. Their approach achieves high precision at the cost of a relatively low recall.

Cormack et al. present an IE system based on Linguamatics' commercial platform I2E[18] [Cor+15]. Their system uses a data-driven rule-based model and a simple supervised classifier. Evalua-

[17] In this context, an *event* is a set of two or more entities that have a specific relationship between them. For example, an event can consist of one protein that phosphorylates another protein.

[18] `http://www.linguamatics.com/products-services/about-i2e`

tion on the test data of the i2b2/UTHealth 2014 challenge[19] yielded an F-Score of 0.91, one percent behind the top performing system.

Miwa and Ananiadou recently extended the EventMine event extraction system to alleviate the need for task-specific tuning, in particular the (hyper-)parameters of machine learning algorithms. This is achieved by integrating and combining a weighting method and a covariate shift method on the training and test instances [MA15][20].

remove task-specific tuning

Another trend in IE are *ensemble methods*, that is: the combination of the output from several system. For example, the U-Compare event meta-service integrates nine event extraction systems [Kon+13; Kan+11b]. Its performance achieved by the ensemble significantly improves over individual systems.

ensemble methods

One other trend in IE is the application to *very large corpora*. For example, BioContext is an open-source event extraction system integrating and extending a number of tools performing NER and event extraction. It has been applied to 10.9 million PubMed abstracts and 234,000 open-access full-text articles from PubMed Central, yielding over 290,000 distinct genes/proteins mentions [Ger+12]. Alternatively, Bjorne et al. [Bjö+10] developed an event detection system that has been deployed at PubMed scale and EvidenceFinder has been deployed at Europe PubMed Central to search over 40M sentences about genes, proteins, diseases and metabolites[21].

very large corpora

2.5 Agile Methodologies

One methodology to support the development of agile text mining applications (ATMA) is *active learning*. The central idea behind active learning is that a machine learning algorithm can achieve higher performance with fewer training labels if it is allowed to select the data from which it learns. [Set10] gives a thorough review of available active learning methods and approaches. Also, various approaches that combine active learning and semi-supervised learning have been presented at the ICML 2011 Workshop[22]. Of particular interest is DUALIST[23], an interactive interface for active learning. DUALIST uses a multiclass naive Bayes classifier, and currently works for document classification only [Set11]. Another interesting idea is generalized expectation (GE) [Dru11], to reduce annotation time by shifting from traditional instance-labeling to feature-labeling.

active learning

Several *software frameworks* support the development of ATMAs. For example, *NLTK* [Bir06] is a popular open-source NLP framework in Python that is extensively documented and comes packaged with a large amount of datasets and pre-compiled models, allowing for rapid experimentation and exploration. NLTK is rather a library than an TMA but could easily be extended by text miners to provide a custom interface for domain experts. *Orange* [Dem+13] is another text mining framework allowing visual programming and Python scripting. However, a limited number of components for text mining are currently available [Xan14] and it is not clear how large-scale analysis can be performed with it. The General Architecture for Text Engineering (*GATE*) [Cun+11] is another popular and freely available TMA. GATE

NLTK

Orange

GATE

[19] https://www.i2b2.org/NLP/HeartDisease/
[20] http://www.nactem.ac.uk/EventMine/
[21] http://labs.europepmc.org/docs/EvidenceFinder.pdf
[22] https://sites.google.com/site/comblearn/
[23] https://code.google.com/p/dualist/

contains an integrated development environment and includes a high-level domain specific language (DSL), JAPE (Java Annotation Patterns Engine) for finite state transductions over annotations based on regular expressions. JAPE allows building complex rule-based TMA systems. Another popular open source TMA is the Unstructured Information Management Architecture (UIMA) [FL04], a flexible and extensible TMA focusing on interoperability of components and scalability. *Ruta*, an imperative rule language (DSL), was written to extend UIMA and enable rapid development of TMAs [Klu+14a; Klu+09]. Ruta comes with a workbench that greatly improves productivity, including testing and semi-automatic rule generation. However, its installation and operation is quite complex for domain experts, it does not easily support multiuser development and requires custom manual deployment to scale on a very large datasets. There is also a wide array of remote *web services* available to perform text mining. However, these are usually focused on a few narrow tasks, and it is often not possible to rely on them for very large analysis due to bandwidth requirements and costs. IBM *Watson* is a commercial cloud-based TMA framework based on UIMA [Fer+13; Fer+10]. Watson is specialized in open question answering and offers several different services like concept expansion, relationship extraction and classification[24]. *Argo*[25] [Rak+12], is a web-based workbench for composing and running TMA . It facilitates the development of custom workflows from a selection of elementary analytics and accommodates users with various areas and levels of expertise

Ruta

web services

Watson

Argo

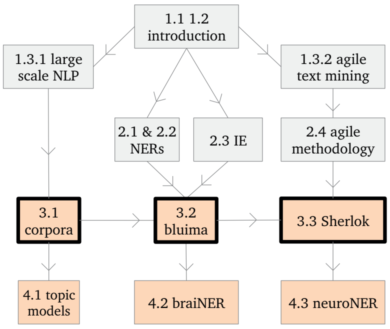
[24] http://www.ibm.com/smarterplanet/us/en/ibmwatson/

[25] `http://argo.nactem.ac.uk/`

3 Achievements and Tools

corpora

Chapter 3 presents the tools developed during this thesis. The first section (3.1) describes the *corpora* that will be used in subsequent experiments, including their preprocessing. This section is important for two reasons. First, this thesis is set to focus on very large corpora and acquiring such is not a light task. Second, as the old adage goes, *garbage in, garbage out*, thus preprocessing is an essential step to produce high-quality results.



Reader's guide (Section 1.4 p. 12).

bluima

The second section (3.2) introduces *bluima*, a natural language processing (NLP) pipeline focusing on the extraction of neuroscientific content. bluima started as an effort to develop a high performance NLP toolkit for neuroscience. In particular, focus was set on extracting entities that are specific to neuroscience (like brain regions) and that are not yet covered by existing text processing systems.

Sherlok

The last section (3.3) introduces *Sherlok*, an NLP system supporting the development of agile text mining applications (ATMA). In this aspect, Sherlok is focused on improving the collaboration between subject matter experts and text miners, a central goal of *agile text mining* (as presented earlier in Section 1.3.2).

leverage existing methods

All tools in this chapter try to leverage existing methods, models and libraries and embrace as much well-established standards as possible (e.g. OBO[1], RDF[2], UIMA[3], REST [FT02]). Except the corpora that could not be published because of copyrights, all tools are *open-sourced* with the goal to offer a simple way to replicate the experiments in this thesis.

open-source

3.1 Corpora and Preprocessing

This section presents the different corpora created during this thesis. A short description of the PubMed database is followed by a presentation of the pdf preprocessing toolchain. The NLP preprocessing steps are reviewed and the different corpora created during this thesis are described.

PubMed

PubMed[4] is a search engine to MEDLINE, a large online database of publications in the biomedical domain. An important property of PubMed is that most if not all significant publications are indexed in PubMed, making it the centralized repository of choice for

[1] obofoundry.org

[2] http://www.w3.org/RDF/

[3] http://docs.oasis-open.org/uima/v1.0/uima-v1.0.html

[4] http://www.ncbi.nlm.nih.gov/pubmed/

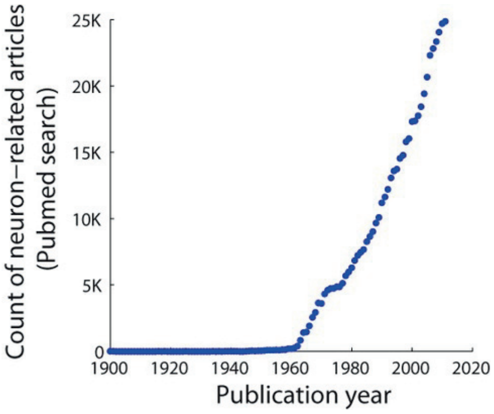


Fig. 3.1: Count of articles in PubMed containing the term `"neuron"` in their titles or abstracts. Approximately 25,000 articles were published in 2013.

biomedical text mining. As of August 2015, PubMed contained over 25 million records of published articles, growing at over 1 million citations a year. Over 25,000 articles per year are published that contain the term "neuron" in their titles or abstracts (see Figure 3.1). Abstracts are available for approximately 60% of the records. Licenses to text mine XML versions of these abstracts can be obtained from PubMed. In addition to abstracts, many PubMed records contain links to full-text articles, some of which are freely available in PubMed Central[5].

3.1.1 Comparison of abstracts and full-text articles

We present below some statistical comparison about the differences between PubMed abstracts and full-text articles. The advantage of PubMed abstracts is that they are available in large quantities and that an abstract captures the essential semantics of that article. On the other hand, full-text articles represent a very important data source, as they potentially contain an order of magnitude more information than abstracts [Kos10].

In terms of raw text length, PubMed abstracts contain on average 996 characters (146 tokens) while full-text articles contain on average 61,251 characters (11,500 tokens)[6]. Full-text articles include several sections that are typically not relevant for text mining (e.g. acknowledgments, funding, references) and ought to be filtered out (see Section 3.1.2). In terms of the quantity of relevant information that could be extracted, we found during our experiments significantly more relevant information in full-text papers than in abstracts. For example in experiments on brain region connectivity (Section 4.2), full-text papers contained on average 6.4 times more connections of brain region mentions than abstracts . In another experiment, over 12 times more information related to neocortical layers was found in a corpus of full-text papers, compared to PubMed abstracts (see Table 3.1). These quantitative results justify the additional efforts required to acquire and preprocess very large corpora in

[5] `http://www.ncbi.nlm.nih.gov/pmc/`
[6] See Section 3.1.4 for details about the corpora.

Tab. 3.1: Comparison of the amount of information available in PubMed abstracts and full-text papers, evaluated for different textual queries related to neocortical layers. A corpus of 630,216 full-text articles contained on average over 12 times as much mentions about layers than a corpus of 13 million PubMed abstracts (see 3.1.4 for details about the corpora).

Query	Full-text	PubMed abstracts	Ratio
layer I	12086	1021	11.84
layer 1	5000	510	9.80
layer II	17548	1451	12.09
layer II/III	5434	441	12.32
layer 2	9512	946	10.05
layer 2/3	6315	476	13.27
layer III	12893	1094	11.79
layer II/III	5434	441	12.32
layer 3	4039	446	9.06
layer IV	20432	1502	13.60
layer 4	11193	740	15.13
layer V	23963	1912	12.53
layer 5	9350	755	12.38
layer VI	8275	695	11.91
layer 6	5368	329	16.32
TOTAL	156842	12759	12.29

order to perform large-scale NLP (see Section 1.3.1). In the following section, we discuss the qualitative aspects of full-text articles.

3.1.2 Content Extraction from PDF Scientific Articles

Most scientific full-text articles are only available in PDF format[7], which is essentially a display format and has no concept of word boundaries or semantics. Additionally, even if PDF has been formalized in a specification, the numerous programs generating PDFs output slightly different kinds of documents, which makes it non-trivial to extract information from them. Hence the need for reliable tools to extract text from PDFs. This section describes an NLP processing pipeline that relies on a commercial PDF extraction library, followed by a set of content preprocessing measures that are devised to correct errors and improve content quality:

PDF parser

A PDF parser is a software library receiving as input a PDF file and returning its textual content. Most advanced PDF parsers already perform preprocessing steps. Several open source PDF libraries were considered. A qualitative evaluation was performed on a corpus of 8 PDFs and wherever necessary on a larger sample of 427 PDFs. The following list describes the evaluated libraries [8]:

- *Rossinante Web Service*[9] is a web-service by Xerox converting PDFs to XML. The service recognizes the structure of the document (e.g. headers/footers, page number, table of content, image caption, footnote) and delivers high-quality extracted text. However, the web-service latency was very high (on average over 30s) so it was decided not to consider that service.
- *LayoutAwarePDF* [Ram+12] performs rule-based PDF content extraction, enabling users to define their own layout rules in order to extract the relevant text of a PDF. Because journals have relatively stable layouts, it is possible to define a set of rules per journal to guide the extraction. Although this technique seems very promising, it was not used due to the large variety of journals considered. Indeed, by mapping the layout of 50 different neuroscience-related journals, only 45% of target PDFs articles would be covered and to reach 75% coverage would require the mapping of 290 journals.
- *Grobid* uses machine-learning to extract bibliographical information from PDFs. This feature could be useful to extract references from an article. Full-text extraction capabilities were limited and lacked precision.
- *PDFTextStream*[10] is a Java library exposing a hierarchical structure of a PDF article through abstractions like pages, blocks, lines and text-units (representing a single character). Eventually, *PdfTextStream* was selected for its superior performance in word splitting, handling of document encodings[11], and built-in aggregation of text.

[7] As opposed to machine-readable formats like XML, RDF or NXML.

[8] The following libraries were considered, but not further evaluated: JPedal, Apache PDF Box, PDF Clown, iText and TET.

[9] `http://open.xerox.com/Services/RossinanteWS`

[10] `http://snowtide.com`; commercial but free to use in single-threaded applications.

[11] Good support for embedded font and exotic character encoding.

- *pdftotext*[12] is a command line tool belonging to the Xpdf software suit. It performed relatively well on the evaluation corpus (paragraph order was respected and most ligatures were converted; however, accents were not correctly extracted). pdftotext was not further considered since it was not cross-platform and offered no significant advantages over PdfTextStream.

PDF preprocessing[13]

The following PDF preprocessing steps were developed to improve the quality of the text output from the selected PdfTextStream PDF parser:

- Removing non-informative footers and headers, based on a heuristics of text position and content (Levenhstein edit distance).
- Glyph mapping correction: In some fonts, the character that is displayed in the PDF does not correspond to the encoded one. For example, Greek letters like λ will be extracted as `l`, or = (equal sign) will be encoded as `1/4` (Table 3.2). To solve this important issue of incoherence between encoded and displayed character, the only possible solution is to know the correct font point for each font. Fortunately, the project pdf2svg[14] contains mapping for the most common fonts describing the correct mapping.
- Ligatures are characters composed of two or more characters merged together for typographic reasons. When extracted from PDF documents, they remain as one single character and must be corrected. Regular expressions have been developed to remove ligatures and replace them with the proper group of characters (see Table 3.3).
- Hyphenated words are words separated by hyphens to accommodate the text layout. In scientific articles, the text is usually justified and layed out in narrow columns, resulting in a lot of words being hyphenated. Hyphens can not be removed blindly, as they are sometimes part of a word (e.g. `blind-folded`) and sometimes added during the PDF document generation (see Table 3.4). A rule-based algorithm was developed to merge hyphens, based on 6 negative rules: W1 (the first word) or W2 (the second word) contains only one character; last character of W1 is a number; first character of W2 is a number; last character of W1 is a Greek letter; first character of W2 is a Greek letter. Only if none of these rules are satisfied will the word be dehyphenated.
- Scientific articles frequently use abbreviations throughout the text. These are identified using an existing machine learning model [MAC12] and then expanded throughout the article to their long form. The model has a reported performance of 98% precision and 93% recall on a standard data set
- Paragraph-Level Filtering is performed to remove non-informative paragraphs like acknowledgments or contact section using manually created rules.
- References: most scientific articles include bibliographical references citing related works. These references bring lots of noise to the NLP analysis, through proper names, journal names, and numbers. Simple rules (regular expressions) based on paragraph title do not perform well (only 83% accurate) and thus a supervised machine learning model (maximum entropy) was developed. The training corpus consisting of 1467

[12] `http://linux.die.net/man/1/pdftotext`

[13] Parts of this section have been published by a students that I co-supervised during my PhD, see [Rol13].

[14] https://bitbucket.org/petermr/pdf2svg

Encoded character	Correct character	Font	PMID
¼ (U+00BC)	= (U+003D)	AdvP4C4E74	16988649
≥ (U+003E)	4 (U+0034)	AdvPi1	16988649
1 (U+0031)	+ (U+002B)	Universal-GreekwithMathPi	10634775
2 (U+0032)	- (U+002D)	Universal-GreekwithMathPi	10634775
3 (U+0033)	= (U+003D)	Universal-GreekwithMathPi	10634775
4 (U+0034)	± (U+00B1)	Universal-GreekwithMathPi	10634775
ª (U+00AA)	© (U+00A9)	AdvPSSym	16962970
2 (U+0032)	- (U+002D)	AdvP7DED	16962970
3 (U+0033)	= (U+003D)	AdvP7DED	16962970
6 (U+0036)	± (U+00B1)	AdvP7DED	16962970

Tab. 3.2: Examples of incoherent glyph/encoding correspondence. For example on the 4th line, without glyph mapping correction, "=" (equal sign) would be incorrectly displayed as "1/4".

Unicode name	Codepoint	UTF-8	In large sample
LATIN SMALL LIGATURE FF	U+FB00	ef ac 80	yes
LATIN SMALL LIGATURE FI	U+FB01	ef ac 81	yes
LATIN SMALL LIGATURE FL	U+FB02	ef ac 82	yes
LATIN SMALL LIGATURE FFI	U+FB03	ef ac 83	yes
LATIN SMALL LIGATURE FFL	U+FB04	ef ac 84	yes
LATIN SMALL LIGATURE LONG S T	U+FB05	ef ac 85	no
LATIN SMALL LIGATURE STF	U+FB06	ef ac 86	no
LATIN SMALL LETTER AE	U+00E6	c3 a6	yes
LATIN SMALL LIGATURE IJ	U+0133	c4 b3	no
LATIN SMALL LIGATURE OE	U+0153	c5 93	yes

Tab. 3.3: List of supported ligatures. Right column states whether the ligature was present in a large sample of PubMed documents.

positive and 1502 negative examples was generated automatically by selecting the text following a "Reference" chapter (positive example), or by selecting a paragraph at random (negative example, verified manually). The machine learning model contains 21 features (see Table 3.5). Selected features capture the structural pattern of a reference, not the lexical structure. Model performance is 0.97 F-score, using 30 independent repetitions of 10-fold cross validation.

3.1.3 Pre-processing for PubMed

The different PubMed corpora (see 3.1.4) were preprocessed according to the following scheme. Tokenization was performed using the OpenNLP-wrappers developed by JulieLab for sentence segmentation, word tokenization and part-of-speech tagging [Tom+06] were used and updated to uimaFIT. Stopword removal: a few frequent tokens are removed which carry little meaning for our application, such as prepositions and conjunctions as well as punctuation marks. This list was created manually and is not very extensive. Lemmatization

W1	W2	Wrong	Correct	PMID
endolysin-β-	galactosidase	endolysin-βgal...	endolysin-β-gal...	21810267
(TC-	344B	(TC344B	(TC-344B	21810267
70-	Hz	70Hz	70-Hz	10634775
OHIP-	NL	OHIPNL	OHIP-NL	18405359
Self-	reported	Selfreported	Self-reported	18405359
(OHIP-	E)	(OHIPE)	(OHIP-E)	18405359

Tab. 3.4: Example of wrong dehyphenations. W1 and W2 are two words to be hyphenated.

```
YEARS
// 1978
years "19[56789]\\d|20[01]\\d"
// 1978b
years_abcd "19[56789]\\d[abcd]|20[01]\\d[abcd]"
// (2010)
years_parenthesis "\\((19[56789]\\d|20[01]\\d)\\)"

VOLUMES, PAGES
// 385-420
volume "\\d+ ?[--] ?\\d+"
// Comp. Neurol. 167: 385-420
volume_more "\\d+: ?\\d{1,4} ?[--] ?\\d{1,4}"
// pages
pages "p.? \\d+ [--] \\\\d+"

AUTHOR
// Gurdjian, E. S.
author1 "[A-Z]\\w+, [A-Z]\\."
// Beckstead RM (1979)
author2 "[A-Z]\\w+ [A-Z][A-Z ,]"
// Newman, R., and S. S. Winans
author3 ", and [A-Z]\\. [A-Z]"
// repetitions: Boussaoud D, Ungerleider LC, Desimone R
author4 "(, [A-Z]\\w+ [A-Z]{1,2}){2,}"
// , {comma, name}
author5 ", [A-Z]\\w+ [A-Z]"
// 4 Brodmann, K., V
// 17. Sorensen OW,
author6 "\\d{1,2}\\.? [A-Z]\\w+,? [A-Z]"
// S. Araki, Y. Tamori, M. Kawanishi, H. Shinoda, J. Masugi....
author6 "(([A-Z]\\.)?[A-Z]\\. [A-Z]\\w+, ){2,10}"
// Diesmann, M., and Morrison, A.
// Ferster, D., and Spruston, N.
author7 "((and )?[A-Z]\\w+, ([A-Z]\\. ?){1,2},? ?){2,6}"

MISCELANEOUS
proceedings "Proceedings of"

NEGATIVE EXAMPLES
// (Beckstead RM <-- parenthesis!
neg_author_parenthesis "\\([A-Z]\\w+ [A-Z][A-Z ,]"
// Gurdjian, E <-- parenthesis!
neg_author_parenthesis2 "\\([A-Z]\\w+, [A-Z]"
// ng (Rosenmund et al., 1998; Smith and Howe, 2000), a
neg_inline_ref "\\([A-Z]\\w+.{3,40}\\d+\\)"

neg_figure "^Fig(ure)?\\.? \\d+.*"
neg_table "^Tab(le)?\\.? \\d+.*"
neg_copyright "[Cc]opyright.{1,10}\\d{4}"
```

Tab. 3.5: Regular expressions to identify bibliographical references. To prevent model overfitting, these features only capture the structural pattern of a reference, not the lexical structure.

Corpus	Raw Text Size	Documents	$\lvert\xi\rvert$ (Raw)	$\lvert\xi\rvert$	$\lvert V\rvert$ (raw)	$\lvert V\rvert$
PubMed Abstracts	11.2GB	13,293,649	2×10^9	1.1×10^9	10.8×10^6	310,000
PubMed Neuroscience	$\sim$ 50GB	630,216	4×10^9	2.3×10^9	3.8×10^7	266,000
PubMed 100K	$\sim$ 1GB	100,000	57×10^6	1×10^9	1.1×10^6	125,000

Tab. 3.6: Statistics of the used corpora. The number of documents refers to non-empty documents after pre-processing. The columns $\lvert\xi\rvert$ (number of tokens) and $\lvert V\rvert$ (size of vocabulary) refer to the pre-processed corpus.

is performed by the domain-specific tool BioLemmatizer [Liu+12]. BioLemmatizer relies on a lexicon, together with rules that generalize morphological transformations to handle out-of-vocabulary words. BioLemmatizer achieves an lemmatization accuracy of 99% on a sampled set of the CRAFT corpus [Bad+12]. Abbreviation recognition (the task of identifying abbreviations in text) is performed by BIOADI [Kuo+09]. See Section 2.2 for further description of the above methods.

3.1.4 PubMed Corpora

Table 3.6 summarizes statistics of the generated corpora. The first corpus consists of all PubMed article containing an abstract (13.2 million in total as of November 2014). The second corpus contains 630,216 full-text articles focused on neuroscience. It was generated by aggregating articles from the personal libraries of all researchers in our research institute. This process was facilitated by the massive collaborative use of Zotero[15]. In addition, full-text papers containing mentions of brain regions were collected from the PubMed Central Open Access Subset and from open access journals related to neuroscience. Full-texts PDFs were subsequently processed using the pipelines described in Section 3.1.2 above.

The third corpus is an evaluation corpus consisting of a subset of 100,000 PubMed abstracts related to neuroscience. The selection was done by randomly selecting abstracts containing one of the following MeSH[16] terms: “Electrophysiology”, “Models, Neurological”, “Nervous System”, “Nervous System Diseases”, “Nervous System Malformations”, “Nervous System Neoplasms”, “Nervous System Physiological Phenomena”, “Neurons”, “Neurons, Afferent”, “Neurons, Efferent”, “Parkinson Disease”, “Parkinson Disease, Postencephalitic”, “Parkinson Disease, Secondary”, “Post-Synaptic Density”.

Unfortunately, the corpora presented in this chapter cannot be openly redistributed, as this is not allowed in the PubMed license. However, all preprocessing tools to generate these corpora from PubMed-leased abstracts are publicly available at https://github.com/BlueBrain/bluima under an open source license (see Section 3.2 for details).

[15] www.zotero.org

[16] See Section 4.1.4 for description of MeSH.

3.2 bluima: a UIMA-based NLP Toolkit for Neuroscience[17]

This section describes *bluima*, a natural language processing (NLP) pipeline focusing on the extraction of neuroscientific content and based on the UIMA framework [FL04][18]. bluima builds upon models from biomedical NLP (BioNLP) like specialized tokenizers and lemmatizers. It adds further models and tools specific to neuroscience (e.g. named entity recognizer for neuron or brain region mentions) and provides collection readers for neuroscientific corpora. The resulting code and models are publicly available at `https://github.com/BlueBrain/bluima`. Three novel UIMA components are proposed: the first allows configuring and instantiating UIMA pipelines using a simple scripting language, enabling non-UIMA experts to design and run UIMA pipelines. The second component is a common analysis structure (CAS) store based on MongoDB, to perform incremental annotation of large document corpora. The third component extracts and normalizes complex scientific measures like *17.3 millimole/l.* from scientific article

3.2.1 Introduction

bluima started as an effort to develop a high performance natural language processing (NLP) toolkit for neuroscience. In particular, focus was set on extracting entities that are specific to neuroscience (like brain regions and neurons) and that are not yet covered by existing text processing systems.

After careful evaluation of different NLP frameworks, the UIMA software system was selected for its open standards, its performance and stability, and its usage in several other biomedical NLP (bioNLP) projects; e.g. JulieLab [Hah+08], ClearTK [Ogr+08], DKPRo [DCG09], cTAKES [Sav+10], ccp-nlp, U-Compare [Kon+13], SciKnowMine [Ram+10], Argo [Rak+12]. Initial development went fast and several existing bioNLP models and UIMA components could rapidly be reused or integrated into UIMA without the need to modify its core system, as presented in Section 3.2.2.

Once the initial components were in place, an experimentation phase started where different pipelines were created, each with different components and parameters. Pipeline definition in verbose XML was greatly improved by the use of uimaFIT [OB09] (to define pipelines in compact Java code) but ended up being problematic, as it requires some Java knowledge and recompilation for each component or parameter change. To allow for a more agile prototyping, especially by non-specialist end users, a pipeline scripting language was created. It is described in Section 3.2.2.

Another concern was incremental annotation of large document corpus. For example, the ability to run an initial pre-processing pipeline on several millions of documents and annotate them again at a later time. The initial strategy was to store the documents on disk, and overwrite them every time they would be incrementally annotated. Eventually, a CAS store module was developed to provide a stable and scalable strategy for incremental annotation, as described in Section 3.2.2. Finally, Section 3.2.3 presents two case studies illustrating

[17] A version of this chapter has been published as [Ric+13]

[18] UIMA stands for unstructured information management applications and is freely available at `http://uima.apache.org/`

Name	Source	Scope	# forms
Age	HBP	age of organism, developmental stage	138
Sex	HBP	sex (male, female) and variants	10
Method	HBP	experimental methods in neuroscience	43
Organism	HBP	organisms used in neuroscience	121
Cell	HBP	cell, sub-cell and region	862
Ion channel	Channelpedia [Ran+11]	ion channels	868
Uniprot	Uniprot [Bai+05]	genes and proteins	143,757
Biolexicon	Biolexicon [Tho+11]	unified lexicon of biomedical terms	2.2 Mio
Verbs	Biolexicon	verbs extracted from the Biolexicon	5,038
Cell ontology	OBO [Bar+05]	cell types (prokaryotic to mammalian)	3,564
Disease ont.	OBO [Osb+09]	human disease ontology	24,613
Protein ont.	OBO [Nat+11]	protein-related entities	29,198
Brain region	Neuronames [BD03]	hierarchy of brain regions	8,211
Wordnet	Wordnet [Fel10]	general English	155,287
NIFSTD	NIF [Ima+11; Bug+08]	neuroscience ontology	16,896

Tab. 3.7: Lexica and ontologies used for lexical matching.

the scripting language and evaluating the performance of the CAS store against existing serialization formats.

3.2.2 bluima Components

bluima contains several UIMA modules to read neuroscientific corpora, perform preprocessing, create simple configuration files to run pipelines, and persist documents on the disk.

UIMA Modules

typesystem

bluima's typesystem builds upon the typesystem from JulieLab [Hah+07], which was chosen for its strong biomedical orientation and its clean architecture. bluima's typesystem adds neuroscientific annotations, like *CellType*, *BrainRegion*, etc.

collection readers

bluima includes several collection readers for selected neuroscience corpora, like PubMed XML dumps, PubMed Central NXML files, the BioNLP 2011 GENIA Event Extraction corpus [Pyy+12], the Biocreative2 annotated corpus [Kra+08], the GENIA annotated corpus [Kim+03], and the WhiteText brain regions corpus [Fre+09].

PDF reader

A PDF reader was developed to provide robust and precise text extraction from scientific articles in PDF format. The PDF reader is described in detail in Section 3.1.2 performs content correction and cleanup, like dehyphenation, removal of ligatures, glyph mapping correction, table detection, and removal of non-informative footers and headers.

Preprocessing is performed as described in Section 3.1.3.

lexical-based NERs

bluima uses UIMA's ConceptMapper [Tan+10] to build lexical-based NERs based on several

Tool	Advantages	Disadvantages
UIMA GUI	GUI	minimalistic UI, can not reuse pipelines
XML descriptor	typed (schema)	very verbose
raw UIMA java API	typed	verbose, requires writing and compiling Java
uimaFIT	compact, typed	requires writing and compiling Java code

Tab. 3.8: Different approaches to writing and running UIMA pipelines.

neuroscientific lexica and ontologies (Table 3.7). These lexica and ontologies were either developed in-house or were imported from existing sources. bluima wraps several machine learning-based NERs, like OSCAR4 [Jes+11] (chemicals, reactions), Linnaeus [Ger+10] (species), BANNER [LG+08] (genes and proteins), and Gimli [Cam+13] (proteins). See Section 2.3 for an overview of biomedical NERs.

machine learning-based NERs

protein NER

Additionally, a protein NER was developed with the goal to reproduce state of the art results like BANNER and tightly integrate with other UIMA components. The NER was developed with the ClearTK [Ogr+08] framework, that allows to reuse existing UIMA components, while handling a lot of common operations like cross-validation and evaluation. It provides a common interface and wrappers for popular machine learning libraries, so that one can change models without changing much of the application code. ClearTK provides some common feature extractors (like e.g. lowercase, hyphen, character N-gram), and allows writing more specific ones. The NER uses a conditional random field model implemented with the Mallet library. It was trained on the BioCreative2 corpus [Wil+07], containing 20,000 annotated sentences. The model has been evaluated using 10-fold cross-validation, and achieves a competitive 80% F-score. The highest F-score for the BioCreative2 shared task was 87%, but the system has not been published [And07].

Pipeline Scripting Language

There are several approaches[19] to write and run UIMA pipelines (see Table 3.8). All bluima components were initially written in Java with the uimaFIT library, that allows for compact code. To improve the design and experimentation with UIMA pipelines, and enable researchers without Java or UIMA knowledge to easily design and run such pipelines, a minimalistic scripting (domain-specific) language was developed, allowing UIMA pipelines to be configured with text files, in a human-readable format (Table 3.9). A *pipeline script* begins with the definition of a collection reader (starting with `cr:`), followed by several annotation engines (starting with `ae:`)[20]. Parameter specification starts with a space, followed by the parameter name, a column and its value. The scripting language also supports embedding of inline Python and Java code, reuse of a portion of a pipeline with `include` statements, and variable substitution similar to shell scripts. Extensive documentation (in particular snippets of scripts) is automatically generated for all components, using the JavaDoc and the uimaFIT annotations. Eventually, the pipeline script language has been superseded by the Ruta scripting language (see 3.3) that allows writing rules for information extraction.

[19] Other interesting solutions exist (e.g. IBM LanguageWare, Argo), but are not open source.

[20] If not package namespace is specified, bluima loads Readers and Annotator classes from the default namespace.

```
# collection reader configured with a list of files (provided as external params)
cr: FromFilelistReader
 inputFile: $1
# processes the content of the PDFs
ae: ch.epfl.bbp.uima.pdf.cr.PdfCollectionAnnotator

# tokenization and lematization
ae: SentenceAnnotator
 modelFile: $ROOT/modules/julielab_opennlp/models/sentence/PennBio.bin.gz
ae: TokenAnnotator
 modelFile: $ROOT/modules/julielab_opennlp/models/token/Genia.bin.gz
ae: BlueBioLemmatizer

# lexical NERs, instantiated with some helper java code
ae_java: ch.epfl.bbp.uima.LexicaHelper.getConceptMapper("/bbp_onto/brainregion")
ae_java: ch.epfl.bbp.uima.LexicaHelper.getConceptMapper("/bams/bams")

# removes duplicate annotations and extracts collocated brainregion annotations
ae: DeduplicatorAnnotator
 annotationClass: ch.epfl.bbp.uima.types.BrainRegionDictTerm
ae: ExtractBrainregionsCoocurrences
 outputDirectory: $2
```

Tab. 3.9: Example of pipeline script for the extraction of brain regions mention co-occurrences from PDF documents.

CAS Store

A CAS store was developed to persist annotated documents, resume their processing and add new annotations to them. This CAS store was motivated by the common use case of repetitively and incrementally processing the same documents with different UIMA pipelines, where some pipeline steps are duplicated among the runs. For example, when performing resource-intensive operations (like extracting the text from full-text PDF articles, or performing syntactic parsing), one might want to perform these preliminary operations once, store these results, and subsequently perform different experiments with different UIMA modules and parameters. The CAS store thus allows to perform the preprocessing only once, to then persist the annotated documents, and to perform the various experiments in parallel.

MongoDB[21] was selected as the datastore backend. MongoDB is a scalable, high-performance, open-source, schema-free (NoSQL), document-oriented database. No schema is required on the database side, since the UIMA typesystem acts as a schema, and data is validated on-the-fly by the module. Every CAS is stored as a MongoDB document, along with its annotations. UIMA annotations and their features are explicitly mapped to MongoDB fields, using a simple and declarative language. For example, a `Protein` annotation is mapped to a `prot` field in MongoDB. The mappings are used when persisting and loading from the database. As of this writing, annotations are declared in Java source files. In future versions, we plan to store mappings directly in MongoDB to improve flexibility. Persistence of complex typesystem has not been implemented yet, but could be easily added in the future.

Currently, the following UIMA components are available for the CAS store:

[21] http://www.mongodb.org/

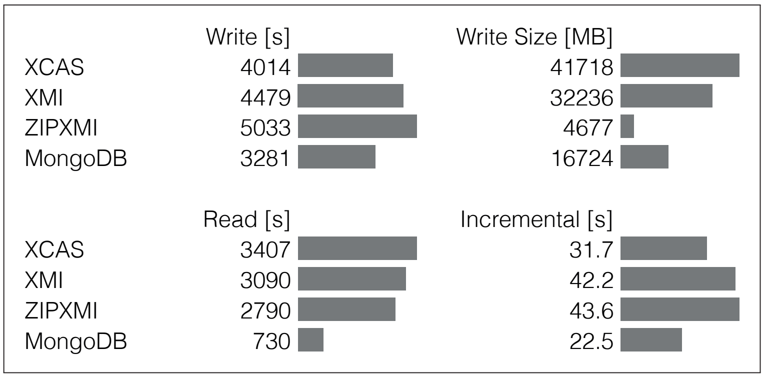


Fig. 3.2: Performance evaluation of MongoDB CAS Store against 3 other serialization formats.

- *MongoCollectionReader* reads CAS from a MongoDB collection. Optionally, a (filter) query can be specified, e.g.
  - `{pmid:  17}` to query a specific PubMed document;
  - `{pmid:{$in:[12,17]}}` to query a list of PubMed documents;
  - `{pmid:{ $gt:  8, $lt:  11 }}` for a range of documents;
  - `{my_db_field:{exists:true}}` for the existence of a field.
- *RegexMongoCollectionReader* is similar to MongoCollectionReader but allows specifying a query with a regular expression on a specific field;
- *MongoWriter* persists new UIMA CASes into MongoDB documents;
- *MongoUpdateWriter* persists new annotations into an existing document;
- *MongoCollectionRemover* removes selected annotations in a MongoDB collection.

With the above components, it is possible within a single pipeline to read an existing collection of annotated documents, perform some further processing, add more annotations, and store theses annotations back into the same MongoDB documents.

3.2.3 Case Study and Evaluation

A first experiment to illustrate the scripting language was conducted on a large dataset of full-text biomedical articles. A second simulated experiment evaluates the performance of the MongoDB CAS store against existing serialization formats. Finally, we describe an experiment to extract measures and units from biomedical articles.

Scripting and Scale-Out

bluima was used to extract brain region mention co-occurrences from scientific articles in PDF. The pipeline script (Table 3.9) was created and tested on a development laptop. Scale-out was performed on a 12-node (144-core) cluster managed by SLURM (Simple Linux Utility for Resource Management). The 383,795 PDFs were partitioned in 767 jobs. Each job was instantiated with the same pipeline script, using different input and output parameters. The processing completed in 809 minutes ($\simeq 8$ PDF/s).

Tab. 3.10: Examples of scientific measures and their normalized form.

Raw Text	Normalized value	Normalized unit
41-55 nanomole	[41-55] $\cdot 10^{-9}$	M
5,5 mm per s	5.5 $\cdot 10^{-3}$	m/s
7.16 +/- 0.09 nm	[7.07-7.25] $\cdot 10^{-6}$	m
five to nine kilograms	[5-9]	kg

MongoDB CAS Store

The MongoDB CAS store (MCS) has been evaluated against 3 other available serialization formats (XCAS, XMI and ZIPXMI). For each, 3 settings were evaluated: writes (CASes are persisted to disk), reads (CASes are loaded from their persisted states), and incremental (CASes are first read from their persisted states, then further processed, and finally persisted again to disk). Writes and reads were performed on a random sample of 500,000 PubMed abstracts and annotated with all available bluima NERs. Incremental annotation was performed on a random sample of 5,000 PubMed abstracts and incrementally annotated with the `Stopwords` annotator. Processing time and disk space was measured on a commodity laptop (4 cores, 8GB RAM).

In terms of speed, the MCS significantly outperforms the other formats, especially for reads (Figure 3.2). The MCS disk size is significantly smaller than XCAS and XMI formats, but almost 4 times larger than the compressed ZIPXMI format. The incremental annotation is significantly faster with MongoDB, and does not require duplicating or overwriting files, like with the other serialization formats. The MCS could be scaled up in a cluster setup, or using solid states drives (SSDs). Writes could probably be improved by turning MongoDB's "safe mode" option off. Furthermore, by adding indexes, the MCS can act as a searchable annotation database.

Large-scale extraction of scientific measures and units

An analytics pipeline was developed to extract measures from scientific articles. In our context, *measures* are defined as numerical values combined with units, e.g. *35 nM*, *two volts* or *102.3 ± 15 millimeters*. They characterize experimental results or experimental conditions. The former are part of the *results* section of a paper, while the later are part of the *materials and methods* section. Extracting and normalizing them is valuable for scientists [PM09], e.g. to incorporate them as model parameters in a brain simulation. Table 3.10 illustrates several examples of measures found in scientific articles, together with the normalized form. The normalized form is composed by a normalized value (where unit prefixes like *milli* have been removed and factored in the value) and the unit normalized to the International System of Units (SI). Measures are first identified using UIMA's RegexAnnotator[22] with a complex, yet compact and readable set of regular expressions. A wide array of units is supported, in particular SI prefixes, SI units and derived units. In addition, written numbers as well as exotic units[23] found in scientific papers are supported. Once measures are identified, they are normalized in order to facilitate the comparison between them. Value normalization is performed with UIMA's RegexAnnotator. Units normalization relies on QUDT [@HK14],

[22] uima.apache.org/d/uima-addons-current/RegularExpressionAnnotator/

[23] E.g. *zeptomole* (10^{-21} mole) and *yoctoseconds* (10^{-24} seconds).

an ontology created at NASA to handle units and measures. QUDT allows to determine whether two quantities are commensurate, and if so, how to convert from one system to another. For example, it is possible to convert *250 metric tons* into *grams*, or *0.1 nanomolar* into *moles per cubic meter*. QUDT was extended to improve its coverage for biomedical units. The measure NER was evaluated against a corpus of 500 full-text articles chosen at random from the PMC open subset. Tokens that contained a digit, but were not covered by the extractor were manually evaluated. Most of these were molecules or proteins. In total, the corpus contained 12,975 measures. 650 measures could not be extracted, resulting in a recall of 94.99%. The text mining system was deployed on 1,066,885 full-text articles from the PubMed Central Open Access Subset[24] and on over 23 million abstracts from the PubMed database. 859,428,656 mentions of numerical values were identified, out of which 172,199,175 mentions contained a unit.

[24] http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

3.3 Agile text mining with Sherlok

The successful development of an intelligent text mining application requires the collaboration of two main stakeholders: subject matter experts and text miners. In order to improve that collaboration, we introduced earlier a new methodology, *agile text mining* (Section 1.3.2). Agile text mining is characterized by short development cycles, frequent tasks redefinition and continuous performance monitoring through integration tests.

agile text mining

This section introduces *Sherlok*, a system supporting the development of agile text mining applications (ATMA). The resulting code is publicly available at http://sherlok.io

Sherlok

Sherlok is designed to support the lightweight development of ATMA by facilitating the collaboration between domain experts and text miners. In Sherlok, each ATMA is modeled as an analysis *pipeline*. A Sherlok pipeline contains the following components: tests, annotation engines, resources and scripts (see Table 3.11). It specifies all steps to perform a text mining analysis (e.g. split words, remove determinants, annotate locations, etc.). For example, one pipeline might perform text preprocessing followed by named entity recognition (NER) of people and places, while another pipeline might extract mentions of neurons in scientific text. A pipeline is expected to contain several *tests* consisting of some sample text and the expected analysis output. Tests enable continuous monitoring of a pipeline's validity and ensure that subsequent development of a pipeline does not break previous progress. *Annotation engines* are UIMA-based components that perform a single text analysis step in a pipeline, for example part-of-speech tagging or sentiment analysis. Annotation engines are a way to conveniently encapsulate complex models (e.g. a machine learning model) into independent, self-contained components. They can be separately configured for each pipeline (e.g. by specifying a different tag-set for part-of-speech) and shared across pipelines.

pipeline

tests

Annotation engines

All pipeline components are connected together using a high-level scripting language. Sherlok uses the *Ruta scripting language* [Klu+14a; Klu+09] to orchestrate the different annotation engines, resources and perform rule-based transformations. The Ruta language and its matching paradigm enable the rapid specification of comprehensible rules for knowledge extraction. They allow compact representation, while still providing a high level of expressiveness. Ruta rules are declared in plain text files and are composed of conditions and actions. For example, the rule `"(red|blue)" -> Color;` contains a condition (the presence of the text "red" or "blue") triggering an action (the creation of a `Color` annotation). The

Ruta scripting language

Tab. 3.11: The different components of a Sherlok ATMA

Component	Requirements	Format
tests	simple to write, continuous testing	JSON
annotation engine	composable, scalable, versionable	UIMA
local resource	editable	txt, obo
remote resource	versionable	git, http
Ruta scripts	expressive, compact, readable, extensible, scalable, versionable	Ruta

Fig. 3.3: Example script to illustrate the syntax of the Ruta language. The script matches entities representing units (using an ontology), proteins (using a machine-learning model) and floating-point numbers (using a regular expression). Finally, it identifies simple constructs of protein concentrations.

```
// declares an annotation for units
// (e.g. ’millivolts’ or ’moles per liter’)
DECLARE Unit(STRING iso);
// Matches unit instances from an ontology
ONTO("$units/units_ontology.obo", Unit, "iso")};

// apply a machine-learning model for protein NER
ENGINE ners.Proteins;
EXEC(Proteins);

// match simple instances of floating-point numbers
DECLARE RealNumber;
"[-+]?[0-9]*.?[0-9]+" -> RealNumber;

// create annotations for protein concentrations
RealNumber Unit "of"? Protein {
    -> MARK(ProteinConcentration, 1, 4)
```

exemplary script in Figure 3.3 allows matching simple occurrences of protein concentrations like “0.5 g of GST” or “1.5 mg/ml bovine serum albumin”.

local and remote resources

Sherlok provides an efficient system to manage *local and remote resources*. Most TMA depend on various resources like code libraries, parameters of machine learning models or ontologies. It is essential that these resources are decoupled from the text mining solution itself. They are often large in size and not necessarily edited and updated by the same person that develops and maintains the TMA. Thus, they require careful and efficient management. Sherlok transparently exposes local resources stored on disk through its RESTful API [FT02], so that they can be uploaded, edited and deleted. Remote resources can be seamlessly integrated in a pipeline by specifying their remote location, using various protocols like “http” and “git”. Sherlok will download and maintain a local copy that can be synchronized with the remote resource if necessary. For example, a pipeline can be configured to include a remote dictionary of brain regions from a git repository. The first time the pipeline is loaded, Sherlok will download the remote dictionary and cache it locally. If the dictionary is edited on the remote repository, the local copy can be flushed, and the updated dictionary will be downloaded afresh the next time the pipeline is used.

OBO ontology format

Sherlok natively supports the *OBO ontology format*[25] and introduces an enhanced version called ROBO to facilitate the creation of synonym-rich ontologies. The OBO format is compatible with other ontology formats like OWL and is a lightweight format to specify ontologies. The OBO format attempts to achieve the following goals: human readability, ease of parsing, extensibility and minimal redundancy. A large amount of ontologies for the biomedical domain are already available in the OBO format through the OBO foundry[26], a suite of orthogonal interoperable reference ontologies. Many ontologies and taxonomies are publicly available. However, most of these ontologies were designed to structure and organize a specific domain of interest, not to serve as resources for TMA. Typically,

[25] OBO format specification at http://owlcollab.github.io/oboformat/doc/obo-syntax.html

[26] http://obofoundry.org/

Tab. 3.12: Comparison of OBO (left) and ROBO (right) ontology formats for the same entry. In this example, the ROBO format allows to compactly define the 6 variants of "layer 2" using a regular expression `layer[ -](II|ii|2)`, instead of having to list them all.

OBO format	ROBO format

`[Term]`
`id: LAYER:001`
`name: layer 2`
`synonym: "layer-2"`
`synonym: "layer 2"`
`synonym: "layer-II"`
`synonym: "layer II"`
`synonym: "layer-ii"`
`synonym: "layer ii"`
	`[Term]`
	`id: LAYER:001`
	`name: layer 2`
	`rsynonym: "layer[ -](II|ii|2)"`

they lack appropriate synonyms, resulting in low recall. To facilitate the management of synonyms in an ontology, Sherlok supports an enhanced version of the OBO format called *ROBO* (for regular-expression OBO)[27]. ROBO allows specifying synonyms through compact regular expressions, thus improving the expressiveness and compactness of the ontology. For example, all 6 synonyms for the "layer two" of the neocortex ("layer-2", "layer 2", "layer-II", "layer II", "layer-ii", "layer ii") can be defined using the regular expression `layer[ -](II|ii|2)` (see Table 3.12).

ROBO

Sherlok allows frictionless horizontal *scale out*. For local development and testing, it can be setup on a single local server with minimal requirements (a Java runtime engine). Sherlok come with several examples and tutorial pipeline to get started with. Once a pipeline has been successfully developed and tested on a small evaluation corpus, Sherlok can be deployed in a distributed setup where one Sherlok instance acts as the "master" node and all other deployed Sherlok instances ("slaves") pull their configurations (pipelines, engines, resources) from the master instance (see Fig. 3.4). The workload is distributed on the slaves through a load balancer. Slaves are dynamically added or removed, depending on the workload. This deployment scheme enables Sherlok to seamlessly and transparently run text mining at virtually any scale. The exact same system is used for local small-scale development and distributed scale out.

scale out

To visualize and analyze results, Sherlok can be integrated with *Elasticsearch*[28] by enhancing its index with semantic information[29]. That is, text mining is applied on every document being indexed and on every incoming search query. The extracted semantic information can then be used to improve the relevance of search results. Sherlok seamlessly integrates with Elasticsearch, and is exposed as a custom analyzer. From the point of view of the Elasticsearch developer, integration boils down to installing the Sherlok plugin on every Elasticsearch node, and configuring the Sherlok analyzer. This configuration is done in Elasticsearch and includes the specification of the pipeline script to be used, and the mapping between the information extracted by Sherlok and the field stored in Elasticsearch.

Elasticsearch integration

[27] https://github.com/sherlok/ruta-ontologies

[28] Elasticsearch is a popular distributed, open source search engine, designed for horizontal scalability and reliability

[29] `https://github.com/sherlok/sherlastic`

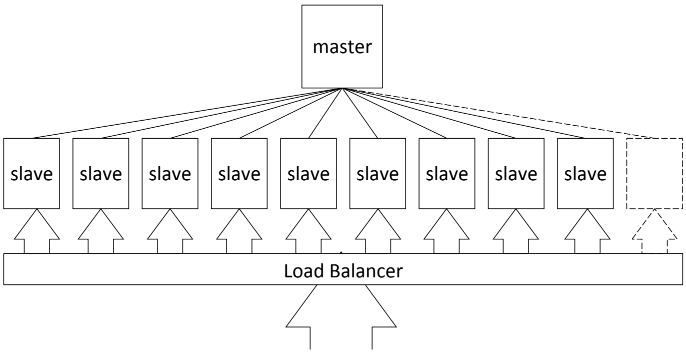
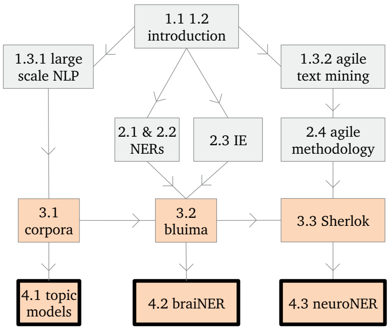


Fig. 3.4: Deployment architecture for a Sherlok-based ATMA. Multiple slave Sherlok instances depend on a master instance that holds all configuration. Slave instances can be horizontally scaled to dynamically accommodate the workload.

4 Experiments and Stories

4.1 topic models

Chapter 4 presents the experiments conducted during this thesis. The first section is dedicated to *topic models* (4.1), a type of unsupervised machine learning models for discovering the hidden thematic structure in document collections. Because they are *unsupervised*, they do not require annotated corpora that are often difficult and costly to acquire. We evaluate several existing libraries for large-scale topic modeling (4.1.2) and train a model on a large corpus (4.1.3). We then measure the correlation between the unsupervisingly learned topics and the manually created MeSH descriptors (4.1.4). Finally, we leverage topic models to generate semantic profiles of several BBP researchers through the articles they read (4.1.5).



Reader's guide (Section 1.4 p. 12).

4.2 brain connectivity extraction

The second section (4.2) present text-mining (TM) models to extract and aggregate *brain regions connectivity* results from a large corpus of 8 billion words. Models are evaluated against in-vivo connectivity data from the Allen Brain Atlas (ABA) with an estimated precision of 78%. The resulting database contains over 4 million brain region mentions, and over 100,000 potential brain region connections. We then evaluate these TM models to automatically suggest targets from the literature for tractography studies. We perform an extensive manual review of the literature to identify the projections of three selected brain structures and compare it with the TM results. We run probabilistic tractography on one structure (nucleus accumbens) and compare the output with the TM suggestions and the literature review. Overall, TM models find three times as many targets as two man-weeks of curation. The overall efficiency of the TM against manual literature review in our study is 98% recall (at 36% precision), meaning that over all the targets for the three selected seeds, only one target has been missed by TM.

4.3 extraction of neurons and their properties

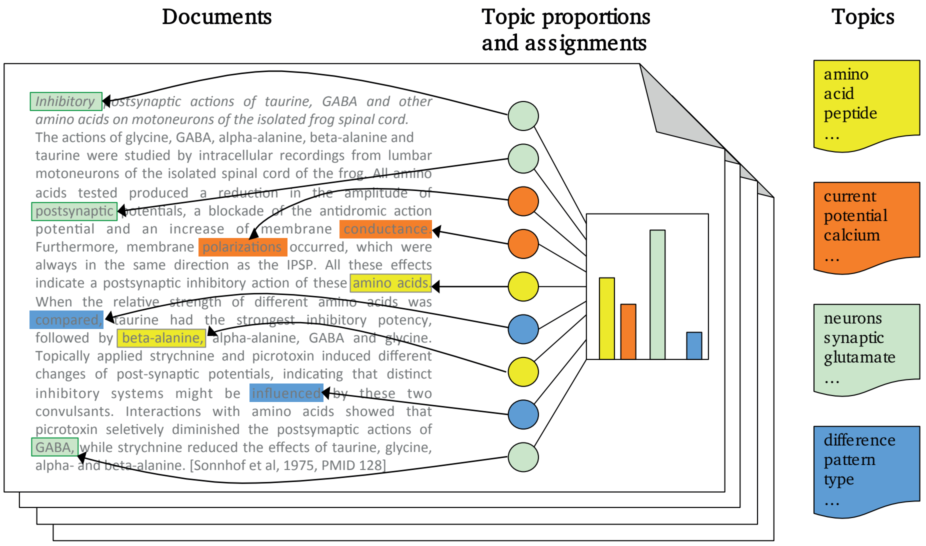
The last section (4.3) introduces neuroNER, an NLP model to perform automated identification and normalization of neuron type mentions in the neuroscientific literature. The model proceeds by decomposing a neuron mention into its specific compositional features. For example, the mention “thalamic CALB1-expressing neurons” is decomposed into two properties: location (“thalamus”) and the genes expressed (“Calbindin”). This decomposition allows comparing neurons at a more semantic level, e.g. the mention “calbindin D-28k-positive neurons in the reticular nucleus of the thalamus” is equivalent to the previous example, since “calbindin D-28k” can be considered a synonym for Calbindin and the “reticular nucleus of the thalamus” is a subregion of the thalamus. This kind of decomposition and normalization is essential for cross-laboratory studies, since neuroscience currently lacks consistent terminologies or nomenclatures for describing neuron types. To demonstrate the utility of our approach, we also apply our method towards cross-comparing the NeuroLex and Human Brain Project (BBP) cell type ontologies.

4.1 Topic models[1]

> *Words do not have meanings – meanings have words.*
>
> — **Geoffrey Williams, on Saussure's theory** [Wil03]

Imagine it would be possible to search for documents based on their underlying themes. Instead of searching for documents by keywords only, it would be possible to first define the theme that we are interested in, and then find the documents that are related to this theme [Ble12]. Such are the goals of topic modeling. *Topic models* are unsupervised Bayesian models for discovering the hidden thematic structure in document collections. They enable semantic clustering and semantic classification of large document collections. Given a large text corpus, a topic model is trained to learn the dominant themes present in that particular corpus. Because they are unsupervised, they do not require annotated corpora that are often difficult and costly to acquire.

topic models



Inhibitory postsynaptic actions of taurine, GABA and other amino acids on motoneurons of the isolated frog spinal cord.
The actions of glycine, GABA, alpha-alanine, beta-alanine and taurine were studied by intracellular recordings from lumbar motoneurons of the isolated spinal cord of the frog. All amino acids tested produced a reduction in the amplitude of postsynaptic potentials, a blockade of the antidromic action potential and an increase of membrane conductance. Furthermore, membrane polarizations occurred, which were always in the same direction as the IPSP. All these effects indicate a postsynaptic inhibitory action of these amino acids. When the relative strength of different amino acids was compared, taurine had the strongest inhibitory potency, followed by beta-alanine, alpha-alanine, GABA and glycine. Topically applied strychnine and picrotoxin induced different changes of post-synaptic potentials, indicating that distinct inhibitory systems might be influenced by these two convulsants. Interactions with amino acids showed that picrotoxin seletively diminished the postsymaptic actions of GABA, while strychnine reduced the effects of taurine, glycine, alpha- and beta-alanine. [Sonnhof et al, 1975, PMID 128]

Fig. 4.1: The intuitions behind latent Dirichlet allocation, illustrated on a single PubMed abstract with PMID 128 [Son+75]; Figure adapted from [Ble12]). Four "topics" are illustrated, each representing distributions over words (far right). Each document is a mixture of corpus-wide topics. Each word is drawn from one of the topics. In this example, the third topic (green) is has a higher probability in this document (histogram on the right). The first word of the document, "inhibitory", is drawn from that green topic. As can be seen, the four topics that are predominant in that abstract accurately summarize the main themes of that abstract. See Section 4.1.1 for an in-depth discussion of these results.

Figure 4.1 illustrates the application of latent Dirichlet allocation (LDA), a kind of topic

illustration

[1] Parts of this section have been published by students that I co-supervised during my PhD. In particular, subsections 4.1.1, 4.1.2 and 4.1.3 have been published in part in [Zim13] and subsection 4.1.4 in part in [Cob14].

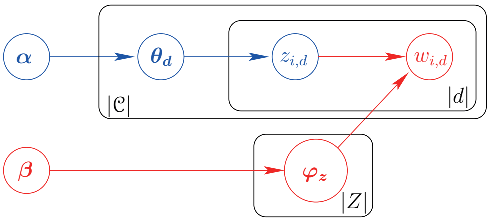


Fig. 4.2: LDA as a graphical model [Cha13]. The circles with $z_{i,d}$ and $w_{i,d}$ denote the topic and the term respectively of the i word in document d.

model, on a single PubMed abstract with PMID 128 [Son+75]. A topic model is trained on a large corpus of PubMed abstracts, with the constraint to contain 200 topics. The trained topic model is then applied on the aforementioned PubMed abstract. Each document is modeled as a mixture of corpus-wide topics. Figure 4.1, right presents four topics that were predominant in the chosen illustrative abstract. The first topic (yellow) is related to amino acids and peptides, while the third topic (green) is related to neurons, synapses and neurotransmitters. In the model, each word from the abstract is drawn from one of the topics. For example, the neurotransmitter "GABA" (bottom left) belongs to the third topic (green). This example illustrates the ability of topic model to grasp the underlying themes of a document.

applications

Topic models enable topic analysis of document collections [GS04]. Other purposes might include feature extraction to improve NER in scientific papers [LW09] or automatic construction of taxonomies [Bak+12a; Wan+10a].

In this section, we briefly introduce topic models (4.1.1). Since our goal is to train a topic model on very large corpora, we then evaluate several existing libraries for topic modeling (4.1.2) and attempt to train a model on a very large corpus of over 600,000 full-text articles related to neuroscience (4.1.3). As a further experiment, we leverage the MeSH (medical subject headings) information from PubMed articles and compute a correlation measure between the unsupervisingly learned topics and the manually created MeSH descriptors (4.1.4). We finally apply topic modeling for generating semantic profiles of several BBP researchers through the articles they read (4.1.5).

4.1.1 Latent Dirichlet Allocation (LDA)

This paragraph gives a very brief introduction to topic models, more specifically Latent Dirichlet Allocation (LDA, [Ble+03]). For a detailed description see [Cha13].

topic

A *topic* is a probability distribution on some vocabulary (set of words). Topic models are generative probabilistic models describing a random process creating documents, that is, for each new word in the document, we choose a topic z and then choose the word w according to a probability distribution of the topic φ_z. There are different ways of choosing a topic and modeling φ_z. In the case of LDA we choose for each document a probability distribution θ_z according to a Dirichlet distribution (prior) parametrized by a *hyper-parameter* $\vec{\alpha}$. Similarly, for a given topic, the distribution of terms are chosen according to a Dirichlet distribution parametrized by the hyper-parameter $\vec{\beta}$. Figure 4.2 summarizes LDA as a graphical model.

Topic 40		Topic 159		Topic 160		Topic 167		Topic 200	
nerve	-2.7	amino	-3.1	current	-3.8	neurons	-3.8	differences	-3.5
spinal	-3.0	acid	-3.4	potential	-4.0	synaptic	-3.9	two	-3.9
cord	-3.3	peptide	-3.5	Ca2+	-4.0	glutamate	-4.0	patterns	-3.9
nerves	-4.1	peptides	-3.6	membrane	-4.2	receptors	-4.1	pattern	-4.1
peripheral	-4.2	acids	-3.9	calcium	-4.4	receptor	-4.1	types	-4.3
sensory	-4.3	activity	-4.3	mV	-4.4	GABA	-4.2	each	-4.3
motor	-4.4	residues	-4.3	currents	-4.5	NMDA	-4.3	distribution	-4.4
injury	-4.4	protease	-4.4	mM	-4.6	hippocampal	-4.3	similar	-4.5
dorsal	-4.6	protein	-4.5	action	-4.6	neuronal	-4.3	three	-4.6
axons	-4.7	enzyme	-4.6	K+	-4.6	acid	-4.6	found	-4.6

Tab. 4.1: The 10 most frequent terms for the most probable topics in the abstract PMID 128 [Son+75], along with their importance ($\log p(w|z)$). The first significant topic (40) consists of terms like *nerve*, *spinal* and *cord* that accurately captures the methods of this article. Topic 167 is about neuromodulation and contains several neurotransmitters (glutamate, GABA, NMDA) and terms related to this theme.

model training

Training an LDA model on a corpus means to estimate φ_z for all topics giving rise to a word-by-topics matrix Φ such as to maximize the likelihood of the entire model. Although the models being mathematically relatively simple, it is a major challenge to accurately estimate its parameters, since many involved quantities, typically marginal probabilities, are intractable to compute exactly, thus approximations have to be made. There exist two basic approaches how to estimate parameters for this type of models. One is based on Gibbs sampling, which is in the realm of Monte Carlo methods. The other approach known as Variational Bayes is a formulation of the training as optimization problem. It is also possible to estimate the hyper-parameters from the training corpus, which is implemented by a few implementations considered. An additional challenge is that, due to the huge targeted amount of data to be processed, implementations of estimation procedures need to be highly optimized in order to scale.

topic model example

We now conduct a more detailed evaluation of the PubMed abstract with PMID 128 [Son+75] from Figure 4.1. The LDA model was trained on all PubMed abstracts with 200 topics and 500 iterations of Gibbs sampling[2]. As can be seen in Figure 4.3 top, this abstract studies the inhibitory post-synaptic effect of several neuromodulators in the frog spinal cord. Figure 4.3 bottom shows the preprocessed document, annotated with the most probable topic number for each word as well as the original abstract. Figure 4.4 shows the inferred distribution of topics, where 5 distinct peaks are apparent[3]. The first peak corresponds to topic 40, whose most frequent terms are *nerve*, *spinal* and *cord*. This topic accurately captures the fact that experiments in this article were performed on the spinal cord. Topic 167 has the highest probability, and its most frequent terms are “neurons”, “synaptic” and “glutamate”. Table 4.1 gives the most frequent terms for the 5 most common topics inferred for this document. As one can see, the LDA model was able to capture the underlying semantic themes of the abstract.

[2] Training took 2 days and 10h with 10 threads on a single machine.

[3] A high topic probability means this topic is predominant in that document.

The actions of glycine, GABA, alpha-alanine, beta-alanine and taurine were studied by intracellular recordings from lumbar motoneurons of the isolated spinal cord of the frog. All amino acids tested produced a reduction in the amplitude of postsynaptic potentials, a blockade of the antidromic action potential and an increase of membrane conductance. Furthermore, membrane polarizations occurred, which were always in the same direction as the IPSP. All these effects indicate a postsynaptic inhibitory action of these amino acids. When the relative strength of different amino acids was compared, taurine had the strongest inhibitory potency, followed by beta-alanine, alpha-alanine, GABA and glycine. Topically applied strychnine and picrotoxin induced different changes of post-synaptic potentials, indicating that distinct inhibitory systems might be influenced by these two convulsants. Interactions with amino acids showed that picrotoxin seletively diminished the postsymaptic actions of GABA, while strychnine reduced the effects of taurine, glycine, alpha- and beta-alanine. But differences in the susceptibility of these amino acid actions to strychnine could be detected: the action of taurine was more sensitively blocked by strychnine compared with glycine, alpha- and beta-alanine. With regard to these results the importance of taurine and GABA as transmitters of postsynaptic inhibition on motoneurons in the spinal cord of the frog is discussed.

actions$_{167}$ glycine$_{167}$ beta-alanine$_{167}$ taurine$_{167}$ studied$_{160}$ intracellular$_{160}$ recordings$_{167}$ lumbar$_{40}$ motoneurons$_{40}$ isolated$_{160}$ spinal$_{40}$ cord$_{40}$ frog$_{160}$ amino$_{159}$ acids$_{159}$ tested$_{167}$ produced$_{160}$ reduction$_{152}$ amplitude$_{160}$ postsynaptic$_{167}$ potentials$_{160}$ blockade$_{167}$ antidromic$_{81}$ action$_{160}$ potential$_{160}$ increase$_{167}$ membrane$_{160}$ conductance$_{160}$ furthermore$_{159}$ membrane$_{160}$ polarizations$_{160}$ occurred$_{160}$ always$_{200}$ direction$_{160}$ effects$_{167}$ indicate$_{167}$ postsynaptic$_{167}$ inhibitory$_{167}$ action$_{160}$ amino$_{159}$ acids$_{159}$ relative$_{200}$ strength$_{167}$ amino$_{159}$ acids$_{159}$ was$_{200}$ compared$_{200}$ taurine$_{167}$ strongest$_{200}$ inhibitory$_{167}$ potency$_{159}$ followed$_{160}$ beta-alanine$_{167}$ glycine$_{167}$ topically$_{40}$ applied$_{160}$ strychnine$_{167}$ picrotoxin$_{167}$ induced$_{167}$ changes$_{152}$ post-synaptic$_{167}$ potentials$_{160}$ indicating$_{167}$ distinct$_{200}$ inhibitory$_{167}$ systems$_{167}$ might$_{167}$ be$_{200}$ influenced$_{200}$ two$_{200}$ convulsants$_{167}$ interactions$_{167}$ amino$_{159}$ acids$_{159}$ showed$_{200}$ picrotoxin$_{167}$ diminished$_{167}$ actions$_{167}$ strychnine$_{167}$ reduced$_{167}$ effects$_{167}$ taurine$_{167}$ glycine$_{167}$ alpha-$_{159}$ beta-alanine$_{167}$ differences$_{200}$ susceptibility$_{167}$ amino$_{159}$ acid$_{159}$ actions$_{167}$ strychnine$_{167}$ be$_{200}$ detected$_{159}$ action$_{160}$ taurine$_{167}$ was$_{200}$ sensitively$_{200}$ blocked$_{167}$ strychnine$_{167}$ compared$_{200}$ glycine$_{167}$ alpha-$_{159}$ beta-alanine$_{167}$ regard$_{200}$ results$_{167}$ importance$_{200}$ taurine$_{167}$ transmitters$_{167}$ postsynaptic$_{167}$ inhibition$_{167}$ motoneurons$_{40}$ spinal$_{40}$ cord$_{40}$ frog$_{160}$ discussed$_{200}$

Fig. 4.3: Top: Original abstract of PubMed abstract with PMID 128 [Son+75] whose title is "Inhibitory postsynaptic actions of taurine, GABA and other amino acids on motoneurons of the isolated frog spinal cord". Bottom: Pre-processed abstract annotated with the most likely topic number for each term.

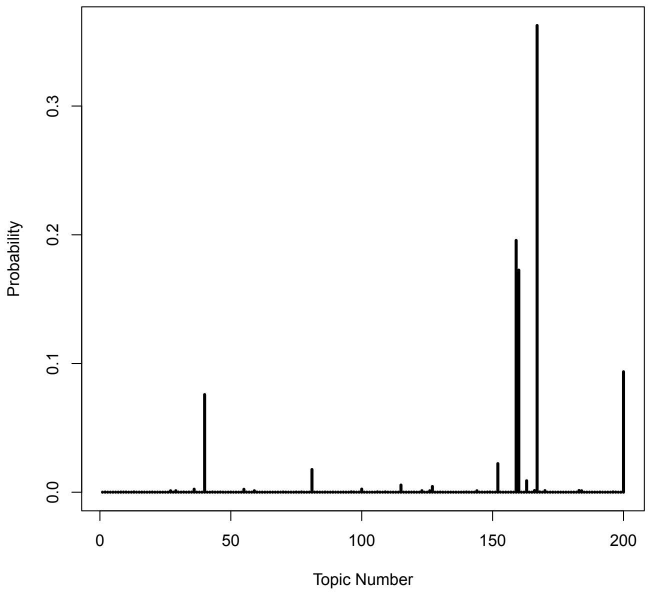


Fig. 4.4: Inferred topic distribution for abstract PMID 128 [Son+75]. The 5 most significant topics (probability larger than 0.05) are 40, 159, 160, 167 and 200. Topic 167 has the highest probability and its most frequent terms are *neurons*, *synaptic* and *glutamate*. See Table 4.1 for the most frequent terms of theses topics.

4.1.2 Evaluated LDA Libraries

evaluation criteria

Several existing implementations of LDA were evaluated. Supplementary table 4.2 shows a summary of the seven LDA implementations considered. From this list, DCA, Mallet, PLDA and Vowpal Wabbit were selected and thoroughly tested. Different types of benchmarks were performed, attempting to answer the following questions:

1. Accuracy: Do the implementations compute models of comparable quality with the same input data? In this first set, we fixed the hyper-parameters (since not all tested implementations support hyper-parameter estimation) and ran every software for a sufficiently long time, i.e. until the estimated likelihood printed by the software converges.
2. Quality: How well can each implementation do in *in principle* using the given training data? This time, we allowed for optimization of hyper-parameters and other options specific to each implementation to enhance the quality.
3. Efficiency: How fast do the implementations compute the models? We assess the approximate computing time needed until the model has converged.
4. Scalability: What are the speed-ups, if we add more processors?

evaluation

The evaluation was run on a set of approximately 100,000 PubMed abstracts (with 100 topics), preprocessed as described in section 3.1.3. We performed 10-fold cross-validation and calculated the held-out likelihood using the Mallet implementation[4]. The final score used is the median. If not stated otherwise, the tests were run on a cluster[5] with a single thread and a sufficient amount of RAM. After each termination of the training of a fold, we estimated the likelihood on the held-out part. We ran each configuration sufficiently long to have converged. Figure 4.5 summarizes the estimated likelihoods for each configuration tested on the PubMed corpus. We now summarize general observations for the selected implementations, also considering experiments on large corpora such as the complete PubMed abstracts corpus as well as on a subset of the PubMed Neuroscience full-text corpus (see Section 3.1 for corpus statistics). The evaluated implementations were the followings:

DCA

- *DCA* (Discrete Component Analysis) [Bun09a] was selected for evaluation since it is implemented in C along with some support for multi-threading and therefore promised to be efficient. Also, it is the only software implementing an unbiased estimation method for the held-out likelihood [Bun09c]. DCA was the most mature implementation considered as well as the most efficient in terms of resource usage. Especially the memory handling of DCA is extremely optimized and efficient.

Mallet

- *Mallet* [@McC02] is a popular software package in the NLP and machine learning community not only used for topic modeling but also for training classifiers and taggers. Mallet supports multi-threaded training of topic models and promised to be a valuable candidate. However, memory problems were discovered during evaluation with large corpora (with an order of magnitude of 10^9 tokens), as input data and intermediate results cannot be split in block and saved on disk during training.

PLDA

- We also evaluated *PLDA* [Wan+09a] as it was the only true parallel implementation of Gibbs sampling. However, PLDA has not been optimized for handling large models

[4] For a complete discussion about likelihood estimation of held-out data, see Chapter 5.2. of [Zim13].

[5] Intel Xeon X5690 2x6 cores @ 3.47GHz per node, 22GB of maximum usable RAM per node, operated by a Red Hat Linux 6.3

Name	Author	Version	Model(s)	Parameter Estimation	Likelihood Estimation	Language	Deployment	Reference
Mallet	McCallum et al.	2.0.7 (last change in Repo regarding topic models: Jan 2012, release with parallel topic models: 2008)	LDA (+ hyper-parameter optimization.)	Distributed Gibbs Sampling	Left-to-right particle sampler	Java	Multi-threaded, single-machine	[@McC02]
DCA	Wray Buntine	0.202 (released August 2009, first release February 2009)	LDA, Gamma Poisson Models, Pachinko Allocation (experimental) + hyper-parameter optimization.	Gibbs Sampling	Left-to-right sequential sampler	C	Multi-threaded, single-machine	[Bun09b], [Bun09a]
Online LDA	Matthew Hoffman	September 2010	LDA (no hyperparameter opt.)	Online VB	NA	Python	Single machine, online	[Hof+10]
Vowpal Wabbit	John Langford and Matthew Hoffman	Last relevant change: August 2012. First version: Jan 2012	LDA (no hyperparameter opt.)	Online VB	NA	C++	Single-machine, no parallelization of LDA	[@Hof], [Hof+10]
PLDA	Yi Wang et al.	Last changes: Jul 2011. First version 2008.	LDA (no hyper-parameter optimization)	Parallel Gibbs Sampling (MPI implementation of asynchronous LDA with optimizations)	NA	C++	Parallel deployment (MPI)	[Wan+09a]
Mr. LDA	Ke Zhai et al.	Last changes: August 2012, Development start: Jan 2012	LDA	Distributed VB	Lower-bound for likelihood given	Java	MapReduce (Hadoop)	[Zha+12]
Hadoop LDA	Xiance Si	Last changes: March 2012, Dev start: April 2010	LDA (no hyper opt) Parallel Gibbs Sampling	Gibbs Sampling	NA	Java	MapReduce (Hadoop)	[@Had]

Tab. 4.2: Overview of LDA implementations considered.

and seems to be more of a proof of concept for a parallel implementation of Gibbs sampling. Additionally, it performed poorly in terms of held-out likelihood (see Figure 4.5).

Vowpal Wabbit

- *Vowpal Wabbit* [@Hof] was selected because of its implementation of an online algorithm to train LDA models [Hof+10]. An online deployment has the advantage that if new documents arrive, they can directly be included into the model and it is not needed to retrain an entire model. The LDA training in Vowpal Wabbit is a re-implementation in C++ of the original Online LDA python scripts [Hof+10]. As can be deduced from the likelihood scores in Figure 4.5 the LDA implementation in Vowpal Wabbit performs systematically worse in terms of held-out likelihood, the source of this remaining unknown.

MapReduce implementations

- Implementations based on the MapReduce paradigm were also considered, i.e. Hadoop-based implementations. An implementation is *Mr. LDA* (Map Reduce LDA). Preliminary tests on a small corpus[6] showed that very large overhead is generated and training a topic model took significantly longer than other implementations. Similarly, *Hadoop LDA* [@Had] did not perform satisfactory. An experiment was run on PubMed abstracts with 100 topics. It took 29 hours to complete 200 iterations of Gibbs sampling on 12 machines which have 2 physical cores and 7.5 GB of RAM, which uses computing resources in order of magnitudes larger than other implementations. The two above implementations failed to deliver sufficient performance and evaluation was not carried on.

4.1.3 Large scale training on full-text PubMed Neuroscience Corpus

issues with full-text corpus

An LDA model was trained on the PubMed neuroscience full-text corpus (see Section 3.1.4). Difficulties were experienced as the learning scores printed by DCA behaved erratically and did not display the expected smooth convergence curve (Figure 4.6). A first hypothesis is that the features extracted from the full-texts are not specific enough to find clear topics. If we look at topics as co-occurrences of terms, then this means, that the relevant co-occurrences are getting lost in the noise of general English, i.e. most of the terms occur in all the documents and only relatively few terms appear in a subset of documents only. This implies that training is quickly finished and the learning score fluctuates around some value. However, this does not explain the strong and in some case very irregular fluctuations of the learning score as in Figure 4.6 right. Another hypothesis is an implementation error in the DCA option necessary to split data-structures created during training into blocks when using a very large corpus. An experiment on the PubMed abstracts corpus, where we explicitly activated that option did not reveal any problems[7].

success with abstracts corpus

On the other hand, training on the PubMed abstracts corpus works relatively well, i.e. the learning scores converge smoothly. This is due to the fact that abstracts are some sort of manual “feature selection” of the entire document where key terms are densely grouped together and are much less "polluted" by general terms.

[6] 20 Newsgroups corpus, containing 20’000 newsgroups messages chosen from 20 different newsgroups.

[7] However, this does not completely rule out the possibility of a bug in the software.

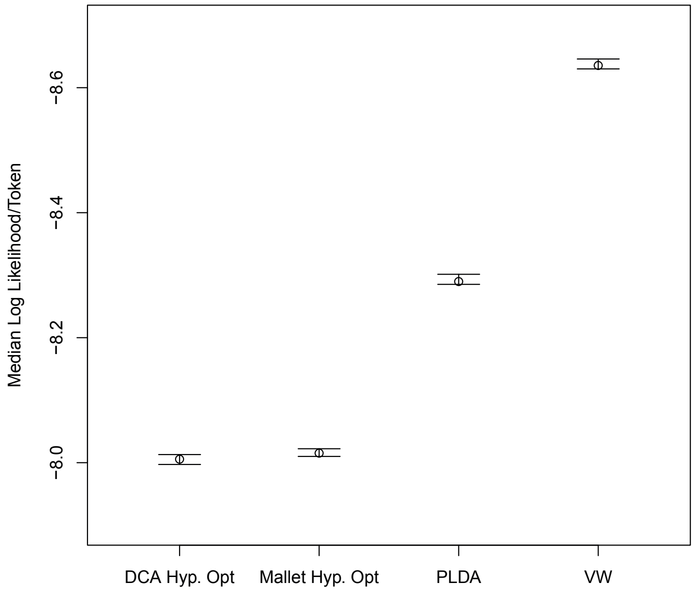


Fig. 4.5: Comparisons of held-out likelihoods (lower is better) on a subset of 100K abstracts from PubMed. This experiment assesses the *accuracy* of each implementation by evaluating model quality, given the same input data. Likelihoods were estimated using 10-fold cross-validation. The vertical bar represent a 95% confidence interval for the median. DCA and Mallet have similar performances but DCA seems to be slightly better. PLDA and Vowpal Wabbit (VW) behave worse as they do not optimize the hyper-parameters. Strikingly, Vowpal Wabbit performs even worse compared to PLDA with the *same* hyper-parameters.

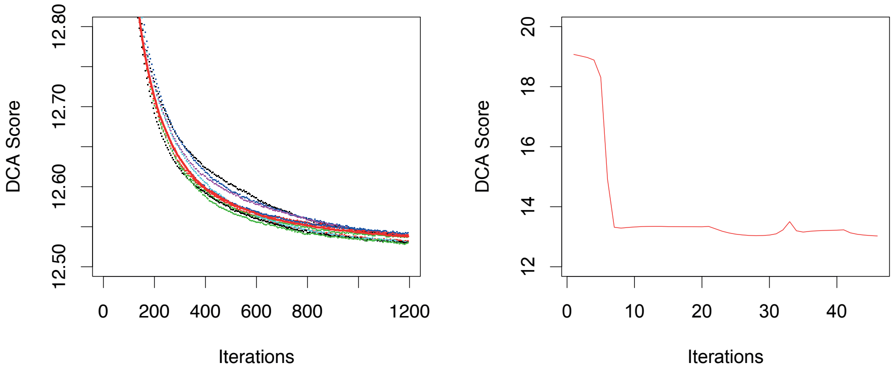


Fig. 4.6: Training convergence plots on PubMed abstracts (left) versus PubMed Neuroscience full-text corpus (right). Training LDA on the full-text corpus does not display the expected smooth convergence curve. Potential explanations are implementation errors or the fact that that features extracted from the full-texts are not specific enough to find clear topics.

This hypothesis could be tested by applying a refined preprocessing and thus improving the quality of the raw texts. For example, bibliographies, citations and perhaps also tables could be removed (see Section 3.1.2). However, it is not clear whether this would fundamentally change the results as many of the artifacts such as citations would get removed anyways by some form of frequency filtering. A simpler and probably more effective way to improve the LDA models would be to massively expand the stop word list with most frequent terms in the topic representing general terms. Many terms occurring in all documents are removed and terms with more discriminating power remain this way. Another approach could be to only extract chemical components and biological entities such as proteins, genes and neurotransmitters. Also there, a few entities appear in many of the documents, however, many entities should be very specific to some concept, i.e. the co-occurrence of a few key terms should be sufficient to characterize a research subject. Finally, only parts of a publication could be considered, such as abstracts, discussion sections and conclusions. This way, many less features are selected from a document, which leads to faster training.

Following the above experiments, it was decided not to further attempt the training of topic models on the full-text corpus and focus instead on exploiting the results from topic models trained on all PubMed abstract. With over 2 billion words, the PubMed abstracts corpus is by itself a substantial body of literature.

4.1.4 Correlation with Medical Subject Headings (MeSH)

Medical Subject Headings MeSH is a comprehensive medical vocabulary managed by the United States National Library of Medicine. It consists of descriptors arranged in a hierarchical structure and is primarily used for indexing journal articles and books in the life sciences. As any descriptor has a list of similar terms, it can also serve as a thesaurus to facilitate information retrieval. The MeSH vocabulary is continually revised and updated by the Medical Subject Headings Section staff from the Library of Medicine. We offer a short introduction to the MeSH structure and its use in the PubMed database in the appendix (Section 6.1).

In this section, we apply standard topic modeling to a corpus tightly related to neuroscience. We show that even such a small corpus (100,000 documents) can be used to train a topic model and deliver good results. We subsequently leverage the MeSH descriptors assigned to PubMed abstracts to calculate a correlation measure between topics and MeSH descriptors with promising results.

applications

The generated topic model and correlation matrix could be used for various practical tasks; we enlist some of them:

- Descriptor prediction for a document d could be performed by building a ranking system using $p(m|d)$.
- Prediction of major descriptors for a document d with a given set of descriptors $M(d)$ could be done in a similar fashion.
- New relations within the MeSH hierarchy could be retrieved using information measures such as the symmetric KL divergence between descriptors.

Some of these applications were researched by [New+09] on a smaller corpus.

Correlation

The probability of cooccurrence of a MeSH descriptor and a topic on the corpus is given by

$$p(m, z) = \sum_{d \in \mathcal{C}} p(m, z|d)p(d)$$

where d is a document, $\mathcal{C}$ is the corpus, z is a topic and m is a MeSH descriptor. We define $p(m|d) = 1$ when descriptor m is present in document d and 0 otherwise. Under the assumption that $z \perp\!\!\!\perp m \mid d$ and that $p(d) = \frac{1}{|\mathcal{C}|}$ we obtain:

$$p(m, z) = \frac{1}{|\mathcal{C}|} \sum_{d \in D(m)} p(z|d) \tag{4.1}$$

where $D(m)$ is the set of documents where m is present. We use this probability to generate a so-called *correlation matrix* between topics and MeSH descriptors.[8]

Results

The corpus used in this section is composed of 100,000 PubMed abstracts related to Neuroscience; every article has at least one MeSH descriptor under the category "Nervous System". The corpus was preprocessed using bluima (see Section 3.2).

correlation matrix

We generated a *correlation matrix* using all descriptors and all topics (see Figure 4.7). Rows (MeSH descriptors) and columns (topics) on the correlation matrix are reordered so as to reveal hidden structure in the data[9]. We choose to maximize the values on the diagonal of the matrix to set the first columns and rows. Once these are fixed, we continue maximizing the diagonal to set the remaining rows.

We can observe a clear line in the matrix diagonal indicating that several topic-descriptor pairs are highly correlated. Certain general topics have a high correlation with the majority of descriptors; this fact is illustrated in the matrices by dark vertical lines. Likewise, general descriptors such as "Brain" relate to a big proportion of topics, which is shown in our matrices by dark horizontal lines. A closer look reveals that the 10 most frequent MeSH terms in the corpus are indeed relatively unspecific: "Humans", "male", "female", "animals", "adult", "middle aged", "aged", "rats", "child", "adolescent". In most cases, these terms characterize experimental test subjects or animal models. They do not provide additional information about the content of a publication other than test subjects and are thus not very discriminating.

Less than half of all MeSH descriptors appear in our corpus (11,641 out of 27,149) and only 9,000 descriptors appear in at least 2 documents. Figure 4.8 shows the number of descriptors appearing in different number of documents (plotted from 1-100 document appearances). These results come from the fact that our corpus is relatively small and focused on Neuroscience. Thus, some MeSH descriptors are less likely to appear in the selected documents.

To verify that our results are semantically valid we perform a qualitative analysis similar to that performed when validating a topic model: we show for selected MeSH descriptors a

[8] The term *correlation* is used throughout this section to refer to $p(m, z)$ unless stated otherwise.
[9] This reordering is a research problem in itself generally referred as *seriation* or *sequencing*.

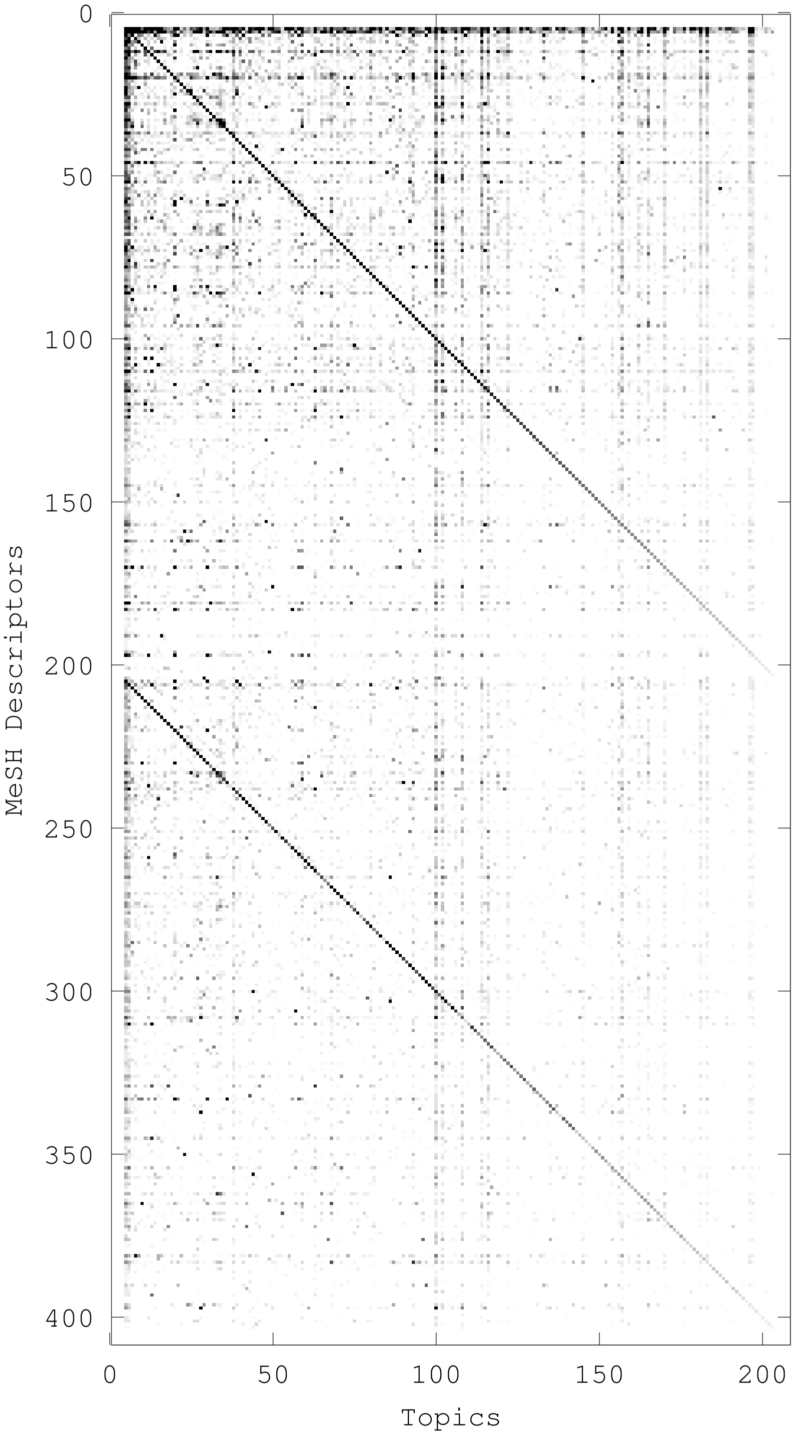


Fig. 4.7: Correlation matrices for the four different descriptor and topics combinations used to calculate $p(m, z)$ on the 100K corpus. Reordered as to maximize the diagonal values and cropped to match twice the number of topics (400)

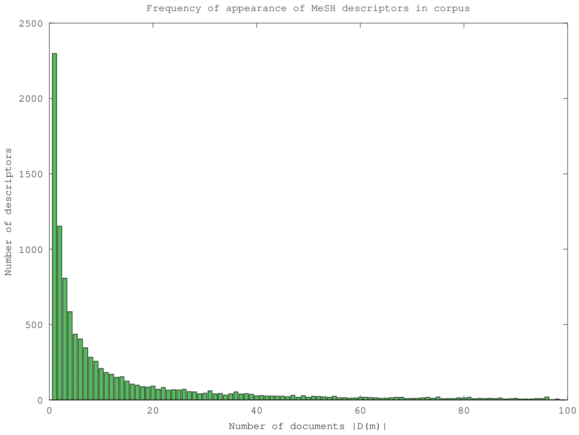


Fig. 4.8: Frequency of appearances of MeSH descriptors in 100K corpus. Each bar represents the number of descriptors that appear in x documents (in x axis).

Topic 144	**0.163**	**Topic 79**	**0.134**	**Topic 126**	**0.051**	**Topic 119**	**0.034**	**Topic 51**	**0.033**
disease	0.070	abeta	0.047	study	0.038	role	0.032	study	0.019
's	0.052	amyloid	0.042	result	0.023	function	0.022	brain	0.016
ad	0.045	ad	0.041	effect	0.021	mechanism	0.021	review	0.015
alzheimer	0.040	alzheimer	0.037	show	0.019	system	0.020	research	0.011
tau	0.032	disease	0.036	suggest	0.018	pathway	0.015	human	0.011
dementia	0.025	's	0.034	human	0.015	play	0.014	model	0.011
patient	0.018	plaque	0.021	change	0.014	neuronal	0.013	recent	0.010
alpha-synuclein	0.013	protein	0.020	present	0.014	involve	0.012	provide	0.010
body	0.012	peptide	0.020	increase	0.011	important	0.012	development	0.009
brain	0.011	app	0.019	previous	0.011	regulate	0.011	clinical	0.009

Tab. 4.3: Five highest related topics for descriptor "Alzheimer Disease" ordered by $p(z|m)$ (bold). For each topic the list of the ten most probable words along with $p(w|z)$ is presented.

list of topics to which it relates (ordered by $p(z|m)$) and the information for each topic. In Table 4.3 we present results for the 10^{th} most frequent descriptor in our corpus (appearing in 1,849 documents): "Alzheimer Disease". We see that the two highest correlated topics show a clear conceptual link to Alzheimer disease. The other three topics could be considered general topics (given our corpus) but are not completely unrelated to Alzheimer disease and have lower probabilities $p(z|m)$ compared to the first two.

In general, common descriptors and topics have higher correlation values when compared with less common descriptors and topics, which does not imply that we will consistently produce bad results when dealing with common descriptors. In fact, as long as the topic model generates only a small number of general topics and all descriptors appear relatively often in the corpus, results will be positive. Nonetheless, we will not be able to completely remove this effect given that our corpus contains common descriptors and our model produces general topics.

Table 4.4 presents the results for the $5,000^{th}$ most frequent descriptor in our corpus (appearing in 7 documents): "Neuropilin-1", a protein involved in vessel formation and axonal guidance. As expected descriptors become more specific as their frequency declines but even though "Neuropilin-1" only appears in seven documents it should still offer good results.

Topic 10	0.357	Topic 30	0.077	Topic 3	0.074	Topic 38	0.059	Topic 119	0.049
axon	0.046	expression	0.027	endothelial	0.067	nerve	0.092	role	0.032
growth	0.030	neural	0.023	vascular	0.046	regeneration	0.031	function	0.022
neurite	0.028	gene	0.021	vessel	0.031	axon	0.029	mechanism	0.021
outgrowth	0.019	development	0.019	cell	0.029	injury	0.023	system	0.020
adhesion	0.017	embryo	0.016	vegf	0.023	axonal	0.020	pathway	0.015
cone	0.016	cell	0.012	capillary	0.019	peripheral	0.017	play	0.014
molecule	0.016	express	0.011	brain	0.019	sciatic	0.017	neuronal	0.013
cell	0.015	early	0.009	cerebral	0.018	schwann	0.014	involve	0.012
guidance	0.014	zebrafish	0.009	blood	0.018	lesion	0.013	important	0.012
axonal	0.012	develop	0.008	factor	0.016	fiber	0.011	regulate	0.011

Tab. 4.4: Five highest related topics for descriptor "Neuropilin-1" ordered by $p(z|m)$ (bold). For each topic the list of the ten most probable words along with $p(w|z)$ is presented.

Topic 10 and 3 directly relate to the two primary functions of Neuropilin-1: axonal guidance and vascular formation. The other three topics are more general topics but still relate indirectly to the descriptor. These are particularly good results given the small sample of documents and the specificity of the concept. Assuming that results for a given descriptor improve with its number of appearances in the corpus we could infer that even with this small corpus we have generated at least 5,000 good correlations.

As a whole, these results are encouraging and show a clear semantic correlation between topics and descriptors, which is remarkable given that we did not perform any special tuning on the corpus or the model.

4.1.5 Semantic profiles of Blue Brain Project researchers

Topic modeling was applied to the article library of several neuroscientists at BBP. Articles were voluntarily collected from their personal computers. The first goal was to create semantic profiles of researchers through the articles they read. Another goal was to be able to compare semantic profiles between researchers. A third goal was to create a semantic recommendation engine for newly published paper.

Two topic models were trained on two different corpora. The first was trained on all PubMed abstracts, while the second was trained on a subset of abstracts related to neuroscience. The first topic model was trained on *all* 21,034,484 PubMed abstracts using DCA with 200 topics, hyperparameters optimization and by removing words with a frequency below 100 or belonging to a stop word list of 524 common English words. Training took approximately one day on a workstation (1000 iterations of Gibbs sampling).

Figure 4.9 shows the aggregated topic distribution among BBP researchers. Topics with high probability are detailed in Table 4.5. Probability distributions among researcher is very consistent, demonstrating that topic modeling is able capture the high-level semantics of a collections of documents.

The second topic model was trained on 1,030,546 PubMed abstracts *related to neuroscience* with same parameters and preprocessing as the first model. Compared to the first corpus, it is much smaller and more focused on the subject of neuroscience. This results in topic

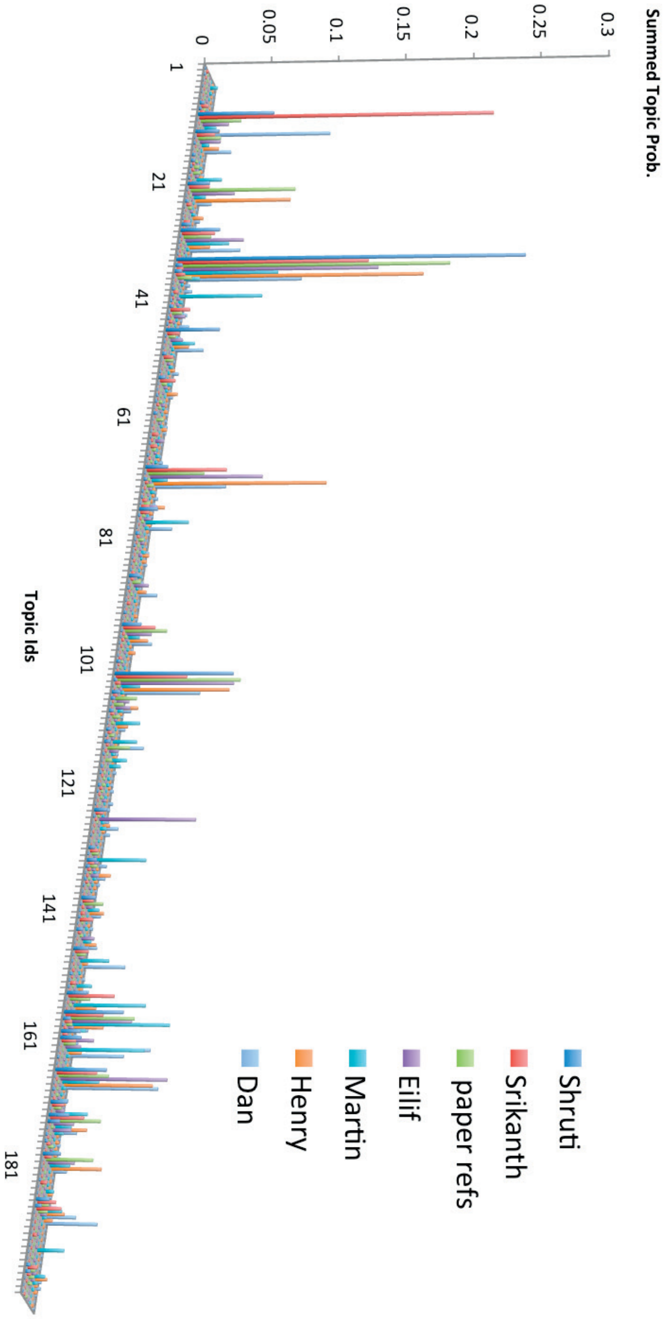


Fig. 4.9: Topic distribution of selected BBP researchers on a topic model trained on *all* PubMed abstracts. Consistency of topics between researchers is very high, a demonstration that topic modeling can capture the high-level semantics of a collections of documents. `papers refs` contains all cited references from a large article submitted by BBP researchers. These references are also very much aligned with the topic distributions from the other researchers. See Table 4.5 for the topics descriptions.

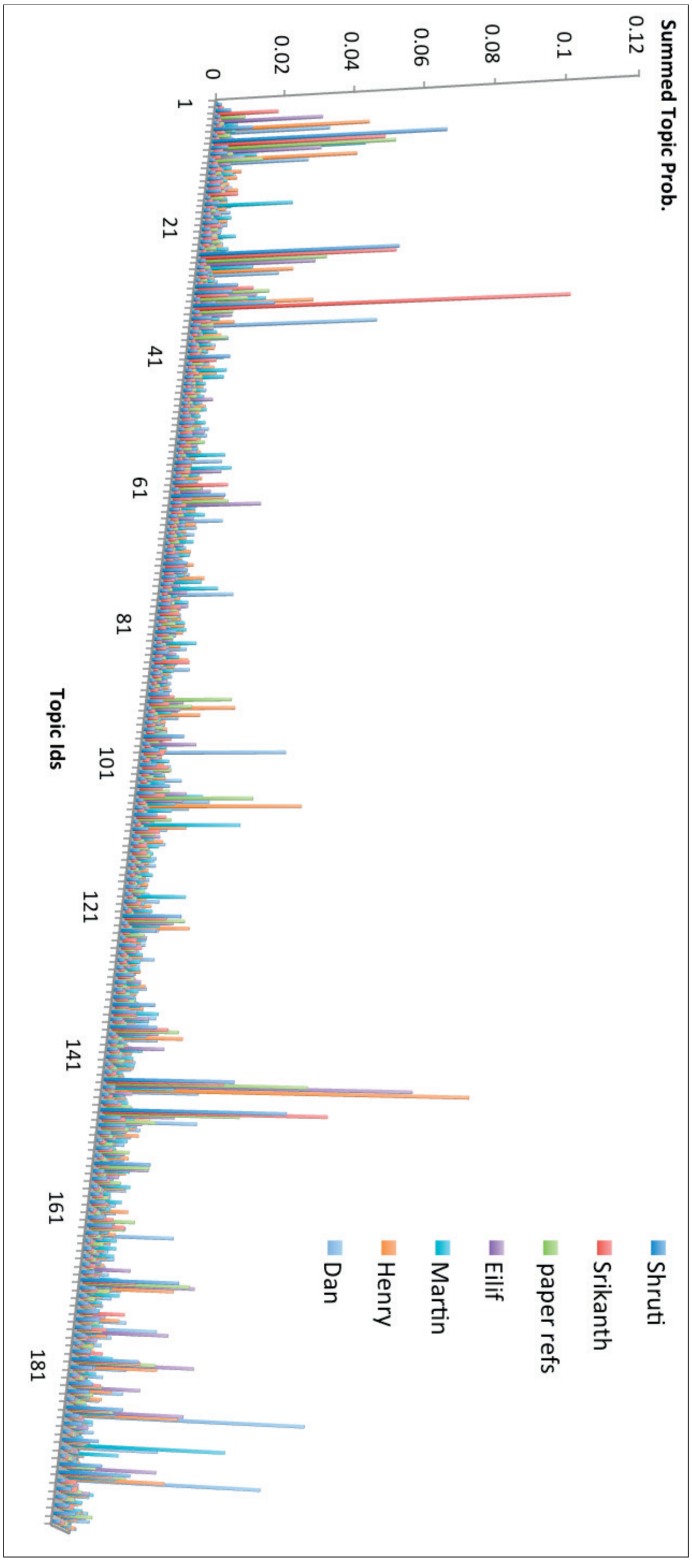


Fig. 4.10: Topic distribution of selected BBP researchers on a topic model trained on PubMed abstracts *related to neuroscience*. Note that in Figure 4.9, topic probabilities are concentrated on a small set of topics, whereas here topic probabilities are more distributed on a larger number of topics. This reflects the fact that the corpus is more focused on a specifc subject (neurosciences). See Table 4.6 for the topics descriptions.

Tab. 4.5: The 10 most frequent terms for the 6 most probable topics among BBP researchers. Terms are sorted by decreasing importance ($\log p(w|z)$). Topic model were trained on *all* PubMed abstracts. One can observe that the topics capture high-level neuroscience concepts. For instance, topic 9 is focused on membrane potential (ion channels, millivolts, action potential, ...).

topic 9	topic 34	topic 69	topic 103	topic 166
current	neurons	model	stimulation	system
channel	cell	data	activity	process
potential	neuronal	analysis	response	mechanisms
membrane	immunoreac.	method	neurons	conditions
mv	brain	based	electrical	development
k+	glial	parameters	evoked	functional
conductance	astrocytes	results	recorded	factors
cells	layer	approach	motor	state
action	synaptic	experimental	stimuli	function
voltage	olfactory	time	reflex	result

that are more specific to neuroscience. Figure 4.10 shows the aggregated topic distribution among BBP researchers. While in Figure 4.9 topic probabilities were concentrated on a small set of topics, in this figure topics probabilities are more distributed among all topics. This results in a more heterogeneous topic distribution between researchers, but also within each researcher's topic distribution.

Aggregated topics from a researcher's reading list can accurately describe that researcher's focus of interest. For example, *Dan* is a lecturer at EPFL on the subject of synaptic plasticity and its most probable topic (186) contain the most frequent words *synaptic* and *plasticity*.

Tab. 4.6: The 10 most frequent terms for the 12 most probable topics among BBP researchers. Topic model trained on selecte PubMed abstracts related to neuroscience (unlike Table 4.5, where the topic model was trained on *all* PubMed abstracts). Terms are sorted by decreasing importance ($\log p(w|z)$). One can observe that topics are much more specific than in Table 4.5.

topic 2	topic 6	topic 7	topic 25	topic 30	topic 33
model	hormone	inhibitory	dendritic	understanding	current
experimental	estrogen	inhibition	layer	recent	channel
data	steroid	excitatory	dendrite	mechanism	potential
predict	estradiol	synaptic	spine	development	inactivation
base	testosterone	transmission	axon	molecular	block
prediction	progesterone	interneuron	cell	review	conductance
simulation	level	gabaergic	neuron	provide	potassium

topic 105	topic 144	topic 148	topic 170	topic 180	topic 186
method	network	potential	cortical	activity	synaptic
data	neural	action	area	firing	plasticity
algorithm	information	membrane	cortex	unit	long-term
approach	input	slice	visual	discharge	potentiation
propose	circuit	recording	subcortical	spike	change
set	functional	depolarization	neocortex	burst	induction
image	processing	amplitude	region	spontaneous	mechanism

4.2 braiNER: Large-scale extraction of brain connectivity from the neuroscientific literature[10]

In the last decades, thousands of experiments on brain region connectivity have been published in scientific journals. However, these have not been systematically normalized and registered in a central repository of brain region connectivity. Instead, these experimental results are published in natural language, scattered among individual scientific publications. This lack of normalization and centralization hinders the large-scale integration of brain connectivity results. Thus, researchers resort to manual searches on PubMed that are very time consuming.

In this section, we present text-mining models to extract and aggregate brain connectivity results from 13.2 million PubMed abstracts and 630,216 full-text publications related to neuroscience. The brain regions are identified with three different named entity recognizers and then normalized against two atlases: the Allen Brain Atlas (ABA) and the atlas from the Brain Architecture Management System (BAMS). We then use three different extractors to assess inter-region connectivity. Named entity recognizers and connectivity extractors are evaluated against a manually annotated corpus. The complete *in-litero* extraction models are also evaluated against in-vivo connectivity data from ABA with an estimated precision of 78%. The resulting database contains over 4 million brain region mentions, and over 100,000 (ABA) and 122,000 (BAMS) potential brain region connections. This database drastically accelerates connectivity literature review, by providing a centralized repository of connectivity data to neuroscientists. The resulting models are publicly available at github.com/BlueBrain/bluima

Brain Connectivity data integration

Brain connectivity data consists of information about one brain region projecting nerve fibers to another region and forming synaptic connections. Additional metadata includes for example connection strength, animal species and experimental methods.

Brain connectivity data can be integrated from different sources. For the mouse brain, one central source is the *Allen Mouse Brain Connectivity Atlas* (AMBCA, [Oh+14]). As of today, the Allen Institute has published 1772 standardized connectivity experiments tracking axonal projections in the adult mouse brain by two-photon imaging of fluorescently labeled neurons. Experimental results have been normalized to a coordinate-based reference space and are freely available to researchers via a publicly accessible API[11]. The AMBCA is a very valuable source of connectivity data because of the consistency of the experimental methods, the standardized brain region naming, the availability of the data and the overall high level of quality of the data.

A second central source of connectivity data comes from curated databases of the published literature. For the rat brain, the most important is the *Brain Architecture Management System* (BAMS, [BS08]). Neuroscientists from the BAMS project have manually curated over 600

[10] A version of this chapter has been published as [Ric+15].

[11] connectivity.brain-map.org/

Tab. 4.7: Example of sentences exhibiting connectivity statements between brain regions. Abbreviations have been manually added.

Sample sentence	Connectivity statement, comment
The nucleus accumbens (AC) receives projections from both the substantia nigra (SN) and the ventral tegmental area (VTA) (Dworkin, 1988).	(SN, VTA) → AC
Substantial numbers of tyrosine hydroxylase-immunoreactive cells in the dorsal raphe nucleus (DR) were found to project to the nucleus accumbens (AC) (Stratford and Wirtshafter, 1990).	DR → AC
The dentate gyrus (DG) is, of course, not only an input link between the entorhinal cortex (Ent) and the hippocampus proper (CAs), but also a major site of projection from the hippocampus (CA), as are the amygdala (Amg), entorhinal cortex (Ent), and septum (Spt) (Izquierdo and Medina 1997).	CAs → DG → Ent, (CA, Amg, Ent, Spt) → DG Complex, long range relationships
This latter nucleus (N?), which projects to the striatum (CP), receives inputs from motor cortex (MO) as well as the basal ganglia (BG), and is situated to integrate these and then provide feedback to the basal ganglia (BG) (Strutz 1987).	MO → N? → CP, BG ↔ N? Anaphora: "latter nucleus (N?)" was defined in previous sentence
In this review, we summarize a classic injury model, lesioning of the perforant path, which removes the main extrahippocampal input to the dentate gyrus (Perederiy and Westbrook 2013).	Injury model, not normal conditions
The most commonly proposed mechanism is that the periaqueductal gray of the midbrain (PAG) or the cerebral cortex (Cx) have descending influences to the spinal cord (SpC) to modulate pain transmission at the spinal cord (SpC) level (Andersen 1986).	PAG → SpC, Cx → SpC "proposed" implies an hypothesis, not a finding

scientific articles. They analyzed each article (including tables, images and supplementary materials) and assessed the quality of the experiment. Finally, they normalized brain region mentions to the BAMS ontology, and recorded the connectivity data into a database (incl. directionality and strength).

One other major source of connectivity data is the analysis of neuroscientific articles. This is commonly performed by *manual search* on databases like PubMed or Google Scholar. The search, curation and integration of these articles might be a manageable task for a researcher focused on one or a few brain regions, but it does not scale for whole-brain models. Furthermore, manual search for brain region connections has several disadvantages. First, the naming of brain regions is diverse [Boh+09b], making it difficult to search for brain region names. There is no single nomenclature that accommodates all the uses and expectations. For each animal model, several nomenclatures are available to name brain regions. These nomenclatures all have different objectives and perspectives. For example, the rat atlas from [PW06] is preferred for stereotactic surgery orientation whereas the rat atlas from [Swa04] is preferred for finer anatomy classification. Moreover these nomenclatures rely on different detection methods (e.g. Nissl staining, immunostaining, functional magnetic resonance imaging, diffusion tensor imaging) that result in different sizes and shapes of brain regions. Typically, a researcher will select the most appropriate nomenclature, depending on which area she focuses on.

Another disadvantage of manual search is its low recall[12]. It is likely to miss a significant part of the brain regions because it *lacks synonym expansion*[13]. For example, exact search for "Basolateral amygdala nucleus" (17 results on PubMed) will neither return results from the synonym "Basolateral nucleus of the amygdala" (297 results) nor from the Latin name "Nucleus amygdalae basolateralis" (8 results). Another reason for low recall is the *lack of abbreviation expansion*. For example, when searching for "Ventral tegmental area", the abbreviated form "VTA" will not be retrieved. A random sample corpus of 179 full-text articles from the Journal of Comparative Neurology contained on average 91.6 brain regions mentions and 29.7 abbreviations of brain regions per article. This represents a maximal possible 32% increase in recall when performing abbreviation expansion[14]. Additionally, for a significant number of articles in PubMed, *only the abstract* is indexed and searchable, not the full article body. On the above-mentioned corpus, the abstracts contained on average 2.8 brain region mentions. This represents a possible 32-fold increase in recall when using full-text instead of abstracts.

In terms of precision[15], a manual search will return all brain regions that co-occur within the same document. Most of these *co-occurrences* do not necessarily represent true neurophysiological connections, but simply that two brain region are mentioned in the same document. At the abstract level, [Fre+12] found that only 2.2% of the co-occurrences represent true connections. At the sentence level, the proportion raises to only 13.3%. Thus, the precision of manual search is expected to be quite low, meaning that researchers will waste time in manually post-processing the search result and probably discard most retrieved co-occurrences.

Information Extraction

The IE process is divided in two phases: named entity recognition (NER) and relation extraction (see Fig. 4.11). Named entity recognition is performed by three different models: *BAMS* and *ABA* (lexical-based), and *BraiNER* (machine learning-based). Once named entities have been identified, we normalize them, so that e.g. both "diencephalon" and "interbrain" resolve to the same entity. Normalization can be performed by automatically or manually attaching synonyms to lexical-based NER, or by performing morpho-syntactic transformations on the brain regions extracted by a NER. For example, [FP12] used transformation to remove prefixes that specify hemispheres ("Contralateral inferior olivary" is transformed into "Inferior olivary"), or to remove neuroanatomical direction specifiers ("Caudal cuneate nucleus" is transformed into "Cuneate nucleus").

The second and last IE step involves relationship extraction. It aims at classifying co-occurrences between two brain region entities and predicting whether they represent neurophysiological connections[16]. Models for relationship extraction include rule-based and supervised machine learning approaches. Relationship extraction between two biomedical entities

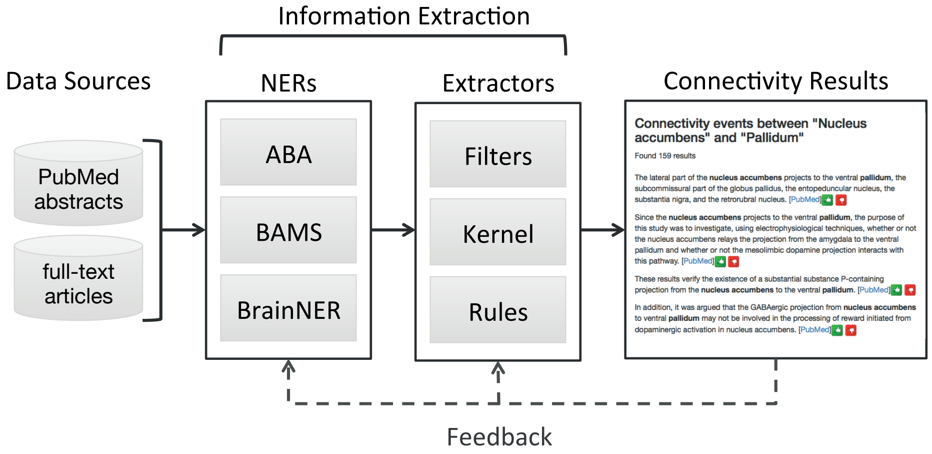
[12] *Recall* is the ratio of the number of relevant records retrieved to the total number of relevant records.

[13] There is actually a synonym expansion mechanism in the PubMed search engine. However, it is mostly limited to molecular entities, not brain regions.

[14] Note however that an article containing an abbreviated brain region might still be returned by a manual search, since abbreviations are almost always explicitly defined in an article, so the expected increase is smaller.

[15] *Precision* is the fraction of retrieved records that are relevant.

[16] It is worth noting that IE only predicts whether the author *reports* a connection between two brain region, not whether the connection *actually exists*, which is out of the scope of such an IE system.



Connectivity events between "Nucleus accumbens" and "Pallidum"

Found 159 results

The lateral part of the **nucleus accumbens** projects to the ventral **pallidum**, the subcommissural part of the globus pallidus, the entopeduncular nucleus, the substantia nigra, and the retrorubral nucleus. [PubMed] 👍 👎

Since the **nucleus accumbens** projects to the ventral **pallidum**, the purpose of this study was to investigate, using electrophysiological techniques, whether or not the nucleus accumbens relays the projection from the amygdala to the ventral pallidum and whether or not the mesolimbic dopamine projection interacts with this pathway. [PubMed] 👍 👎

These results verify the existence of a substantial substance P-containing projection from the **nucleus accumbens** to the ventral **pallidum**. [PubMed] 👍 👎

In addition, it was argued that the GABAergic projection from **nucleus accumbens** to ventral **pallidum** may not be involved in the processing of reward initiated from dopaminergic activation in nucleus accumbens. [PubMed] 👍 👎

Fig. 4.11: Overview of datasets, methods, and models. Three named entity recognizers (NER) identify and normalize brain region mentions: *BAMS* and *ABA* (lexical-based), and *BraiNER* (machine learning-based). Three different extractors predict the connectivity probability of brain region co-occurrences: *Filters* takes a top-down filtering approach, *Kernel* is a machine learning-based classifier, and *Rules* consists of hand-written extraction rules. Connectivity results are presented in a searchable web interface. In the future, feedback from the interface can be used to retrain the NERs and extractors for continuous model improvement.

is a current research topic, applied to problems like protein-protein interaction [Kra+11] or pathway curation [Oht+13]. The difficulty of the task resides in the complexity of the relation between two or more brain regions (see Table 4.7). [Fre+12] developed and evaluated several models to extract brain region connectivity. Their simple co-occurrence based methods yielded high recall but low precision, whereas the advanced machine learning models recalled 70.1% of the sentence-level connectivity statements at 50.3% precision. More complex models based on dependency parsing were successfully evaluated by [Fre+12], but discarded because of their high computational cost.

Our work builds on top of French *et al.* (2009, 2012 and 2012) and extends it in several aspects: ensemble of three different extractors and application to a large corpus of over 8 billion words.

4.2.1 Methods

To build a database of brain region connectivity data from the literature, two steps are required. First, named entity recognizers[17] (NER) identify brain region mentions in text and normalize them to a standard brain region ontology. Second, extractors are developed to determine whether two brain region co-occurrence mentions are semantically connected. Finally, the connectivity results are stored in a database to be accessible by researchers.

Tab. 4.8: Named entity recognizers for brain regions.

NER Name	Description	Brain Regions	Terms
ABA	lexicon from Allen Brain Atlas Institute	1,197	1,197
ABA-SYN	ABA + automated synonyms enrichment from other lexica	1,197	3,882
BAMS	lexicon from Brain Architecture Management System (BAMS), version Swanson 2004	832	832
BAMS-SYN	BAMS + automated synonyms enrichment from other lexica	832	2,705
BraiNER	machine learning-based NER (linear chain conditional random field)	(∞)	(∞)

Brain Region Named Entity Recognizers

Three different NERs have been developed to identify and normalize brain region mentions (Table 4.8). The first lexical NER[18] (`ABA`) consists of all 1197 entities from the Allen Mouse Brain Atlas[19]. As discussed in section 4.2, the atlas is designed to structure and organize brain regions within the Allen Brain Institute and not as a lexical resource for IE. Thus, the `ABA` NER contains no synonyms. To retrieve more relevant data (and improve recall), a second NER (`ABA-SYN`) is automatically augmented with corresponding synonyms found in several lexica of rodent brain region: BAMS [BS08], [Hof+00], Neuronames [BD03], [PW06], [Swa04] (see Section 2.1 for a detailed description of the different lexica). For example for the `ABA` entity "Pontine gray", the Neuronames lexicon also contains several synonyms (e.g. "Nuclei pontis"), that are added back to the corresponding `ABA` entry. This results in a 3-fold increase in recall between `ABA` and `ABA-SYN`. To further improve recall, `ABA-SYN` is manually augmented with brain region mentions appearing frequently in scientific articles, but not included in `ABA-SYN`. Additionally, abbreviation expansion is performed on the input text using a machine learning-based model (hidden Markov model, [MAC12]). The same procedure for `ABA` is applied to the `BAMS` ontology[20]

The third brain region NER, `BraiNER`, extends the work from [Fre+09] and relies on a supervised machine learning model (linear chain conditional random field [@McC02]. The model is trained on WhiteText[21], a manually annotated corpus of brain region mentions composed of 1,377 PubMed abstracts from the *Journal of Comparative Neurology*, containing 18,242 brain region mentions. Inter-annotator agreement was evaluated by [Fre+09] by two curators for a subset of the documents, and reached 90.7% and 96.7% for strict and lenient matching respectively.

The model features from [Fre+09] are primarily derived from existing neuroanatomical lexica. These include for example lexical features such as the presence of directionality words like *dorsal* or *ventral*, or morphological features like the word length or whether it contains only lowercase letters, numbers or special characters. `BraiNER` uses the following additional features: the presence of species information in the document (identified using the Linnaeus

[17] See Section 2.3 for a thorough description of named entity recognition.

[18] Lexical matching is performed using UIMA ConceptMapper, with order-dependant lookup, longest contiguous match, and a stemmer that removes ending s of words longer than 3 characters.

[19] Allen Reference Atlas - version 2 (2011), Mouse Brain Atlas Ontology

[20] [Swa04]

[21] www.chibi.ubc.ca/WhiteText

NER [Ger+10]), and the presence of a measure entity (e.g. a measure like *10mm* or *10(-7) molar*). Indeed, a qualitative analysis of the performance of BraiNER on full-text articles revealed that measures were often incorrectly labeled as brain regions (false positives). Furthermore, several other features are developed to improve robustness on full-text articles, motivated by the large amount of false positives when analyzing full-text articles, in particular when processing bibliographical information or tables.

Connectivity Extractors

Connectivity extractors are binary classifiers. They take as input a sentence containing at least two brain region mentions (as identified by the above NERs) and take a decision whether the sentence enunciates a connection between these two brain regions. The models developed here focus on extracting connections with high precision. They are limited to brain regions that are co-located within the same sentence (no anaphora resolution), and do not extract the directionality of the connection.

Three different approaches are developed to classify connectivity statements (Figure 4.11). 1) `FILTER` considers all possible co-occurrences of brain regions, and subsequently applies filters to remove unlikely ones. More precisely, it starts with all permutations of brain regions within a sentence, and then keeps only nearest neighbors, that is: only co-occurrences that are located closest to each other. After that, co-occurrences in sentences longer than 500 characters are removed, since longer sentences are unlikely to be meaningful sentences. Similarly, sentences containing more than 7 brain regions are removed, since they are too complex to extract. These filters were developed based on our experience with full-text articles that can contain very long sentences or lists of brain regions. Finally, only sentences containing one of the following trigger character sequences are retained: *afferent, efferent, project, connecti, pathway, inputs*. 2) `KERNEL` relies on a supervised classifier (shallow linguistic kernel [Giu+06], identical to [Fre+12]) that requires only shallow parsing information such as word occurrences and part-of-speech tags. 3) `RULES` consists of 9 rules of the kind *"projection from the regionA (of the regionB) to the regionC and the regionD"*. Here, the strategy is to identify characteristic sentence constructs, and thus achieve a very high precision at the cost of recall. Rules are manually crafted using the Apache UIMA Ruta scripting language [Klu+14b; Klu+09]. The Ruta language enables a rapid and iterative development of lexical rules (see Section 3.3).

4.2.2 Evaluation

We begin by quantitatively evaluating the performance of the brain region NERs and connectivity extractors against annotated corpora. We then build a database by applying these models on three different corpora. We describe the database and conclude by performing a qualitative evaluation of the database against the connectivity data from ABA.

NERs Evaluation

All five NERs described in table 4.8 are evaluated against *WhiteText* [Fre+09] (see Table 4.9). Two types of evaluations are performed: exact comparison (meaning that the span of a proposed brain region must exactly match a manually annotated brain region) and lenient comparison (meaning that the span of an identified brain region may be equal or smaller than

Tab. 4.9: Performance comparison of brain region NER models against the WhiteText corpus (partially matching spans). For machine learning-based NERs ([Fre+09] and BraiNER), average values over 8-fold cross validation with splits at document level and 5 repetitions, including std. deviation in parenthesis where appropriate.

Model	Exact comparison			Lenient comparison		
	Precision	Recall	F-score	Precision	Recall	F-score
ABA lexicon	58.4%	11.1%	18.6%	89.9%	16.9%	28.5%
ABA-SYN lexicon	58.4%	21.9%	31.9%	**92.1%**	34.2%	49.9%
BAMS lexicon	61.1%	11.0%	18.6%	90.7%	16.2%	27.5%
BAMS-SYN lexicon	61.3%	17.5%	27.2%	89.8%	25.5%	39.7%
WhiteText [Fre+09]	81.3%	76.1%	78.6%	91.6%	**85.7%**	**88.6%**
BraiNER-W (WhiteText features)	83.6% (3.3)	76.4% (4.6)	79.8% (3.9)	87.1% (3.6)	77.8% (7.4)	82.1% (5.8)
BraiNER (additional features)	**84.6%** (1.3)	**78.8%** (1.2)	**81.6%** (0.9)	88.4% (1.0)	81.0% (1.8)	84.6% (1.3)

a manually annotated brain region). When performing exact comparison, lexical-based NERs score low on both precision and recall. For both NERs enriched with synonyms (ABA-SYN and BAMS-SYN), recall is significantly higher (21.9% and 17.5% respectively). Using lenient comparison, lexical-based NERs score much higher on precision (between 89.8% and 92.1%). However, recall is low, even with synonyms (between 16.2% and 34.2%). One reason why lexical-based NERs do not achieve perfect precision is that they wrongly label implicit brain regions (e.g. they will label "midbrain" in "midbrain ventral tegmental area" or "midbrain lateral tegmental field"). Another reason is that they sometimes label brain regions that are more specific (e.g. "brachium of the superior colliculus" was labeled, whereas the gold-standard only includes "superior colliculus").

For machine learning NERs, we first reproduce the results from [Fre+09], using the same model and features[22]. This model is denoted BraiNER-W and its performance is slightly higher than the results reported by [Fre+09] for exact comparison (83.6% precision against 81.3% and 76.4% recall against 76.1%). This can be explained by the differences in preprocessing (tokenization, part-of-speech, abbreviation expansion). For lenient comparison, results from BraiNER-F are slightly worse, probably because we use a stricter lenient comparison criterion. Finally, we evaluate BraiNER, that includes additional model features. Performance is slightly higher than BraiNER-W (e.g. F-score 81.6% against 79.8% in strict comparison and 84.6% against 82.1% in lenient comparison). However, differences are not statistically significant. Nevertheless, qualitatively we found that the performance of BraiNER is higher when analyzing full-text articles.

Compared to lexical-based NERs, both machine learning-based NERs score slightly higher on precision, but have a much higher recall (more than twice as much). However, the low recall of lexical-based NERs is still acceptable for our purpose, since we apply theses NERs on very large corpora and focus on precision.

[22] github.com/leonfrench/public/

Connectivity Extractors Evaluation

The connectivity extractors are evaluated on the *WhiteText connectivity* corpus from [Fre+12], that goes beyond the original *WhiteText* corpus and contains 3,097 manually annotated connectivity relations across 989 abstracts and 4,338 sentences from the *Journal of Comparative Neurology*. Inter-annotator agreement reaches a precision and recall of 93.9% and 91.9%, respectively (partially matching spans, two curators). In this evaluation, the locations of the brain region entities in the text are provided, so we are only concerned with the evaluation of the extractors.

Table 4.10 presents the evaluation results. The baseline connectivity extractor returns all permutations of two brain regions within a sentence, and has a perfect recall of 100% but a very low precision of 9%[23]. Subsequently, 4 filters are applied and evaluated. The first two (filter if sentence is longer than 500 characters or contains more than 7 brain regions) do not significantly improve precision on the evaluation corpus, but they proved very effective when dealing with full-text articles. The next filter requires certain trigger words (like *project*) to be present in the sentence and improves the precision to 15%. The last filter (keeping only nearest neighbors co-occurrences) improves the baseline precision (9%) threefold to 28%. When combining all filters (`FILTERS`), almost half of the extracted connections are correct (45% precision). However, only 31% of the connections are recalled.

For the machine learning model (`KERNEL`), 10-fold cross-validation with splits at document level is performed, resulting in a precision of 60%. Recall (68%) is significantly higher than with `FILTERS`. Finally, `RULES` (manually created rules) yields the highest precision, at the cost of a very low recall. Still, this performance is quite remarkable, considering its simplicity (only 9 rules).

Ensemble of extractors are also considered to improve precision. For example, the connections returned by all three extractors have a highest precision of 82% at only 7% recall. For connections returned by `FILTERS` or `KERNEL`, together with `RULES`, the performance is 80% precision at 11% recall.

Database

The models presented in this section are applied to two large corpora of biomedical literature (see Section 3.1). The resulting brain connectivity statements are stored in a database, and an interface is created to navigate and make the results accessible to neuroscientists (see Fig. 4.11).

Connections are extracted using bluima (see Section 3.2. The processing is distributed on a cluster and the extraction results are aggregated in a database. The resulting database contains several million brain region mentions (see Table 4.11). In the PubMed abstracts, 42, 50 and 189 thousand connection pairs are extracted for ABA, BAMS and BraiNER, respectively. For the full-text neuroscience corpus, 62, 72 and 279 thousand connection pairs are extracted for ABA, BAMS and BraiNER, respectively. Comparatively, [Fre+15] extracted 68,957 connections between 88,088 brain region mentions.

[23] Note that [Fre+12] estimated that over a forth of all connectivity relations are formed with regions spanning different sentences. Extracting connections that span sentences was not considered and the evaluation is performed without accounting for the relations spanning sentences.

Tab. 4.10: Evaluation of extraction models against the WhiteText corpus.

Extractor	Prec.	Recall	F-score
all co-occurrences (all permutations)	9%	**100%**	16%
filter sentence >500 characters	10%	93%	18%
filter sentence with >7 brain regions	11%	80%	19%
keep if contain trigger words	15%	53%	23%
keep nearest neighbour co-occurrence	28%	51%	36%
all filters (FILTERS)	45%	31%	37%
shallow linguistic kernel (KERNEL)	60%	68%	**64%**
Ruta rules (RULES)	**72%**	12%	21%
FILTERS and KERNEL	66%	**19%**	**29%**
FILTERS and RULES	80%	7%	13%
KERNEL and RULES	81%	10%	18%
FILTERS and KERNEL and RULES	**82%**	7%	12%
(FILTERS or KERNEL) and RULES	80%	11%	19%

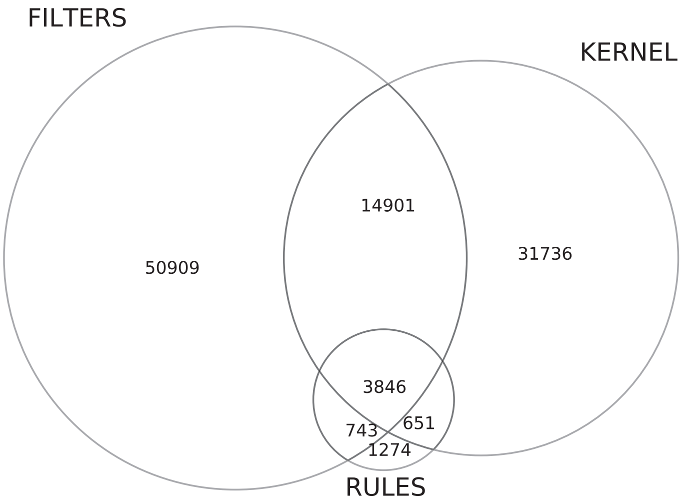


Fig. 4.12: Overlap between extractors. Venn diagram depicting the number of extracted connections for the three extractors, on PubMed and full-text corpora using the ABA-SYN NER. As one can see, while there is some overlap among the connections extracted by the three different extractors, there is also a significant amount of connections that were extracted from a single extractor (no overlap).

Tab. 4.11: Statistics of the corpora used, extracted brain regions and connections using all three extractors (FILTERS or KERNEL or RULES). The number of documents and words refers to non-empty documents after pre-processing (see Section 3.1.4 for detailed information about the corpora). Two generic terms from BAMS "brain" and "nerve") are omitted.

Corpus	Corpus statistics		Brain Regions			Connectivity statements		
	Documents	Words	ABA	BAMS	BraiNER	ABA	BAMS	BraiNER
all PubMed abstracts	13,293,649	2.1×10^9	1,705,549	1,918,561	1,992,747	41,965	50,331	188,994
full-text neuroscience articles	630,216	6.1×10^9	2,327,586	2,514,523	2,751,952	62,095	72,602	279,100

Figure 4.12 highlights the overlap of the results from all three extractors. For example, 31,736 connections are extracted uniquely by KERNEL, whereas all three extractors return 3,846 connections. Thus, each extractor contributes to extracting a different set of brain region connections, with a different performance. This will turn out to be useful to display connectivity data: the connections that are returned by all three extractors have a higher estimated precision and ought to be displayed at the top of the list of proposed results.

The database is accessible through a web service, with a simple web front end. It allows neuroscientists to search for a given region and display all other connected regions. It also allows to provide a feedback on the results for future model improvements. Normalization and standardization of brain region entities identified by BraiNER can be manually performed by the user (no morpho-syntactic transformation).

Database Evaluation against AMBCA

Results extracted from the literature (LIT) are evaluated against connectivity data from the *Allen Mouse Brain Connectivity Atlas* (AMBCA). AMBCA data was aggregated in the following way: for each experiments, the normalized projection volume in both hemisphere is retained; injection areas are filtered out if they do not represent at least 5% of the total volume; connectivity data is aggregated over all experiments, and only the highest density value is retained for each injection-projection pair. Nevertheless, we present an evaluation of our results against 1379 mouse brain connectivity experiments from AMBCA[24]. The AMBCA validation corpus consists of the normalized connectivity data from 469 in-vivo experiments[25]. Regions were filtered by two criteria (bigger than 50 voxels and containing enough data for the signal to be well linearly separable), (e.g. at the boundary between two regions there are only two injections but which are perfectly on top of each other such that the separate contributions of the two regions cannot be well separated). resulting in 213 selected regions (out of a total of 1,204 regions in the complete ontology). Thus, AMBCA consists of a square matrix of 213 brain regions, whose values represent normalized ipsilateral connection strengths (integral of the segmented area in the target region, normalized to the injection

[24] Retrieved using ABA's public API http://www.brain-map.org/api/index.html; stand as of Feb 2014

[25] See [Oh+14], supplementary Table 3 for the underlying data.

volume). 16,954 brain region pairs are reported as connected (37%), and 28,415 as not connected.

The evaluation of LIT against AMBCA proved to be quite complex. First, it is not possible to determine which articles are missing in LIT (that is: articles that should have been retrieved by LIT but were missed). Therefore, it is not possible to correctly evaluate the *recall* of LIT. Second, AMBCA contains 213 regions, whereas LIT contains 451 regions, thus 238 regions from LIT cannot be evaluated and were removed from the evaluation. Third, many ABA brain regions never occur in the literature (mainly because they are very specific, like "Anterior cingulate area, dorsal part, layer 2/3"). In fact, half of the ABA regions (603 out of 1,204) are never found in the literature by the ABA lexical NERs. Forth, AMBCA uses one single and systematic experimental method, whereas many different methods and experimental settings are reported in scientific reports from AMBCA, making the comparison problematic. Fifth, it is important to highlight that the *frequency* of a brain region connection reported in scientific articles does not necessarily reflect the *physiological intensity* of a connection; the former reflecting the *popularity* of a region.

Despite all these limitations, the evaluation is highly relevant, as it allows to compare our models with experimental data. Figure 4.14 illustrates the evaluation results. 904 brain region pairs are correctly predicted (present in LIT and connected in AMBCA) and 261 brain region pairs are incorrectly predicted (present in LIT but not connected according to AMBCA), resulting in a 78% precision, which is an impressively good result regarding the five previously mentioned limitations of this evaluation. In comparison, the precision of co-mentioned brain region mentions (two brain regions within the same sentence, without any filtering) is 67%. By thresholding co-mentions to those predicted at least four times, precision reaches 72%, suggesting that frequent co-mentions can successfully predict connectivity.

The 6784 brain region pairs present in LIT but not in AMBCA (represented as the blue square with white background in Figure 4.14) are valuable connections that might complement experimental datasets like AMBCA[26]. Furthermore, when using another NER like BrainNER, even more brain regions (not present in AMBCA) would be retrieved, resulting in an even larger size of LIT.

Figure 4.13 shows the *in-vivo* connectivity matrix from AMBCA (left), the symmetrised matrix from AMBCA (middle, required, to compare against the NLP models that do not extract directionality), and the *in-litero* connectivity matrix extracted from the literature (LIT, right). The LIT matrix is much sparser than AMBCA, as was previously noted. However, both matrices exhibit a similar structure. To evaluate this similarity, the precision between LIT and AMBCA (symmetrised) matrices are compared against 1000 random matrices created by shuffling the brain region names in the same way for rows and columns. That ensures symmetry with the same node degree distribution and density. LIT is significantly closer to AMBCA than the random matrices ($p < 0.01$).

No significant difference in precision can be observed between the connections originating from abstracts and the ones from full-text papers. Similarly, no significant difference in distance can be observed between abstracts and full-text papers. We also evaluate the depth of the extracted connections, measured as the mean number of parents (higher structures) in

[26] However, it is impossible to quantitatively evaluate these brain regions, because of the lack of objective reference.

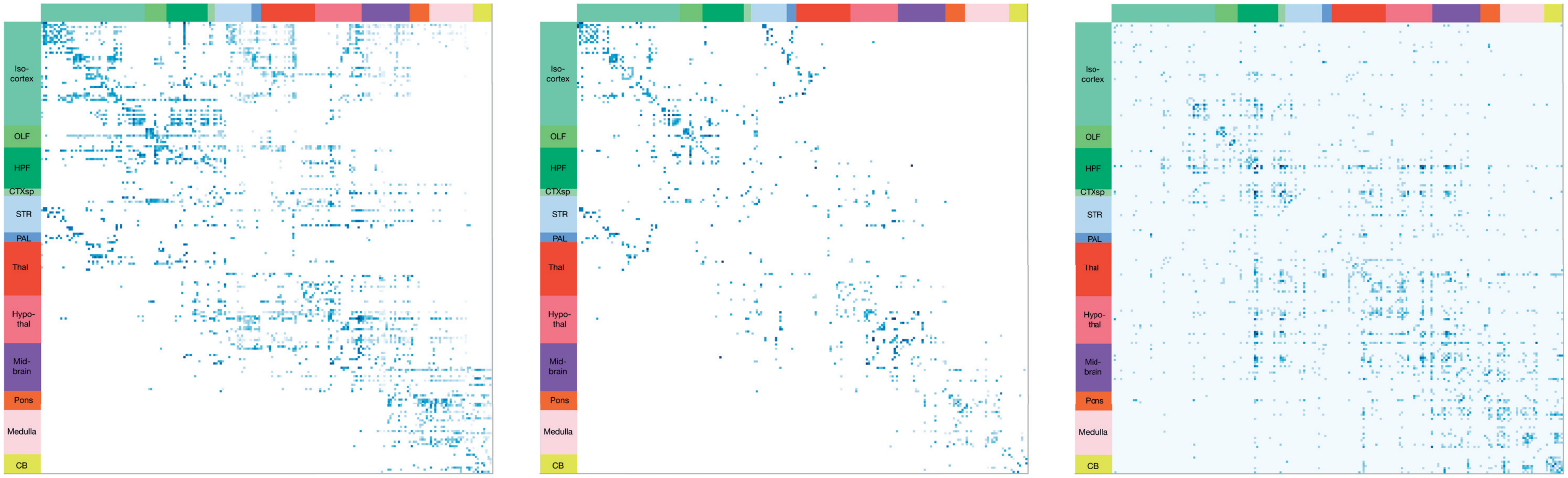


Fig. 4.13: Comparison of the inter-region connectivity matrices, renormalized between 0 (white) and 1 (blue). Rows and columns correspond to ABA brain regions. *Left:* connection matrix from AMBCA (ipsilateral), using ABA's inter-region connectivity model, with values representing a combination of connection strength and statistical confidence (see Figure 4a of [Oh+14]). *Middle:* same matrix from AMBCA, but symmetrized (connection directionality is ignored, since the NLP models do not extract directionality). *Right:* connection matrix from the results extracted from the literature (LIT) with values representing the number of extracted connectivity statements, weighted by the estimated precision of each connectivity extractor.

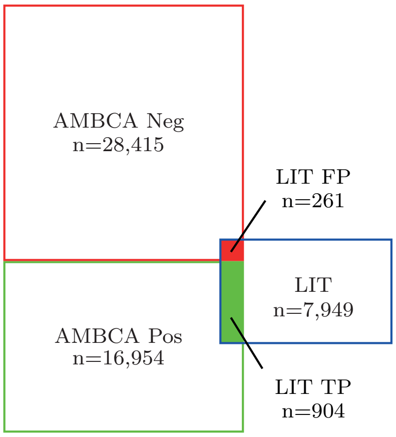
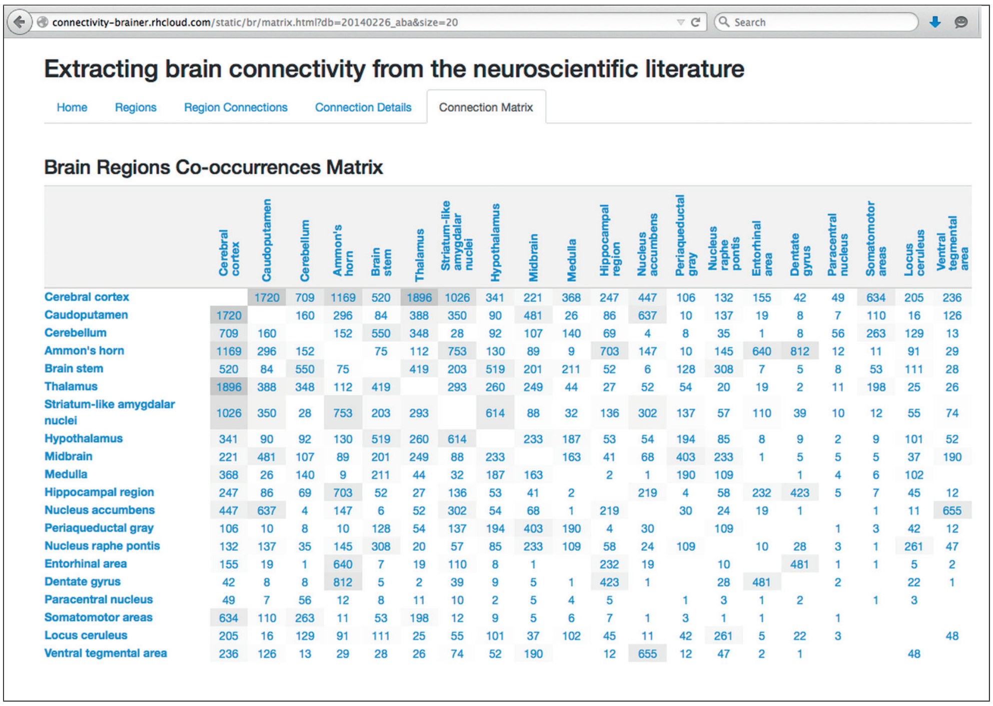


Fig. 4.14: Evaluation against AMBCA. AMBCA contains 16,954 distinct connected brain region pairs (AMBCA Pos) and 28,415 unconnected pairs (AMBCA Neg). Connectivity data extracted from the literature contains 7,949 distinct connected brain region pairs (LIT), of which 904 are connected in AMBCA (LIT TP), and 261 are not connected in AMBCA (LIT TN).

the ABA ontology. Connections from AMBCA have a mean depth of 6.21, whereas connections extracted from the literature have a depth of 5.08.

4.2.3 Connectivity Database

A database of connectivity statements created with the above models is publicly accessible through a simple and intuitive web application. This application provides a matrix of brain regions co-occurences displaying the top N regions for which the most connection mentions was found (see Fig 4.15). All matrix values are linked to the corresponding detailed list of sentences from neuroscientific articles. For example, Suppl. Fig. 4.17 displays the extracted sentences between the Allen Brain Atlas regions "Periaqueductal gray" and "Nucleus accumbens". Each sentence is itself linked to PubMed so that the user can go back to the original article. Additionally, the user has the ability to provide feedback by either validating the sentence or rejecting it. Finally, it is possible to search for one particular brain regions of interest, and then list all the other brain regions potentially connected to it (for which connectivity events have been found in the literature), see Fig. 4.16. The web application also exposes a REST API to interact with the extracted connectivity programmatically.



connectivity-brainer.rhcloud.com/static/br/matrix.html?db=20140226_aba&size=20
Search

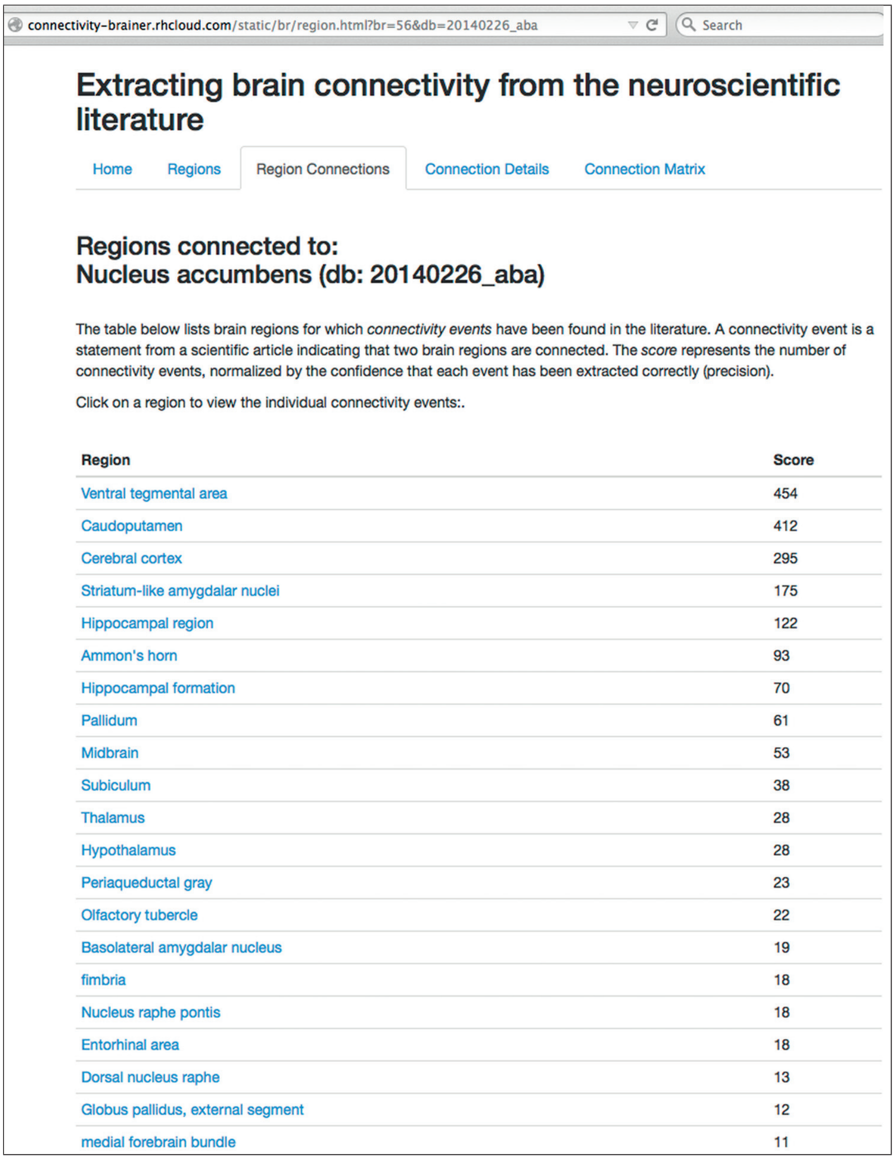
Extracting brain connectivity from the neuroscientific literature

Home
Regions
Region Connections
Connection Details
Connection Matrix

Brain Regions Co-occurrences Matrix

	Cerebral cortex	Caudoputamen	Cerebellum	Ammon's horn	Brain stem	Thalamus	Striatum-like amygdalar nuclei	Hypothalamus	Midbrain	Medulla	Hippocampal region	Nucleus accumbens	Periaqueductal gray	Nucleus raphe pontis	Entorhinal area	Dentate gyrus	Paracentral nucleus	Somatomotor areas	Locus ceruleus	Ventral tegmental area
Cerebral cortex		1720	709	1169	520	1896	1026	341	221	368	247	447	106	132	155	42	49	634	205	236
Caudoputamen	1720		160	296	84	388	350	90	481	26	86	637	10	137	19	8	7	110	16	126
Cerebellum	709	160		152	550	348	28	92	107	140	69	4	8	35	1	8	56	263	129	13
Ammon's horn	1169	296	152		75	112	753	130	89	9	703	147	10	145	640	812	12	11	91	29
Brain stem	520	84	550	75		419	203	519	201	211	52	6	128	308	7	5	8	53	111	28
Thalamus	1896	388	348	112	419		293	260	249	44	27	52	54	20	19	2	11	198	25	26
Striatum-like amygdalar nuclei	1026	350	28	753	203	293		614	88	32	136	302	137	57	110	39	10	12	55	74
Hypothalamus	341	90	92	130	519	260	614		233	187	53	54	194	85	8	9	2	9	101	52
Midbrain	221	481	107	89	201	249	88	233		163	41	68	403	233	1	5	5	5	37	190
Medulla	368	26	140	9	211	44	32	187	163		2	1	190	109		1	4	6	102	
Hippocampal region	247	86	69	703	52	27	136	53	41	2		219	4	58	232	423	5	7	45	12
Nucleus accumbens	447	637	4	147	6	52	302	54	68	1	219		30	24	19	1		1	11	655
Periaqueductal gray	106	10	8	10	128	54	137	194	403	190	4	30		109			1	3	42	12
Nucleus raphe pontis	132	137	35	145	308	20	57	85	233	109	58	24	109		10	28	3	1	261	47
Entorhinal area	155	19	1	640	7	19	110	8	1		232	19		10		481	1	1	5	2
Dentate gyrus	42	8	8	812	5	2	39	9	5	1	423	1		28	481		2		22	1
Paracentral nucleus	49	7	56	12	8	11	10	2	5	4	5		1	3	1	2		1	3	
Somatomotor areas	634	110	263	11	53	198	12	9	5	6	7	1	3	1	1		1			
Locus ceruleus	205	16	129	91	111	25	55	101	37	102	45	11	42	261	5	22	3			48
Ventral tegmental area	236	126	13	29	28	26	74	52	190		12	655	12	47	2	1			48	

Fig. 4.15: Brain regions co-occurrences matrix displaying the top 20 regions for which the most connection mentions was found. Matrix values represent the number of connectivity events, normalized by the confidence that each event has been extracted correctly (precision). All matrix values are linked to the corresponding detailed list of article sentences (see Figure 4.17). The corresponding url for that figure is http://connectivity-brainer.rhcloud.com/static/br/matrix.html?db=20140226_aba&size=20


connectivity-brainer.rhcloud.com/static/br/region.html?br=56&db=20140226_aba
Search

Extracting brain connectivity from the neuroscientific literature

Home
Regions
Region Connections
Connection Details
Connection Matrix

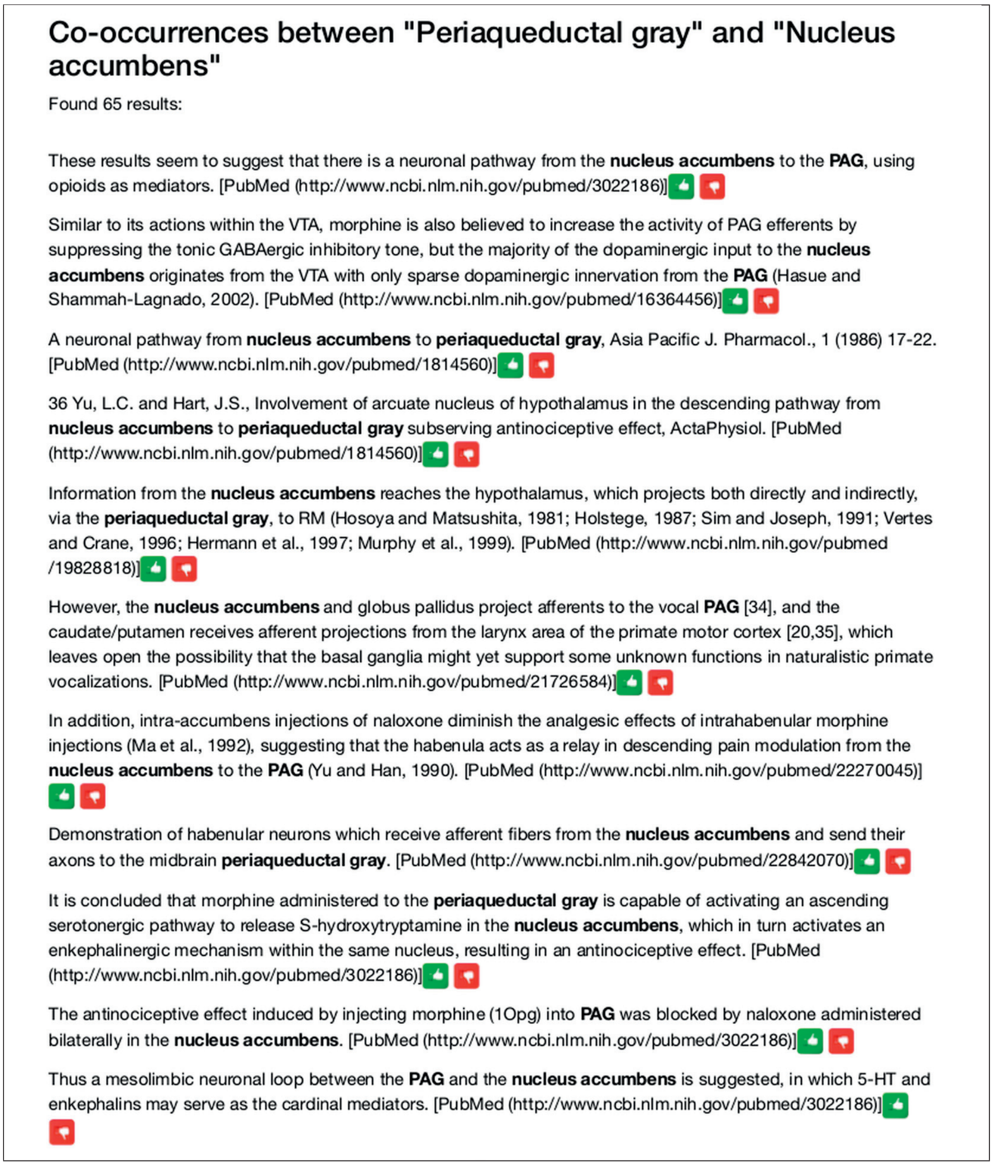
Regions connected to: Nucleus accumbens (db: 20140226_aba)

The table below lists brain regions for which *connectivity events* have been found in the literature. A connectivity event is a statement from a scientific article indicating that two brain regions are connected. The *score* represents the number of connectivity events, normalized by the confidence that each event has been extracted correctly (precision).

Click on a region to view the individual connectivity events:.

Region	Score
Ventral tegmental area	454
Caudoputamen	412
Cerebral cortex	295
Striatum-like amygdalar nuclei	175
Hippocampal region	122
Ammon's horn	93
Hippocampal formation	70
Pallidum	61
Midbrain	53
Subiculum	38
Thalamus	28
Hypothalamus	28
Periaqueductal gray	23
Olfactory tubercle	22
Basolateral amygdalar nucleus	19
fimbria	18
Nucleus raphe pontis	18
Entorhinal area	18
Dorsal nucleus raphe	13
Globus pallidus, external segment	12
medial forebrain bundle	11

Fig. 4.16: Listing of brain regions potentially connected to Nucleus accumbens, for which connectivity events have been found in the literature. The score represents the number of connectivity events, normalized by the confidence that each event has been extracted correctly (precision). All regions are linked to the corresponding detailed list of article sentences (see Figure 4.17). The corresponding url for that figure is http://connectivity-brainer.rhcloud.com/static/br/region.html?br=56&db=20140226_aba.



Co-occurrences between "Periaqueductal gray" and "Nucleus accumbens"

Found 65 results:

These results seem to suggest that there is a neuronal pathway from the **nucleus accumbens** to the **PAG**, using opioids as mediators. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/3022186)] 👍 👎

Similar to its actions within the VTA, morphine is also believed to increase the activity of PAG efferents by suppressing the tonic GABAergic inhibitory tone, but the majority of the dopaminergic input to the **nucleus accumbens** originates from the VTA with only sparse dopaminergic innervation from the **PAG** (Hasue and Shammah-Lagnado, 2002). [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/16364456)] 👍 👎

A neuronal pathway from **nucleus accumbens** to **periaqueductal gray**, Asia Pacific J. Pharmacol., 1 (1986) 17-22. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/1814560)] 👍 👎

36 Yu, L.C. and Hart, J.S., Involvement of arcuate nucleus of hypothalamus in the descending pathway from **nucleus accumbens** to **periaqueductal gray** subserving antinociceptive effect, ActaPhysiol. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/1814560)] 👍 👎

Information from the **nucleus accumbens** reaches the hypothalamus, which projects both directly and indirectly, via the **periaqueductal gray**, to RM (Hosoya and Matsushita, 1981; Holstege, 1987; Sim and Joseph, 1991; Vertes and Crane, 1996; Hermann et al., 1997; Murphy et al., 1999). [PubMed (http://www.ncbi.nlm.nih.gov/pubmed /19828818)] 👍 👎

However, the **nucleus accumbens** and globus pallidus project afferents to the vocal **PAG** [34], and the caudate/putamen receives afferent projections from the larynx area of the primate motor cortex [20,35], which leaves open the possibility that the basal ganglia might yet support some unknown functions in naturalistic primate vocalizations. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/21726584)] 👍 👎

In addition, intra-accumbens injections of naloxone diminish the analgesic effects of intrahabenular morphine injections (Ma et al., 1992), suggesting that the habenula acts as a relay in descending pain modulation from the **nucleus accumbens** to the **PAG** (Yu and Han, 1990). [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/22270045)] 👍 👎

Demonstration of habenular neurons which receive afferent fibers from the **nucleus accumbens** and send their axons to the midbrain **periaqueductal gray**. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/22842070)] 👍 👎

It is concluded that morphine administered to the **periaqueductal gray** is capable of activating an ascending serotonergic pathway to release S-hydroxytryptamine in the **nucleus accumbens**, which in turn activates an enkephalinergic mechanism within the same nucleus, resulting in an antinociceptive effect. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/3022186)] 👍 👎

The antinociceptive effect induced by injecting morphine (1Opg) into **PAG** was blocked by naloxone administered bilaterally in the **nucleus accumbens**. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/3022186)] 👍 👎

Thus a mesolimbic neuronal loop between the **PAG** and the **nucleus accumbens** is suggested, in which 5-HT and enkephalins may serve as the cardinal mediators. [PubMed (http://www.ncbi.nlm.nih.gov/pubmed/3022186)] 👍 👎

Fig. 4.17: Detailed list of sentences from neuroscientific articles, in this case between "Periaqueductal gray" and "Nucleus accumbens" (list truncated for readability). Each sentence is linked to the original article on PubMed. Additionally, the user has the ability to provide feedback: clicking on the red icon (thumbs down) will remove that sentence, and log it into the database. Similarly, clicking on the green icon (thumbs up) will confirm that sentence and log it in the database. The corresponding url for that figure is http://connectivity-brainer.rhcloud.com/static/br/details.html?br1=795&br2=56&db=20140226_aba.

4.2.4 Experiments: Automatic Target Validation for Tractography[27]

In this section, we propose to assess the previously-described text-mining (TM) models to automatically suggest targets from the neuroscientific literature for tractography studies. Target identification for tractography studies requires solid anatomical knowledge validated by an extensive literature review (LIT) across species for each seed structure to be studied. Manual LIT to identify targets for a given seed region is tedious and potentially subjective. Therefore, complementary approaches would be useful. We propose to use TM models to automatically suggest potential targets from the neuroscientific literature, so that they can be used for anatomical connection studies and more specifically for tractography. We applied TM models to three structures: two well studied structures, since validated deep brain stimulation targets, the *internal globus pallidus* and the *subthalamic nucleus* and, the *nucleus accumbens*, an exploratory target for treating psychiatric disorders. We performed a systematic review of the literature to document the projections of the three selected structures and compared it with the targets proposed by TM models, both in rat and primate (including human). We ran probabilistic tractography on the nucleus accumbens and compared the output with the results of the TM models and LIT. Overall, TM models could find three times as many targets as two man-weeks of curation could. The overall efficiency of the TM against LIT in our study was 98% recall (at 36% precision), meaning that over all the targets for the three selected seeds, only one target has been missed by TM. We demonstrate that connectivity for a structure of interest can be extracted from a very large amount of publications and abstracts. We believe this tool will be useful in helping the neuroscience community to facilitate connectivity studies of particular brain regions. The TM tools used for the study are part of the HBP Neuroinformatics Platform, publicly available at http://connectivity-brainer.rhcloud.com/.

Introduction

Determining the wiring diagram of the human brain is one of the greatest challenges in neurosciences [Spo11]. In initiatives such as the Human Connectome Project (HCP)[28], tractography occupies a key place in establishing the structural basis of the human connectome. Diffusion tensor imaging (DTI) has been introduced to document and measure in vivo anatomical connectivity between regions [JJB11]. DTI offers an overall view of brain anatomy, including the pattern and degree of connectivity between different regions, raising immediate hypothesis for brain function and for clinical applications such as deep brain stimulation (DBS). In combination with other technologies, DTI represents a powerful tool providing further insight on the networks influenced by neuromodulation [Bar+10] and consequently a better understanding of the mechanism of action and effects of DBS.

It is essential to have a thorough previous knowledge of the connections between the regions under investigation in order to validate the relevant fibers depicted via tractography, to pinpoint misses and for the choice of the method to be used. Target identification is a further crucial step for guided tractography from a seed region, to estimate the probability of their interconnection. Target identification requires solid anatomical knowledge documented by

[27] A version of this section has been published as [Vas+15] (co-authorship).

[28] `http://www.humanconnectome.org`

an extensive LIT across species for each seed structure to be studied. Existing literature in human is often conflicting and limited. Furthermore, experiments studying connectivity between individual brain regions are not reported in a normalized, structured and centralized repository, but published in plain text, scattered among individual scientific publications (see Section 4.2). Consequently, manual LIT to identify targets for a given seed region is tedious and potentially subjective. Therefore, complementary approaches would be very useful for the neuroscience community. We use the above-describe TM models to automatically generate potential targets from the neuroscientific literature, so that they can be used for anatomical connection studies and more specifically for tractography studies. To illustrate and evaluate the models, we applied TM models to three structures: two well studied structures, since validated DBS targets for movement disorders, the *internal globus pallidus* (GPi) and the *subthalamic nucleus* (STN) and, the *nucleus accumbens* (NAcc), exploratory target for treating psychiatric disorders. We performed a systematic LIT to document the projections of the three selected structures and compared it with the structures proposed by TM models, both in rat and primate (including human). To assess the results of the TM models, a comparison has been made between the two methods for the well-described GPi and STN. Finally, we ran probabilistic tractography on the NAcc and compared the output with the results of the TM models and LIT. The objective here is to document/support the validity of the TM models approach in helping to identify the targets to be explored for a given seed structure in (probabilistic) tractography projects.

Relevant publications were obtained using the PubMed database and references from the consulted articles. The PubMed database was manually searched for articles describing connections of the three nuclei, globus pallidus internus, subthalamic nucleus and nucleus accumbens. MeSH headings used were “globus pallidus”, “entopeduncular nucleus” (corresponding to the medial segment of the globus pallidus in rats), “subthalamic nucleus” and “nucleus accumbens”. We further searched for the following terms: “globus pallidus internus”, “pallidum internum”, “internal globus pallidus”, “globus pallidus pars interna” and “medial globus pallidus”. We combined them with the following MeSH headings for the studied species: “rats”, “primates” and “human” and with the following key words: “connections”, “projections”, “afferents” and “efferents”. Only articles written in English were reviewed. We used Terminologia Anatomica as reference for official nomenclature of the studied regions and structures.

Detailed description of the methods used for guided probabilistic tractography of NAcc can be found in [Vas+15].

Evaluation

The manual literature review (LIT) has been performed by two neuroscientists[29] and took approximately 5 working days for the three regions. The summary of the systematic review is presented in Table 4.12.

We compared and analyzed the results between TM and LIT for the three structures. Table 4.13 lists potential targets for the GPi and STN, as provided by the TM models. The potential targets are ranked by their decreasing score, the score representing the rounded number of connection mentions, normalized by the confidence[30] that each connection has been

[29] Dr. Laura Cif and Dr. Jocelyne Bloch

[30] Confidence (precision) has been evaluated for each extractor.

Tab. 4.12: Summary of the manual literature review. This review has been performed by two neuroscientists and took approximately 5 working days for the three regions. A detailed description of the three seed structures and their connections can be found in [Vas+15].

Afferents	Efferents
Globus Pallidus internus	
Subthalamic nucleus	Thalamus
Substantia nigra pars compacta	Lateral habenula
Ventral tegmental area	Substantia nigra
Neostriatum	Pedunculopontine nucleus
	Cerebral cortex (rat)
	Neostriatum
Subthalamic nucleus	
Primary motor cortex	Globus Pallidus internus
Supplementary motor area	Globus Pallidus externus
Frontal eye field	Substantia nigra pars compacta
Somatosensory cortex	Substantia nigra pars reticulata
Anterior cingulate	Ventral thalamic nuclei ipsilaterally
Globus Pallidus externus	Parafascicularis thalamic nucleus contralaterally (rat)
Substantia nigra pars compacta	Substantia innominata
Ventral tegmental area	Ventral pallidum
Dorsal raphe nucleus	Pedunculopontine nucleus
Pedunculopontine nucleus	Ipsilateral cortex (rat)
Centro-median/parafascicularis complex	Neostriatum (rat)
	Spinal cord (rat)
Nucleus Accumbens	
Orbitofrontal cortex	Ventral pallidum
Anterior cingulate	Substantia nigra pars compacta
Subgenual cortex	Substantia nigra pars reticulata
Pregenual cortex	Ventral tegmental area
Hippocampus	Hippocampus
Parahippocampal cortex	Caudate
Amygdala	Putamen
Substantia nigra pars compacta	Medio-dorsal thalamus
Ventral tegmental area	Cingulate gyrus
	Substantia innominata (rat)
	Lateral preoptic area (rat)
	Lateral hypothalamic area (rat)

extracted correctly. Therefore, a high score means that many articles have been found. We stress the fact that the frequency of a brain region connection reported in the scientific literature does not necessarily reflect the physiological intensity of a connection; the former reflecting the interest for the region.

GPi

For the GPi, all LIT targets have been correctly suggested by the TM algorithm using ABA lexicon, except for one, *ventral tegmental area* (VTA). However, VTA is correctly proposed while searching using ABA or braiNER for "Pallidum" or "Pallidum, ventral region" instead of globus pallidus, internal segment. TM proposes more targets for the GPi than LIT, including connections with *hypothalamus* (3 publications), *cerebellar nuclei* (2), *midbrain* (2), *parafascicular nucleus* (2) and *lateral preoptic area* (2). The majority of the suggested targets includes or belongs to targets resulted from LIT: *midbrain* includes SN; *parafascicular nucleus* relates to *thalamus*. However some of the targets proposed by TM were not found by LIT. Analyzing one such abstract suggested by TM, *globus pallidus* connexion to the *hypothalamus*, the *parafascicular nucleus* and the *lateral preoptic* area are explicitly reported. TM found confirmatory sentences for the previously mentioned connections: « On the other hand, the dense substance P-positive woolly-fiber plexus filling the internal pallidal segment (entopeduncular nucleus) expands medialward into the lateral hypothalamic region. » or « The entopeduncular nucleus invades the hypothalamus also with a loose plexus of enkephalin-positive woolly fibers » (Haber and Nauta, 1983). For connexions with the cerebellar nuclei, TM suggests papers that were not found by LIT, but these papers do not contain evidence of a connection. For illustration, we found three sentences that do not contain evidence of a connection with the *cerebellar nuclei* and all of them concern the cat. One example is « Seventy seven thalamic neurons in the VA-VL nuclear complex of the cat which projected to the anterior sigmoid gyrus (ASG) were studied extracellularly, and their responses to stimulation of both the cerebellar nuclei (CN) and the entopeduncular nucleus (ENT) were examined. » (Jinnai et al., 1987). This sentence is an example of a coordinating conjunction (e.g. « Region A and Region B were examined. »). It was suggested by the simplest TM model that is not capable of filtering out coordinating conjunctions (even though they very rarely represent a connection).

STN

For the STN, all the LIT targets have been found by TM, except for specific subdivisions of a given, such as *ipsilateral ventral thalamic nuclei*, *ventral pallidum* or the *anterior cingulate*. However, less specific regions (*thalamus, pallidum*) are correctly proposed. In addition, when using the machine learning NER, the connection between STN and the *ventral pallidum*, *anterior cingulate* and *ventral lateral thalamus* are found[31].

NAcc

For NAcc, Table 4.14 (left) lists brain regions for which connections have been found in the literature based on the ABAlex named entity recognizer. Additionally, Table 4.14 (right) also includes results from BraiNER (machine learning named entity recognizer). As discussed in Chapter 4.2.1, BraiNER is not constrained on a list of brain regions (like ABAlex) and is able to identify complex brain region names, even if they are not present in a lexicon. However, the regions returned by BraiNER have to be manually identified and curated[32].

[31] As shown in: http://connectivity-brainer.rhcloud.com/static/br/region.html?db=20140522_brainer&br=1922

[32] As shown in http://connectivity-brainer.rhcloud.com/static/br/region.html?db=20140522_brainer&br=912

Tab. 4.13: Brain regions for which connections have been found in the literature for the globus pallidus, internal segment and the subthalamic nucleus using text-mining models. All the results including suggested articles, nucleus and scores can be found in `http://connectivity-brainer.rhcloud.com`.

Subthalamic Nucleus		Globus Pallidus internus	
Region	Score	Region	Score
Globus pallidus, external segment	105	Caudoputamen	143
Caudoputamen	74	Globus pallidus, external segment	117
Cerebral cortex	43	Pallidum	23
Pallidum	34	Substantia nigra, reticular part	21
Pedunculopontine nucleus	16	Subthalamic nucleus	20
Thalamus	16	Lateral habenula	12
Globus pallidus, internal segment	15	Thalamus	10
Primary motor area	11	internal capsule	7
Somatomotor areas	9	Cerebral cortex	4
Substantia nigra, reticular part	9	Hypothalamus	3
Parafascicular nucleus	7	Substantia nigra, compact part	3
Zona incerta	5	Pedunculopontine nucleus	2
Substantia nigra, compact part	5	Cerebellar nuclei	2
Ventral tegmental area	3	Midbrain	2
Midbrain	2	Parafascicular nucleus	2
Lateral hypothalamic area	2	Lateral preoptic area	2
Hypothalamus	2	Cerebellum	1
Brain stem	2	Reticular nucleus of the th.	1
Pons	1	internal medullary lamina of the thalamus	1
internal medullary lamina of the th.	1	Striatum-like amygdalar nuclei	1
Red nucleus	1	Zona incerta	1
striatonigral pathway	1	stria medullaris	1
Isocortex	1	Fields of Forel	1
Dentate nucleus	1	Magnocellular nucleus	1
Substantia innominata	1	Central lateral nucleus of the th.	1
Bed nuclei of the stria terminalis	1	Claustrum	1
Islands of Calleja	1	Substantia innominata	1
Dorsal nucleus raphe	1	Brain stem	1
Cerebral nuclei	1	nigrostriatal tract	1
Olfactory tubercle	1	Interbrain	1
Auditory areas	1	optic tract	1
		Ammon’s horn	1

Tab. 4.14: Regions with highest scores for which connections have been found in the literature for the nucleus accumbens based on ABA and braiNER. Only regions with a score larger than 1 are displayed. The complete results are openly published and searchable at `http://connectivity-brainer.rhcloud.com`.

Nucleus Accumbens					
ABA		braiNER			
Region	Score	Region	Score	Region	Score
Ventral tegmental area	454	ventral tegmental area	238	amygdaloid nuclei	4
Caudoputamen	412	striatum	95	dentate gyrus	3
Cerebral cortex	295	prefrontal cortex	68	ventral pallidal	3
Striatum-like amygdalar nuclei	175	amygdala	54	cortical regions	3
Hippocampal region	122	medial prefrontal cortex	52	putamen	3
Ammon's horn	93	hippocampus	47	medial striatum	3
Hippocampal formation	70	hippocampal	41	telencephalon	3
Pallidum	61	basolateral amygdala	40	basolateral nucleus	3
Midbrain	53	caudate-putamen	39	basolateral nucleus of the amygdala	3
Subiculum	38	cortical	35	globus pallidus	3
Thalamus	28	mesolimbic	31	smith ad	3
Hypothalamus	28	hippocampal formation	29	limbic areas	3
Periaqueductal gray	23	ventral pallidum	26	caudatoputamen	3
Olfactory tubercle	22	ventral striatum	20	bla	3
Basolateral amygdalar nucleus	19	caudate putamen	16	prelimbic cortex	3
fimbria	18	thalamus	14	neocortex	2
Nucleus raphe pontis	18	neostriatum	13	ventral hippocampal	2
Entorhinal area	18	septum	13	substantia nigra pars compacta	2
Dorsal nucleus raphe	13	caudate nucleus	13	ventromedial striatum	2
Globus pallidus, external segment	12	mesencephalic	13	limbic system	2
medial forebrain bundle	11	amygdaloid	12	cingulate	2
Paraventricular nucleus of the thalamus	11	limbic	12	dorsomedial prefrontal cortex	2
Lateral preoptic area	9	dorsal raphe nucleus	11	nucleus raphe magnus	2
Nucleus of the solitary tract	8	paraventricular of the thalamus	11	anterior olfactory nucleus	2
stria terminalis	8	corpus striatum	11	ventral pallidus	2
Substantia innominata	7	forebrain	10	ventromedialmesencephalic tegmentum	2
Locus ceruleus	6	neocortical	9	anterior striatum	2
Orbital area	6	caudate	9	dorsomedial striatum	2
Interpeduncular nucleus	5	substantia nigra	9	accessory olfactory bulb	2
internal capsule	4	periaqueductal gray	9	bed nucleus of the stria terminalis	2
Infralimbic area	4	frontal cortex	8	paraventricular nucleus	2
Ventral posteromedial nucleus of the th.	4	subicular	8	anterior limbic cortex	2
Substantia nigra, compact part	4	anterior cingulate cortex	7	thalamic paraventricular nucleus	2
Prelimbic area	4	thalamic	7	tegmenti pedunculopontinus pars comp.	2
Lateral hypothalamic area	3	striatal	7	mesencephalic ventromedial tegmental	2
Nucleus raphe magnus	3	median raphe nucleus	7	ventral pallidal area	2
Median eminence	3	dorsal striatum	7	arcuate nucleus	2
Cerebellum	2	ventral subiculum	7	medial substantia nigra pars reticulata	2
Medial preoptic area	2	basal ganglia	6	striatal caudate putamen	2
Pedunculopontine nucleus	2	ventral tegmental	6	substantia innominata	2
Paraventricular hypothalamic nucleus	2	mesoaccumbens	6	lateral septum	2
fasciculus retroflexus	2	fimbria	6	ventral striatal	2
Arcuate hypothalamic nucleus	2	a10	6	anterolateral hypothalamus	2
Midbrain reticular nucleus, retrorubral area	2	subiculum	6	vta	2
Isocortex	2	paraventricular nucleus of the th.	5	rostral substantia innominata	2
Dorsomedial nucleus of the hypothalamus	2	ventral hippocampus	5	caudoputamen	2
Brain stem	2	lateral preoptic area	5	basolateral amygdaloid nucleus	2
Bed nuclei of the stria terminalis	2	limbic structures	5	lateral hypothalamic area	2
Substantia nigra, reticular part	2	olfactory tubercle	5	ventral tegmentum	2
Parafascicular nucleus	2	medial frontal cortex	5	limbic forebrain	2
Main olfactory bulb	2	ventral mesencephalic tegmentum	5	locus coeruleus	2
		lateral hypothalamus	5	anterior cingulate	2
		midbrain	5	nucleus accumbens septi	2
		accumbens	4	hypothalamus	2
		entorhinal cortex	4	parafascicular nuclei	2
		subpallidal areas	4	spinal	2
		orbitofrontal cortex	4	medial caudate-putamen	2
		ventral mesencephalon	4	medial substantia nigra	2
		prefrontal	4	thalamic nuclei	2
		cortex	4	mesolimbic dopaminergic pathway	2
		nucleus tractus solitarii	4	paraventricular nucleus of the hypoth.	2
		basal forebrain	4	neocortical fields	2

Tab. 4.15: Number of publications and percentage for which connections have been found for the 3 nuclei by species using text-mining.

	NAcc		STN		GPi	
Species	Publications	Percent	Publications	Percent	Publications	Percent
Rattus	1572	45.0	198	29.7	260	41.8
Mus	133	3.8	14	2.1	10	1.6
Homo Sapiens	83	2.3	34	5.1	13	2.0
Simiiformes	23	0.6	12	1.8	2	0.3
Chordata	72	2.0	12	1.8	15	2.4
Felidae	36	1.0	21	3.1	54	8.6
Canis	17	0.4	3	0.4	20	3.2
None	1550	44.4	372	55.8	247	39.7

Tab. 4.16: Overall performance of TM against LIT, in terms of number of publications.

	found by LIT	proposed by TM	missed by TM	precision	recall
Gpi	10	32	0	0.31	1.00
STN	23	31	1	0.76	0.95
Nacc	21	85	0	0.24	1.00
Overall	54	148	1	0.36	0.98

All the LIT targets, except the *subgenual* and *pregenual cortex*, have been found by the TM with the exact terminology. The two exceptions are explained by the fact that they are subdivisions of the anterior cingulate that figures as target.

overall TM performance

Overall, TM has a precision of 36%, meaning that it proposed three times as many targets as could be identified with LIT. Such a low precision is acceptable for the task at hand, since the priority is to suggest all targets (high recall), even if that requires manual curation of search results (since precision is only 36%) The overall recall of TM against LIT in our study was 98%, meaning that over all the targets for the three selected seeds, only one target have been missed by TM (Frontal eye field for the STN) (Table 4.15).

Species differentiation

Table 4.15 lists the number of publications found by text mining, ordered by species. Species were identified using Linnaeus, a machine-learning model to identify species in biomedical text and resolve it to the NCBI taxonomy [Ger+10]. One interesting observation is the difference between the number of studies on NAcc in rat and in primates, demonstrating the little available information on NAcc connectivity coming from studies in primates including human.

Probabilistic tractography

The targets for NAcc found during LIT and TM were used to perform tractography. We selected one human subject to illustrate the DTI results. Figure 4.18 shows the strength of connectivity of NAcc to its targets by depicting the number of voxels within the NAcc that has a probability superior to 1% to be connected to a specific target. Cortical targets such

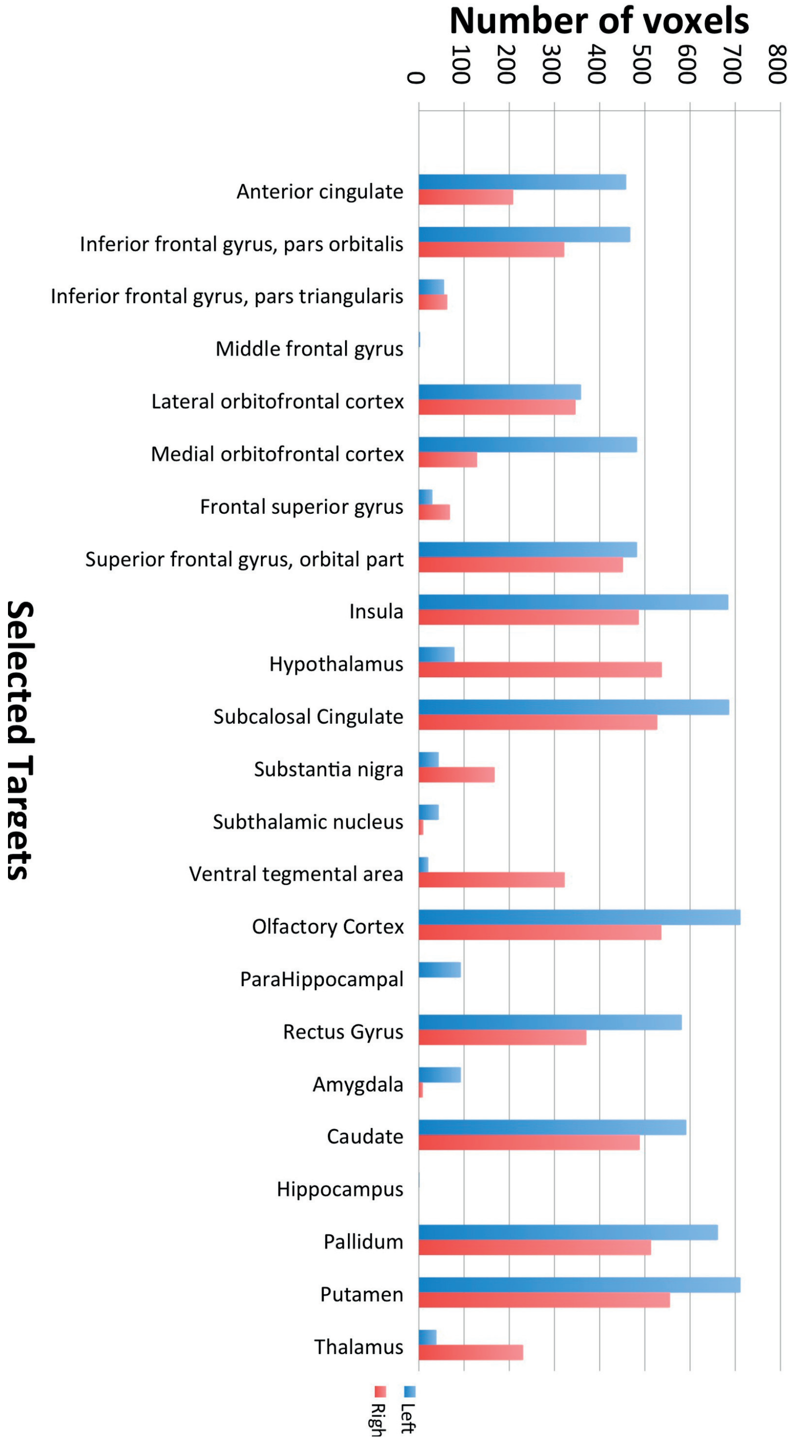


Fig. 4.18: Number of voxels within the nucleus accumbens that have a probability of more than 1% to be connected to a specific target in one subject (healthy control, right handed male, age 42). Left nucleus accumbens (in blue) has a total of 712 voxels and the right nucleus accumbens (in red) has a total of 559 voxels.

as the *anterior and subcalosal cingulate*, *medial* and *lateral orbitofrontal cortex*, *ventrolateral prefrontal cortex, insula, gyrus rectus, olfactory cortex* all exhibited connection to NAcc. Conversely, *hippocampus* and *amygdala* exhibited a lower probability of connection to NAcc than expected. *Hypothalamus* and *thalamus* and *basal ganglia* including *caudate, putamen* and *pallidum* as well as STN exhibited a strong probability of connexion. In agreement with previous knowledge, *midbrain dopaminergic structures*, SN and VTA exhibited high probability of connections with NAcc.

4.2.5 Discussion

Text mining models

We demonstrate that an exploitable brain region connectivity database can be extracted from a very large amount of scientific articles. Our models extract large amounts of connectivity data from unstructured text and compare favourably against in-vivo connectivity data. They provide a helpful tool for neuroscientists to facilitate the search and aggregation of brain connectivity data.

Our work builds on top of French *et al.* (2009, 2012 and 2012) and extends it in several aspects: Our connectivity extraction model uses a combination of three different extractors, including a novel rule-based extractor that achieves state-of-the-art precision. Models were applied to a comprehensive corpus of over 8 Billion words, consisting of all available PubMed abstracts and a very large number of full-text articles related to neuroscience. New model features and extraction filters were added to improve robustness on full-text extraction. Connectivity results are presented to neuroscientists in a interface to rapidly search, aggregate and evaluate connectivity results.

We highlight the fact that the presented text-mining models are not meant to replace manual and individual evaluation of the connectivity between two brain regions. The objective is to speed-up this evaluation and complement in-vivo or manually curated connectivity data. We assume that the extracted connectivity data will be reviewed and validated before being included in further analysis or models. Manual review is also mandatory since connection extractors have a very limited capacity to differentiate between hypothesized or contradictory connections, connections referred from another article, or connections supported by experimental data. Therefore, the efficient representation of connectivity data is important, so that domain experts can rapidly evaluate it.

A drawback of manual search (as it is most commonly performed for literature search) is the inability to provide *feedback* on search results. More that 3 million manual searches are performed *daily* on the PubMed web site[33]. Yet, a manual search performed by a researcher will neither improve future searches nor contribute to the building and curation of a structured knowledge base. In contrast, our database interface allows researcher to rate search results (collaborative filtering). Once enough feedback data is collected, the models can be retrained to achieve even higher performance.

The present study highlights the differences in complexity and performance between machine learning and rule-based approaches. The former delivers superior performance but requires a significantly more complex setup, in particular in terms of knowledge required (model and

[33] www.nlm.nih.gov/services/pubmed_searches.html

feature selection) and time for corpus annotation and model training. On the other hand, rule-based approaches are much simpler and require less time to develop. They are also less tightly bound to the domain they are applied to. For example, the *FILTERS* extractor (sect. 4.2.2) could be applied to relationship extraction between other entities (like neurons or proteins) without significant modification. However, the performance of rule-based approaches is significantly lower, especially in terms of recall.

Automatic target validation for tractography

In the previous section (4.2.4), we proposed to assess text-mining (TM) models to automatically suggest targets from the neuroscientific literature for tractography studies. Although current tractography methods have limitations, the ability to localize fiber bundles is of great help to understand connections and structural organization of the human brain. Anatomical knowledge can be used to impose constraint in the tract reconstruction, thereby effectively reducing the likelihood of the occurrence of erroneous results. Even if this approach is applied to anatomically well-documented tracts , it is essential to validate probabilistic results and in particular in DBS, to explore a specific seed by studying patterns of connectivity, sub-parcellation and confirmation of functional zones [Bar+10]. Brain structures like the *nucleus accumbens* (NAcc), are less documented in human. We believe that TM approaches can help neuroscientist to use the provided information to identify targets for tractography and document them in human.

Two well-established DBS targets for movement disorders have been studied (GPi and STN) and, NAcc, an exploratory DBS target for psychiatric disorders. The output of the TM method was compared with the output of a manual, systematic review of the literature (LIT) and the output of the probabilistic tractography using NAcc as seed structure. The concordance with data from manual search is significant and robust. The overall performance of the TM algorithm against LIT in our study was 98% recall, meaning that almost all regions found with LIT were also proposed by TM. In particular, when compared with the systematic search of the literature, for the "Globus pallidus, internal segment", all LIT targets but one (VTA) have been correctly suggested when using the restricted ABA lexicon. This missing target could be recovered when using the machine learning named entity recognizer (BraiNER). For the STN, all the targets identified by LIT have been found with TM, except for subsequent divisions of a given target, identified (again) when using braiNER. For NAcc, all the targets, except for the subdivisions of the anterior cingulate cortex have been identified. Overall and as expected, TM returns and proposes more targets than LIT, but also provides indication for the plausibility of a given connection between two regions. As an example, the connection between GPi and the Caudoputamen has a score of 143, making the connection highly probable. In contrast, only one single article has been found for the connection between GPi and Ammon's horn (Hippocampus).

The key advantage of TM is the ability to screen millions of documents and billion of words in a matter of hours. This way, the complete available biomedical literature can be processed and analyzed. Another advantage is the possibility to search within results, and order them according to relevance. It is also possible to provide feedback to the models and subsequently retrain them with that additional data in order to improve results. However, TM has several shortcomings and manual post-processing of results is mandatory. For example, complex sentences are tedious to analyze and often yield incorrect or empty results. In fact, one has to keep in mind that the estimated precision of the proposed target regions by TM is 36%. TM is not yet able to extract the directionality of the connection, nor metadata like neurotransmitter type or if the connection is inhibitory or excitatory. Additionally, TM lacks the ability to clearly differentiate between facts and hypothesis and is not yet able to trace the source of a connectivity statement (e.g. when an articles cites another reference).

When compared to the TM models, the manual, systematic search of the literature has the major advantage to select and interpret data in the light of the known anatomy, resulting in a deep and thorough analysis of the available literature. Researchers are able to filter, synthesize and aggregate very disparate and complex information into a consistent knowledge base. They are capable of interpreting every connectivity statement, of replacing it in its specific context (including experimental setting, field of expertize of the authors), and therefore of judging the exact pertinence of a connectivity statement. This detailed manual analysis comes at the cost of scaling, meaning that only a fraction of the published data will be considered.

Obviously, both approaches have compelling advantages. However, we found that the winning strategy is to combine and leverage the strength of both approaches. Indeed: TM can be deployed as a first step to screen and aggregate the scientific literature, capable of ingesting millions of documents. Thereafter comes the time for a manual and meticulous analysis and verification of the suggested connectivity statements, with the possibility to drill down to the original source (published article). The manual effort can be directed on intelligent tasks like validating and searching proposed connectivity statement, instead of their painstakingly identification from within millions of publications. Using this dual strategy (TM prior to LIT), it took less that two hours to have proposed a set of 25 potential targets for NAcc. In comparison, it took approximately a week for a user trained in neuroanatomy to conduct the isolated literature review of NAcc as presented in Section4.2.4. Therefore, the connectivity database significantly accelerates the manual search of metascale brain region connectivity, by providing a centralized repository of connectivity data for neuroscientists. Another advantage of this dual approach is the possibility for neuroscientist to collectively curate a knowledge base and therefore improve it.

Regarding the distinction of connectivity statements from different species: as demonstrated by the review for the NAcc, the majority of the available data comes from rodent studies. There is a striking need to disentangle human data from non-human primate data. Frequently, information reported in humans is inferred from animal studies without further notice. As provided by the results section, there is no sharp correspondence for the nomina between species for a given structure (e.g., globus pallidus, internal segment) rendering inferences from specie to another highly risky. Furthermore, the pattern of connectivity for a given structure may differ between species [Boh+09a]. Whether significant connections are reported between NAcc, hippocampus and amygdala through the available literature as identified via manual search and suggested by TM, the strength of connections between the aforementioned structures as output of the probabilistic tractography in healthy controls is not confirmatory of this result. However, there are many examples of fibre pathways that are reported in dissection and tracer studies that are lacking in diffusion tensor tractography studies, highlighting the importance of the selected tractography technique, its limitations and the potential role of the TM in validating connectivity information and support further investigations.

In the current study, we focused on the target identification using TM for tractography studies. Specific TM improvements are also required to improve tractography applications. For example, TM could be integrated in a 3D atlas to enhance the visualization and exploration of projections extracted from the literature, and to better evaluate topology, and speed up evaluation of results.

In conclusion, we demonstrate that connectivity for a structure of interest can be extracted from a very large amount of publications and abstracts. We believe this kind of approach will be useful in helping neuroscience community to facilitate connectivity studies of particular brain regions. The TM tools used for the present study are indeed part of the HBP Neuroinformatics Platform and are freely available for the neuroscience community.

4.3 neuroNER: finding neuron types and subtypes by identifying their properties[34]

lack of consistent terminologies for neurons

Since the earliest investigations by Santiago Ramòn y Cajal, neuroscientists have partitioned brain cells into types and subtypes according to their constituent features, including their morphology, electrophysiology, and molecular profiles (to name a few). However, neuroscientists regularly disagree on how neuron types should be described [MA13; DeF+13; Asc+08]. Thus the lack of consistent terminologies or nomenclatures for describing neuron types makes cross-lab study of neuron types immensely difficult.

compositional approach for analyzing neurons

In this section, we present a general approach for identifying and normalizing mentions of specific neuron types from the biomedical literature. Our method relies on identifying and analyzing each of the domain features used to annotate a specific neuron mention, like the morphological term "basket" or brain region "hippocampus". For example, a "pyramidal cell", is a neuron with a "pyramidal"-like morphology, whereas a "CA1 pyramidal cell" is a neuron with a pyramidal shape that has its soma in the pyramidal cell layer of hippocampal region CA1. By decoupling a neuron mention's identity into its specific compositional features, our method can identify specific neuron types even if they are not explicitly listed within a predefined neuron type lexicon, like NeuroLex [LM13] or the Cell Ontology [Bar+05].

results

We apply our method to two corpora of 13,293,649 PubMed abstracts and 630,216 full-text articles (see Section 3.1.4 for details about the corpora used). Our methods rely on Sherlok for information extraction and Elasticsearch for full-text search and analysis (see Section 3.3). We found over 500,000 unique neuron type mentions (see Table 4.18 below). A detailed analysis of these results is ongoing, including assessing whether neuron types with similar features in terms of their morphology, protein expression, or electrophysiological properties exist in different brain regions. To demonstrate the utility of our approach, we also apply our method towards cross-comparing the NeuroLex and Human Brain Project (BBP) cell type ontologies. The resulting code and models, including evaluation, is publicly available at http://github.com/renaud/neuroNER.

4.3.1 Introduction

The majority of what is known about the vast array of diverse neuron types is present in the neuroscience literature. For example, since the time of Cajal, neurons have been studied, defined, and partitioned into various types and subtypes along a number of dimensions, including their cellular location and morphology, electrophysiology, molecular profile, connectivity, and functional responses (among others).

inconsistent conventions

However, systematically accessing and utilizing this immense set of knowledge about neuron types is made difficult by the *inconsistent conventions* and nomenclatures about reporting information and data about neurons. Moreover, scientists regularly disagree on how

[34] This section results from a fruitful collaboration with Dr. Shreejoy Tripathy from the Centre for High-Throughput Biology at the University of British Columbia. It has been accepted as a conference abstract to the 2015 International Conference on Brain Informatics and Health, London. It will be further developed into a full-length paper following the thesis defense.

neuron types should be described, or what constitutes an adequate definition of a neuron type [MA13; DeF+13; Asc+08]. Thus, unlike domains like chemistry or genetics, where the objects of study, molecules and genetic loci, are clearly delineated and are referenceable using unique identifiers, the study of neuron types is inherently more nebulous.

complete listing of neuron types

While there has been recent work by neuroscience domain experts to enumerate the *complete listing of neuron types* (e.g., NeuroLex [LM13], BIRNLex [Bug+08], Cell Ontology [Bar+05] and Subcellular Neuroanatomy Ontology [Lar+07]), these listings are not completely comprehensive. Perhaps more critically, it is unclear whether such approaches capture the lexical and semantic richness of how neuroscientists refer to neuronal data in the scientific papers they publish. Moreover, for the value of a centralized listing of neuron types to be realized, neuroscientists and journal publishers must commit to using such a classification scheme when referring to their neuronal data, for example, through the use of unique unambiguous names or machine-readable identifiers. In the absence of such a formal classification scheme, downstream efforts by neuroscientists to normalize specific neuron instances to a neuron ontology or lexicon (e.g., ModelDB [Hin+04], Hippocampome [Ham+13], NeuroElectro [Tri+14]) inherently loses specificity. More problematically, such ambiguity in how neuron types are referred to makes it incredibly difficult to compare results on neuron function across labs, leading to needless replication of efforts and overall slowing of progress.

compositional approach

In contrast to this "top-down" approach for referring to data on neuron types, in this section we explore an alternative scheme for identifying and normalizing mentions of neuron types. Specifically, our idea is to think of neuron identity as *compositional*, or that neuron types are defined through conjunctions of modifying statements that span various domains, like morphology, electrophysiology or neurotransmitter released. For example, a "Neostriatum cholinergic cell", is a neuron that expresses "acetylcholine" and is located in the "Neostriatum". Such a neuron is semantically equivalent to "cholinergic neurons in the neostriatum" (see Table 4.19.). Similarly, "nest basket cell" is a subtype of a "basket" cell that further displays a "nest" morphology. These basic examples illustrate that neuron types can be naturally referred to at varying levels of resolution, given the specific modes of investigation of the author.

defining components

Thus, by decoupling a neuron's identity into its *defining components*, our hypothesis is that it may be possible to automatically identify each of these component features separately, and recognize a specific neuron type that an author is referring to, even if such a neuron type may not exist in a corresponding predefined lexicon. For example, the exact example of a "insulin-expressing CA1 pyramidal cell" does not currently exist in a corresponding ontology, but its meaning can be naturally expressed as a "Hippocampus CA1 pyramidal cell"[35] that further expresses the protein insulin.

A useful analogy for the compositional hypothesis is similar to that of the condensed (or semi-structural) chemical formulas. Such a scheme allows the decomposition of chemical formula of gamma-aminobutyric acid (i.e., GABA) into its constituents: $NH_2CH_2CH_2CH_2COOH$, or, a 4-carbon chain (i.e., butane) with an amino group at carbon position 1 and a carboxylic acid group at carbon position 4 (i.e., gamma). Similarly, a neuron mention like "Layer 2/3 pyramidal cell" can be decomposed into its constituents: `layer: 2/3`,

[35] Neurolex: `http://neurolex.org/wiki/Category:Hippocampus_CA1_pyramidal_cell`

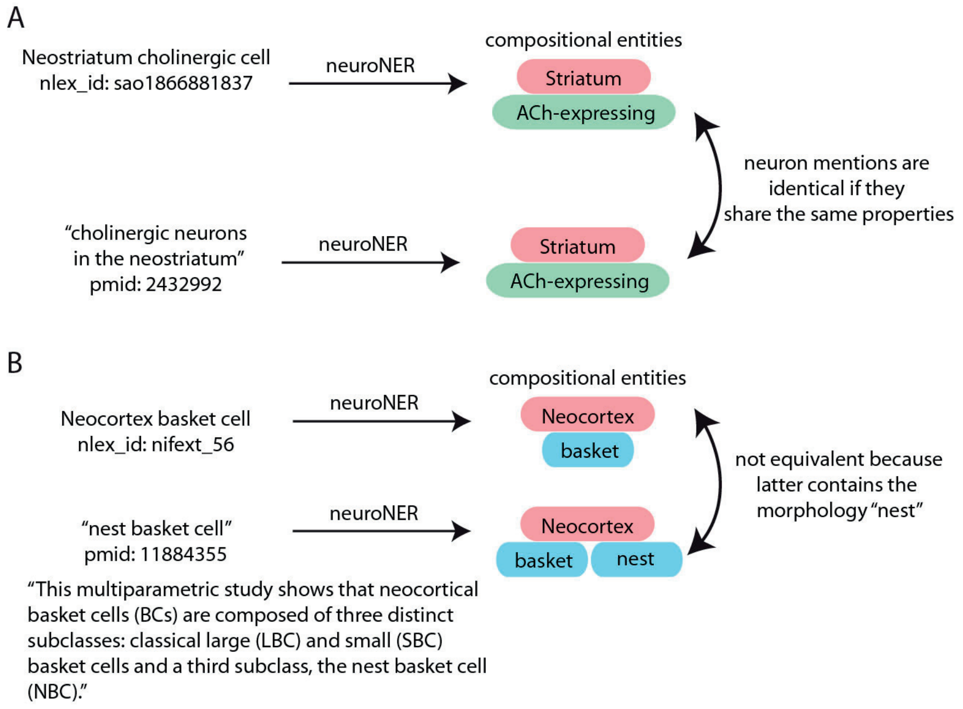


Fig. 4.19: The compositional hypothesis for neuron mentions. Our idea is to think of neuron identity as *compositional*, or that neuron types are conjunctions of modifying statements spanning various domains, like morphology or electrophysiology. In the first example, a "Neostriatum cholinergic cell" can be analyzed as a neuron expressing "acetylcholine" and located in the "Neostriatum". Such a neuron is semantically equivalent to "cholinergic neurons in the neostriatum".

`morphology:pyramidal`. As another example, the mention *thalamic Calb1-expressing neurons* can be decomposed into two properties: one related to the location of the neuron in the brain (in the *thalamus*) and the other regarding the genes expressed (*Calb1*, the gene for *Calbindin*). This decomposition allows comparing neurons at a more *semantic* level, e.g. the mention "calbindin D-28k-positive neurons in the reticular nucleus of the thalamus" is equivalent to the previous example, since *calbindin D-28k* refers to the protein coded by the gene *Calb1* and the *reticular nucleus of the thalamus* is a subregion of the *thalamus*. This kind of decomposition and normalization is essential for seamlessly comparing neuron mentions across laboratories and studies.

comparing neurons at the semantic level

Petilla nomenclature

Our approach draws much from previous efforts, namely, the *Petilla nomenclature* convention and later efforts by the Neuron Registry Task Force [Asc+08; Ham+12], where various aspects of neuron function were enumerated and a semi-standardized terminology was proposed to describe various aspects of cortical interneuron function. Our specific contribution to these earlier efforts made by neuron type experts is the explicit codification of the rules they proposed, as well as their large-scale application to the neuroscience literature.

Related to our efforts is *Virk*, an active learning system trained on identifying articles potentially contains information about neurons [Amb+13]. More specifically, it extracts (neuron type, relation, value) tuples (e.g., "CA1 pyramidal cell", "located in", "CA1 stratum oriens"). In contrast to neuroNER, Virk is a classifier that takes a decision on whether an article potentially contains information about neurons or not. It does not yet facilitate the localization of the information within the paper, nor does it perform the information extraction itself.

Applications

The following applications for neuroNER are envisioned:

- Locating instances of neuron subtypes in the literature (and in particular, rarely studied neuron subtypes). This would help prioritize articles for curation and incorporation to a structured database like NeuroMorpho [PA13] or NeuroElectro [Tri+14].
- Assisting authors submitting an article by automatically suggesting specific neuron types from a structured list to use as article keywords. This use case has been discussed with Elsevier.
- Allowing researchers to search through tagged neuron corpora. For example, given a query protein, like somatostatin, researchers can find all literature mentions where that protein has been found to be expressed in a neuron.
- Extract co-mentions of proteins and subcellular compartments (e.g., "synapsin is localized in CA1 pyramidal cell distal dendrite synapses").
- Extract co-mentions of neuron types and electrophysiological measurements.
- Applying an algorithmic tool for highlighting neuron mentions found in text can help a domain expert curate the neuron type to a specific instance from a predefined neuron lexicon.
- Count which neuron types are the most studied or least studied.

Category	Examples
brain region	CA1, pedunculus cerebri, cortical
species	rat, mouse, human, bovine
size	large, medium, narrow, giant
developmental stage	foetal, embryonic, post natal day 2
electrophysiology	depolarized, burst, fast-spiking
brain function	olfactory, primary motor, presynaptic
neuron morphology	bipolar, candelabrum, tufted, early bifurcating
neurotransmitter	glycinergic, dopaminergic, GABAergic
protein and genes	calbindin, mGluR1, Cck, NPY-expressing

Tab. 4.17: The 9 major categories of neuron properties.

4.3.2 Methods

Initial development

The first development iteration of neuroNER took approximately one week and consisted in structuring the specialized domain of neuron mentions. For this, an exploratory corpus of 5000 random sentences from the *Journal of Neuroscience* containing the term "neuron" or "cell" was created. It was used to structure the different neuron properties into 9 classes (see Table 4.17). Then, existing resources for each of the neuron properties were collected. For example, the Uniprot protein ontology [Con+08] was used to find proteins, together with a flat list of proteins and abbreviations frequently found in neuroscientific publications. An initial project was created in Sherlok and all initial resources were registered. A set of simple rules was created to match examples of neuron mentions acquired in the exploratory corpus.

In a second iteration, the system was applied on a larger corpus consisting of 100,000 abstracts that contained the MeSH term *neuroscience* selected at random. These results could be easily visualized and were analyzed for missing properties. In particular, adverbs and adjectives relative to brain regions were included, for example "hippocampal", "spinal" and "cortical".

Identifying Neuron Properties

The identification of neuron mentions in text is triggered by a short list of lexical variants of the words "neuron" and "cell". This list of trigger words is further augmented with specific cell type names like "astrocyte" or "microglia". Once a trigger word is identified, the algorithm searches for potential neuron properties that are located before and after the trigger word.

For each domain, extensive dictionaries of terms were manually created. For example, the dictionary for "neurotransmitter" contains an entry for the term "acetylcholine". That entry also contains two lexical variants ("cholinergic", "acetylcholinergic") and the abbreviation "ACh".

Many neuroner lexical resources evolved from simple flat lists of words related to a given

evolution of lexical resources

property, into structured OBO ontologies. For example, electrophysiological properties were at first identified with one single, large regular expression aiming at extracting the largest number of mentions. In subsequent developments, all possible electrophysiological properties were stored in a flat text file. Eventually, that list was refactored into two different OBO ontologies, one for electrophysiological trigger words like “spiking” or “bursting”, and one for electrophysiological properties like “fast” or “low threshold”. In many of the feature domains, we could find no suitable existing ontology, thus we chose to propose novel ontology terms and identifiers using “HBP” and the domain like “morphology” or “layer” as the namespace.

identification of proteins

Similarly, the *identification of proteins* was initially performed using diverse lists of proteins from existing ontologies and taxonomies. These resources were subsequently curated and condensed into an OBO ontology of approximately two thousand neuroscience-relevant genes and proteins, drawing from term lists internal to BBP or from projects like Hippocampome, which have enumerated listings of commonly used marker genes and proteins. To further expand these lists, we used synonym lists from NCBI and UniProt, using NCBI gene IDs as unique identifiers for each gene or protein. Here, we chose to employ a dictionary-based strategy for identifying genes and proteins as opposed to a machine learning NER approach because in initial tests, the machine learning NERs, like BANNER [LG+08] displayed too low precision in initial tests. However, in later iterations we are considering the application of a dual strategy where first, properties are identified with a manually-constructed dictionary containing high-frequency terms, and second, a machine-learning NER identifies less-frequent properties.

The “species” domain is identified using Linnaeus [Ger+10], a species NER that uses a dictionary-based approach and a set of heuristics to resolve ambiguous mentions about species. The Linnaeus model is able to resolve 97% of all mentions in PubMed Central full-text documents to unambiguous NCBI taxonomy identifiers.

An area of neuron properties which remains yet to be implemented in the neuroNER is the domain of neuroanatomical connectivity, including incorporating features like “L2/3-targeting” or “VTA-projecting”.

Normalization of property values

Normalization of property values is necessary to reduce lexical diversity and allow semantic matching. At the *lexical* level, normalization is handled through OBO synonyms or ROBO regular expressions (see Table 3.12 on page 42). At the *semantic* level, normalization is handled through explicitly defined ontological relationships. For example, neurotransmitters were labeled with their corresponding functional classes (inhibitory/excitatory) whenever appropriate. In the case of neocortical region, the OBO resource specifies whether a region is a subregion of another (e.g. “layer 5a” is a “layer 5” and “layer 2/3” is the union of layer 2 and 3. Similarly, species can be compared with a common ancestor up the NCBI taxonomy tree. Normalization at the *word* level could be handled with transformation rules, as in [FP12]. For example, the phrase “neocortical basket and Martinotti cells” could be split into two distinct mentions: “neocortical basket cells” and “neocortical Martinotti cells”. For some property classes like functions or morphologies, no normalization could be performed.

implicit correspondence

One important part of normalization is *implicit correspondence*. These represent implicit assumption, well understood by any SME, but not explicitly mentioned in the text. Examples of implicit correspondences include the following facts:

- Parvalbumin (PV) is expressed in GABAergic interneurons,
- PV is expressed in cerebellar Purkinje cells,
- Purkinje cells are only found in the Cerebellum.

These implicit correspondences could be extensively listed and handled during a post-processing step. Implicit correspondences can also consist of negative facts, e.g. that there are no Martinotti cells in the midbrain.

Semi-supervised Population of Lexical Resources

To improve neuroNER's recall, we collected unmatched words frequently occurring near a neuron mention. For example, during early developments, the neuron mention "cholinergic parasympathetic neurons" was not fully extracted: "cholinergic" was correctly identified, but not the region "postganglionic"[36]. Such a pattern of *"missed" properties* could be often observed. Missed properties were aggregated over a scale out on a large corpus. Only 1, 2 and 3 grams were considered. These n-grams were sorted by decreasing frequency in the corpus, and manually added to the resources by the SME if deemed relevant. This data-driven, semi-automatic algorithm for lexical resource enrichment was very helpful to improve neuroNER's recall. Moreover, it represents an alternative "bottom-up" approach towards populating ontological resources.

"missed" properties

4.3.3 Experiments and Results

Scale Out extraction

The neuroNER algorithm was applied to identify neuron mentions within a large corpus of 13,293,649 PubMed abstracts and 630,216 full-text articles, representing approximately 2 and 6B words, respectively (see Section 3.1.4 for details about corpus statistics). Table 4.18 lists the number of extracted properties. A detailed analysis and interpretation of these results is forthcoming, but these preliminary results illustrate that brain regions and morphology and considerably more likely to be used by authors to describe neurons than neurotransmitters, electrophysiology, or protein expression. This is in contrast to Petilla nomenclature, which suggests that each of these domains is a more-or-less equivalent way of describing neuron function.

Figure 4.20 illustrates the number of properties that were extracted per neuron. We observe that for the PubMed corpus, almost a third of all neuron mentions do not contain any property (i.e., an isolated instance of "neuron" or "cell", in the absence of an additional neuroNER-identified compositional feature term), while over half of all neuron mentions contain a single property. This relatively low number of properties can be explained by two primary reasons. First, many articles do not provide or require detailed descriptions of neurons or cells. Their granularity is above the level of neurons. Second, even for articles providing detailed descriptions of neurons, these often contain the generic words "neuron" or "cell" to refer to previously defined cells[37]. One such example can be found in [PMID

[36] In other words, one unknown word occurring between two known words

[37] In linguistics, this frequent phenomenon is known as *anaphora*. For example, the previous sentence used the anaphora "this".

Tab. 4.18: Extracted properties, grouped by property classes, based on the PubMed abstract corpus.

Category	Extracted properties
brainregion	525,934
function	308,652
morphology	167,513
species	147,259
size	63,983
neurotransmitter	52,834
developmental	52,492
orientation	10,717
protein and genes	9,287
layer	4,583
electrophysiology	2,940

22593736]: "*[...] we calculated the mean score for layer 2/3, layer 5, and layer 6 pyramidal neurons. [...] In most neurons we verified that [...]*". In the first instance, neurons are described with several layer properties, whereas the subsequent mention is a reference and thus does not repeat the layer information.

Figure 4.21 shows which combinations of neurons often co-occur in the same mention. For example, 30,929 neuron mentions contained both brain region and function properties. Some neuron mention exhibit a very large number of properties, e.g. "CA3 hippocampal and layer V motor cortical pyramidal neurons in adult male Wistar rats" [PMID 9766395] or "excitatory and inhibitory stellate cells in layer 4 of ferret visual cortex" [PMID 12631564] or "enkephalin positive striatopallidal MSNs (medium spiny projection neurons)" [PMID 17934457]. While these histograms illustrate a convenient way of summarizing the extracted information, we are currently exploring interactive ways of visualizing and drilling down into this data to further investigate the relationships contained within.

Evaluation of system recall and precision

To evaluate neuroNER's precision and recall, an evaluation corpus was created. This corpus consists of 97 full-text articles closely related to the research at BBP, so as to ensure that they contain complex forms of neuron mentions. Indeed, Figure 4.20 shows that the evaluation corpus contains neurons with more properties than the above PubMed corpus. From that evaluation corpus, 200 sentences containing neuron trigger words were manually evaluated[38]. These sentences contained in total 253 neuron properties, of which 207 could be recognized by neuroNER resulting in a *0.82 recall*. Two main types of error occur. In the first case, a property is not (yet) part of neuroNER and cannot be identified (e.g. "corticopontine", or "tripolar"). In the second case, no rules yet exist to match relatively rare patterns. For example, the phrase "interneurons recorded in layers 2/3 and

0.82 recall

[38] The corpus, details about articles selection and preprocessing are available at `https://github.com/renaud/neuroNER/tree/master/input/evaluation_corpus`.

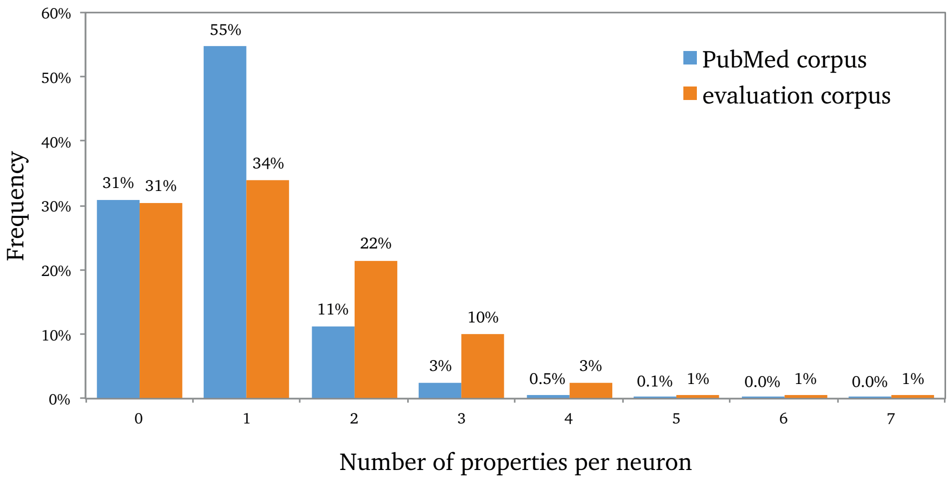


Fig. 4.20: Histogram of property counts.

5" could not be match since no rule of the form `{neuron} recorded in {layer}` exists yet.

Only 3 properties were incorrectly extracted, resulting in a *0.98 precision*. For example in the mention "fast-spiking-cell spike", the second instance of "spike" is a noun and was incorrectly extracted. In another case, "changes in morphology and input cell body doubled", a neuron mention was extracted while in fact the word "cell" does not represent an explicit neuron mention. Such errors could be solved by post-processing properties, and ensuring that they have the correct part-of-speech (e.g. adjectives or adverbs for neuron properties and nouns for neuron trigger words, like "cell" or "interneuron").

0.98 precision

Evaluation against NeuroLex and BBP cell ontology

As a concrete example of how the neuroNER can be used to facilitate normalization of neuron type mentions, we used the neuroNER to cross-compare the NeuroLex and BBP neuron lists[39]. Given that BBP entities refer to cortical cell types, we appended the brain region "Neocortex" to each of the BBP cells.

Figure 4.22 illustrates the correspondence between NeuroLex and the BBP cell ontology. The top example (A) shows the correspondence between HBP's "Layer II/III Pyramidal Cell" and NeuroLex's "Neocortex pyramidal layer 2-3 cell". Although both forms are lexically different, they are normalized to the exact same semantic representation. The middle example (B) illustrates a partial correspondence between HBP's "Layer V Nest Basket Cell" and NeuroLEX's "Neocortex basket cell", indicating that the former is a subtype, or more specific neuron instance than the latter. In the bottom example, a complex neuron mention from the literature [PMID 16369481] is only partially described by the BBP ontology, illustrating the need for a compositional feature approach.

[39] We downloaded the listing of vertebrate neuron types using the NeuroLex web portal and obtained an OBO file provided by Martin Telefont of the BBP cell types

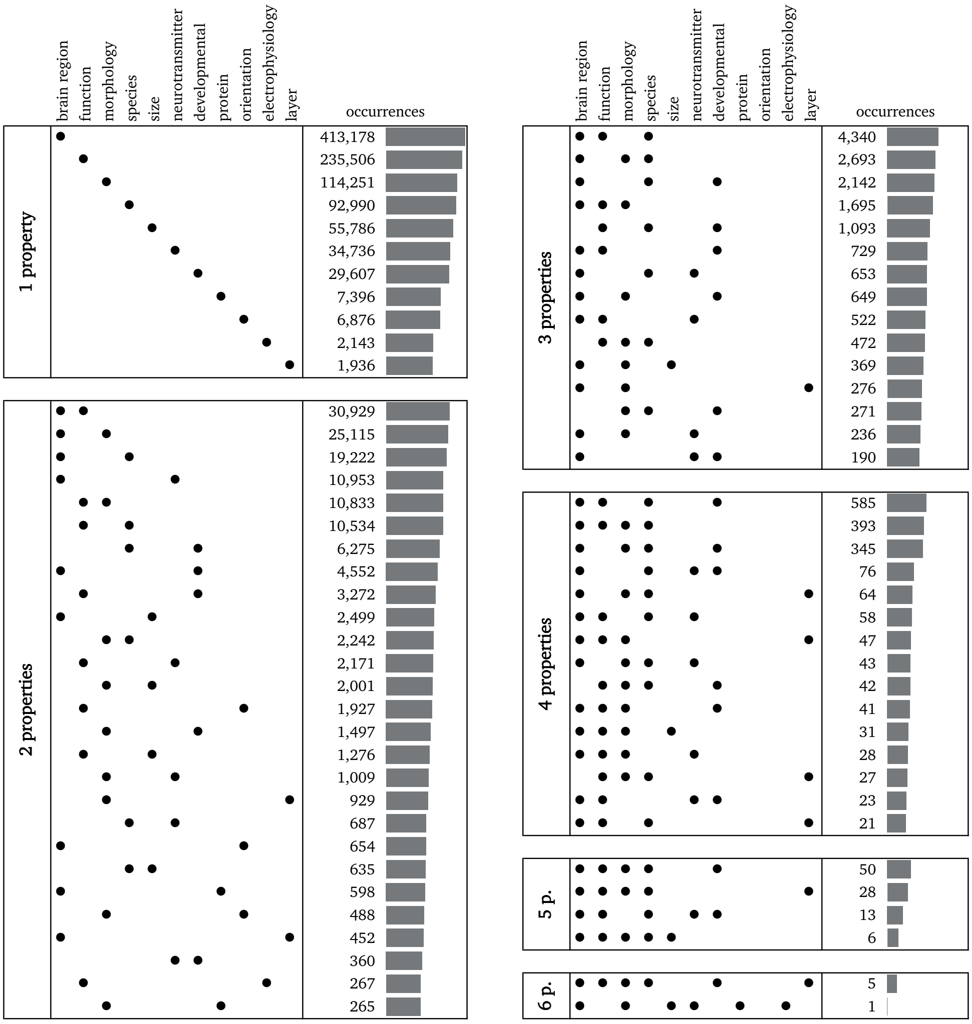


Fig. 4.21: Evaluation of inter-domain frequencies: how many neuron mentions were found with the listed combination of domains. Only combinations with a high frequency are shown. Histogram is in logarithmic scale.

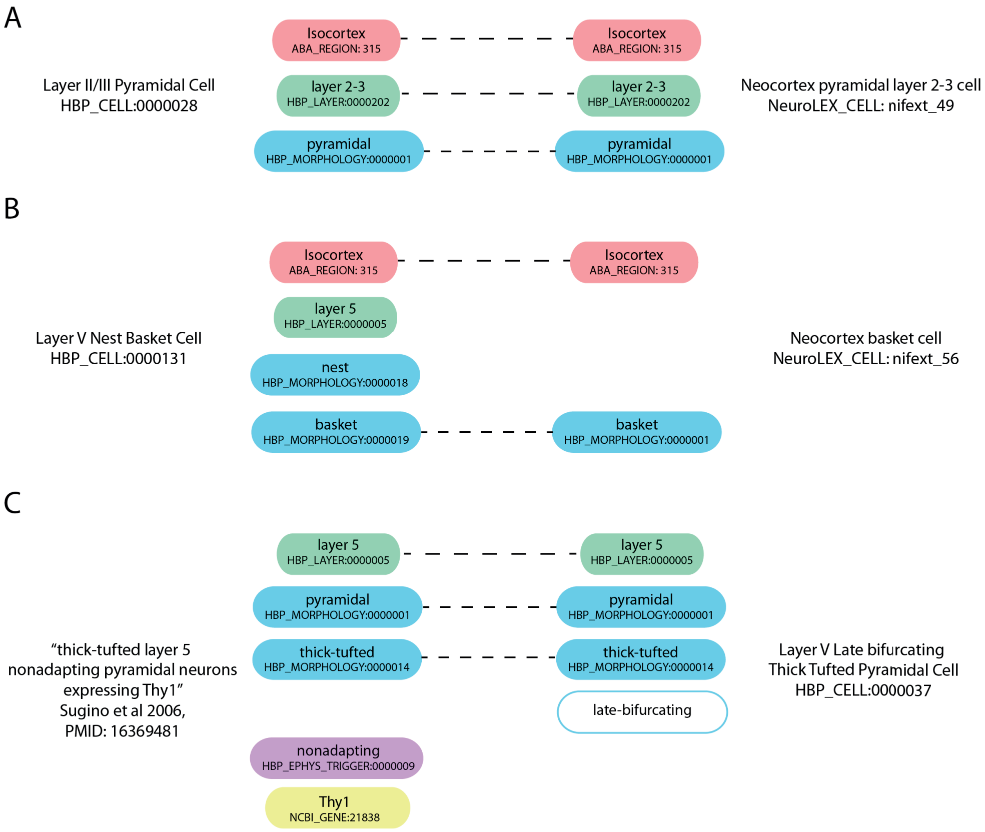


Fig. 4.22: Schematic of correspondence between NeuroLex and BBP (A and B). C is an example showing that even BBP cell types are insufficient to fully describe some of the things in the literature, illustrating the need for a compositional feature approach.

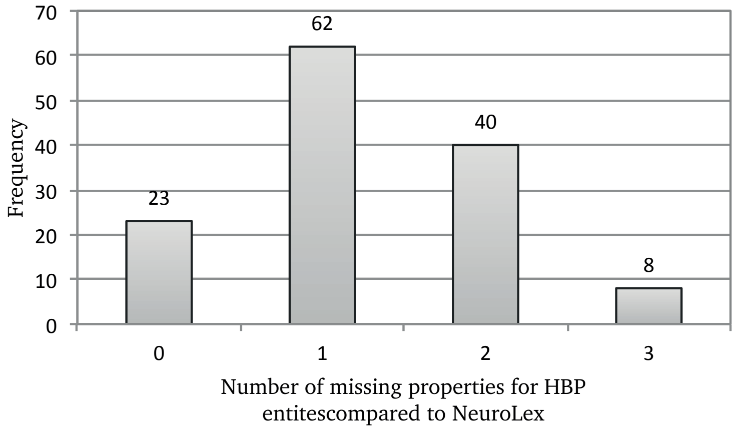


Fig. 4.23: Histogram contrasting the number of missing terms per BBP entity when compared to NeuroLex neuron types. The x axis is the number of missing neuroNER terms per BBP cell type, while the y axis is the frequency. Most BBP cell types are missing at least 1 property (62%) but usually not more than 2 when compared to NeuroLex.

The correspondence between NeuroLex and the BBP cell ontology was automatically evaluated, by using neuroNER to find the closest matching NeuroLex term for each BBP cell type entity. Figure 4.23 presents the number of missing terms per BBP entity when compared to NeuroLex neuron types. 23% of BBP entities can be fully normalized to NeuroLex, whereas 62% are missing at least 1 property. For most of these cases, the missing property is a layer term, where the BBP cell types are usually defined explicitly by layer whereas the NeuroLex cell types generally are not (e.g., "Layer 4 candelabrum cell" in BBP vs. "Neocortical candelabrum cell" in NeuroLex.). Additionally, BBP cell types tend to contain more specific morphology adjective terms, such as "nest" or "thick tufted".

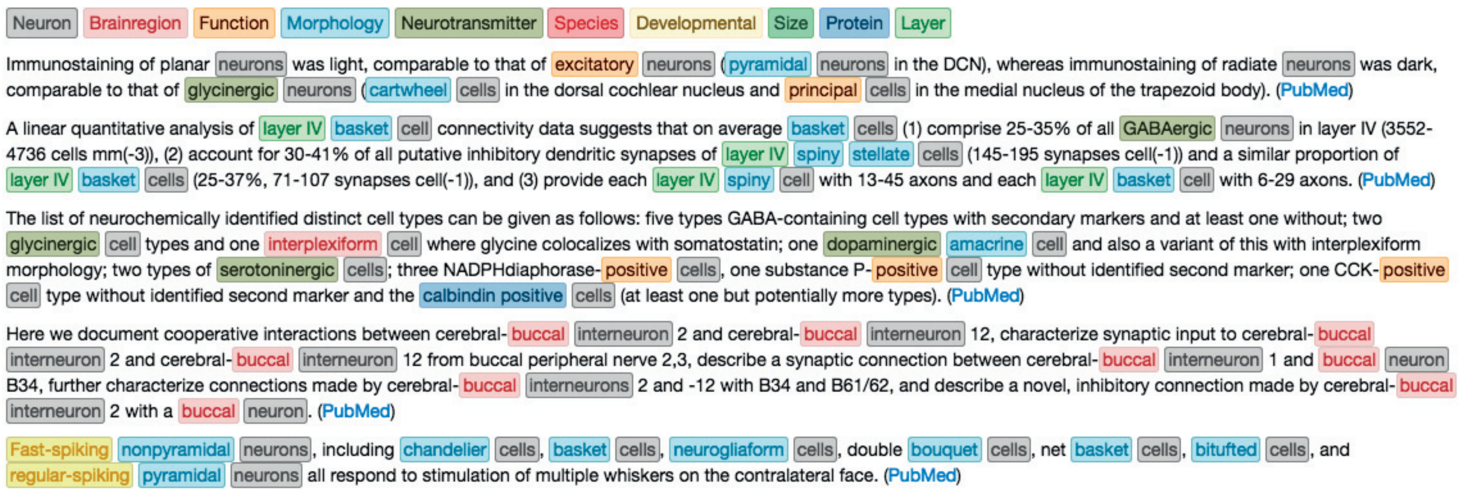
Overall, comparing the two ontologies indicates that the NeuroLex neuron lexicon tends to "lump" neuron types whereas the BBP tends to "split". However, the example in Figure 4.22 B shows that there are clear instances in the literature where even the BBP cell type ontology is insufficient to adequately represent an observed neuron instance. In general, this exercise illustrates that neuron types are defined hierarchically at varying levels of detail, and thus, that a compositional feature approach is needed for comparing and normalizing such neuron mentions.

neuroNER Visualization and Search Engine

The corpus, together with the extracted neuron mentions is stored in an Elasticsearch full-text index and can be easily interactively searched (Figure 4.24). The interface presents a useful method for quickly searching neuron mentions from the vast literature given an input query, like "somatostatin" or "layer V thick tufted".

Conclusions, ongoing improvements and future work

To date, the majority of the work for neuroNER has been in the development of algorithms for identifying neuron compositional features, including populating terms lists, like for



Neuron Brainregion Function Morphology Neurotransmitter Species Developmental Size Protein Layer

Immunostaining of planar neurons was light, comparable to that of excitatory neurons (pyramidal neurons in the DCN), whereas immunostaining of radiate neurons was dark, comparable to that of glycinergic neurons (cartwheel cells in the dorsal cochlear nucleus and principal cells in the medial nucleus of the trapezoid body). (PubMed)

A linear quantitative analysis of layer IV basket cell connectivity data suggests that on average basket cells (1) comprise 25-35% of all GABAergic neurons in layer IV (3552-4736 cells mm(-3)), (2) account for 30-41% of all putative inhibitory dendritic synapses of layer IV spiny stellate cells (145-195 synapses cell(-1)) and a similar proportion of layer IV basket cells (25-37%, 71-107 synapses cell(-1)), and (3) provide each layer IV spiny cell with 13-45 axons and each layer IV basket cell with 6-29 axons. (PubMed)

The list of neurochemically identified distinct cell types can be given as follows: five types GABA-containing cell types with secondary markers and at least one without; two glycinergic cell types and one interplexiform cell where glycine colocalizes with somatostatin; one dopaminergic amacrine cell and also a variant of this with interplexiform morphology; two types of serotoninergic cells; three NADPHdiaphorase-positive cells, one substance P-positive cell type without identified second marker; one CCK-positive cell type without identified second marker and the calbindin positive cells (at least one but potentially more types). (PubMed)

Here we document cooperative interactions between cerebral-buccal interneuron 2 and cerebral-buccal interneuron 12, characterize synaptic input to cerebral-buccal interneuron 2 and cerebral-buccal interneuron 12 from buccal peripheral nerve 2,3, describe a synaptic connection between cerebral-buccal interneuron 1 and buccal neuron B34, further characterize connections made by cerebral-buccal interneurons 2 and -12 with B34 and B61/62, and describe a novel, inhibitory connection made by cerebral-buccal interneuron 2 with a buccal neuron. (PubMed)

Fast-spiking nonpyramidal neurons, including chandelier cells, basket cells, neurogliaform cells, double bouquet cells, net basket cells, bitufted cells, and regular-spiking pyramidal neurons all respond to stimulation of multiple whiskers on the contralateral face. (PubMed)

Fig. 4.24: Output from neuroNER to automatically identify and normalize neuron type mentions from the neuroscientific literature. This output is composed of a random selection of sentences from PubMed abstracts related to neuroscience. Neuron properties are highlighted based on the category they belong to. The first line lists all different categories of neuron properties. In the last sentence, the words “double” and “net” are not yet identified and will be collected as “missing terms” to semi-automatically enrich the lexical resources (see Section 4.3.2).

morphological or electrophysiological concepts, and in developing lexical rules for identifying these terms in text. Given the .82 recall and .98 precision measures based on the evaluation corpus, we can consider the bulk of efforts in developing the neuroNER tool to be complete (though of course, further efforts can be taken to improve the recall, as outlined above).

We have applied the neuroNER towards locating and normalizing neuron mentions from an unprecedentedly large corpus of abstracts and neuroscience full texts and making these results easily searchable using a web-based interface. However, significant work remains in analyzing and summarizing these results and in interpreting their content. For example, by analyzing the neuroNER results, we should be able to provide estimates for which neurotransmitters, neuron morphologies, proteins, and electrophysiological phenotypes are present in which brain regions. Such in litero based measures can be validated against other databases, like gene expression from the Allen Brain Atlas [Lei+07].

As experimental methodology improvements (e.g. single-cell RNA sequencing [Zei+15]) promise to completely redefine neuron classification schemes, we feel that it is essential to first summarize decades worth of neuron study.

5 Synthesis

> ❞ ***Synergy:***
> *from Greek sunergos, 'working together'*
> *The interaction or cooperation of two or more organizations,*
> *substances or other agents to produce a combined effect*
> *greater than the sum of their separate effects*
>
> — **Oxford Dictionaries**

contributions

In this thesis, we introduced natural language processing (NLP) models and systems to mine the neuroscientific literature. In particular, we presented integrated NLP models designed to automatically extract *brain region connectivity* statements from very large corpora. We demonstrated the usefulness of these models through evaluations against in-vivo connectivity data and against manual review of the neuroscientific literature (Section 4.2). We also presented NLP model to perform automated *identification and normalization of neuron type mentions* in the neuroscientific literature. This kind of decomposition and normalization is essential for cross-laboratory studies, since neuroscience currently lacks consistent terminologies or nomenclatures for describing neuron types (Section 4.3). During the development of those two NLP models, we acknowledged the need for novel NLP approaches to rapidly develop custom text mining solutions. This led to the formalization of the *agile text mining*methodology to improve the communication and collaboration between subject matter experts and text miners (Section 1.3.3).

synergies

This thesis has been dedicated to researching and creating *synergies* between different fields:

context: Human Brain Project

It was conducted in the context of the *Human Brain Project*, an audacious project with the goal to create synergies between high-performance computing and neuroscience. If we dare an oversimplification we may observe, on one extreme, computer scientists creating brain-scale neural network simulations. However, their models of neurons and synapses bear very little resemblance to biological systems. On the other extreme, neuroscientists are capable of simulating neurons at an incredibly detailed level, but usually only for a handful number of neurons. The Human Brain Project is set to change this by providing the necessary computing power to build and simulate multi-level models of brain circuits and functions.

fields: NLP & neuroscience

This doctoral thesis lays at the boundaries between NLP and neuroscience and has been dedicated to creating useful and living links between these two disciplines. The objective was neither to develop novel machine learning algorithms nor to discover groundbreaking neuroscience principles. It was instead to push the state of the art in developing and applying NLP models and methodologies onto large corpora of neuroscientific literature, in order to extract knowledge and integrate it into neuroinformatics models.

At the NLP level, synergies between different methodical approaches were researched. In particular, the interaction between machine learning-based and lexical-based NERs. Also the combination of supervised and unsupervised methods was studied. NLP methods

We acknowledged the need for novel NLP methodologies to rapidly develop custom text mining solutions. This led to the formalization of the *agile text mining* methodology (Section 1.3.2) to create synergies between subject matter experts and text miners through facilitated communication and collaboration. Our tool, Sherlok (Section 3.3), supported us in successfully developing several agile text mining applications (ATMA). agile text-mining

Future directions of research include further improvements to neuroNER and analytics (Section 4.3). For example, the investigation into specifying 2 pivot properties (e.g. a morphology, “double bouquet” and a brain region “neocortex”) in order to identifying what other properties (e.g. electrophisiological or genetic types) can be induced from the literature. Another line of research could be to infer implicit correspondence from the literature (e.g. that parvalbumin neurons are implicitly fast-spiking). One more line of research could seek to confirm (or infirm) in-vivo data (e.g. in [Mar+15], Table 1: to validate that double bouquet expresses VIP but not NPY). future research

Further lines of research include the development of additional ATMAs to extract additional neuroscientific entities and relationships (e.g. synaptic connections, ion channels, and materials and methods sections, see Figure 1.1 page 4). Through these future applications, we hope to further refine our agile text mining methodology.

Appendix 6

6.1 Introduction to the MeSH Structure

Subject headings

Subject headings, *main headings* or *descriptors* are the principal components of the MeSH hierarchy. They are used to index articles from 5,400 of the world's leading biomedical journals for the PubMed database [@Nata]. Every descriptor is accompanied by a short definition, links to related descriptors and a set of synonyms or *entry terms*. They are generally updated on an annual basis to reflect changes in vocabulary and additions to the medical literature. There are 27 149 descriptors in the 2014 MeSH vocabulary, each one of which has a Unique Identifier assigned to it (starting with D and followed by 6 to 9 digits).

Structure

Descriptors are arranged in a twelve-level hierarchy where the most general terms as "Body Regions" or "Mental Disorders" appear in higher levels and more specific headings are found at deeper levels. The first level is composed of sixteen *categories* represented by a capital letter: "Anatomy" [A], "Organisms" [B], "Diseases" [C], "Chemicals and Drugs" [D], etc. Categories serve as an initial division for MeSH descriptors but are not descriptors themselves.

Each MeSH descriptor appears in at least one place in the hierarchy (sometimes referred as a "tree") and may appear in as many additional places as may be appropriate. Every appearance in the hierarchy is uniquely represented by a *tree number*. Therefore, each MeSH descriptor can have various tree numbers.

The MeSH structure can be viewed as a directed acyclic graph with nodes representing a single MeSH descriptor and directed edges representing the parent-child relation between descriptors. In this case, each graph node consists of a MeSH descriptor and a set of tree numbers, one for each place in the hierarchy where the descriptor appears.

For example, the MeSH descriptor "Body Regions" with Unique ID D001829 has tree number A01, showing that it is directly under the category Anatomy [A] while MeSH descriptor "Mouth" with Unique ID D009055 has tree numbers: A01.456.505.631 (Body Regions → Head → Face → Mouth), A03.556.500 (Digestive System → Gastrointestinal Tract → Mouth) and A14.549 (Stomatognathic System → Mouth). A graphical representation of the MeSH hierarchy for descriptor "Mouth" can be found at Figure 6.1.[1]

[1] The entire tree structure can be browsed online on `nlm.nih.gov/mesh/2014/mesh_browser/MeSHtree.html`.

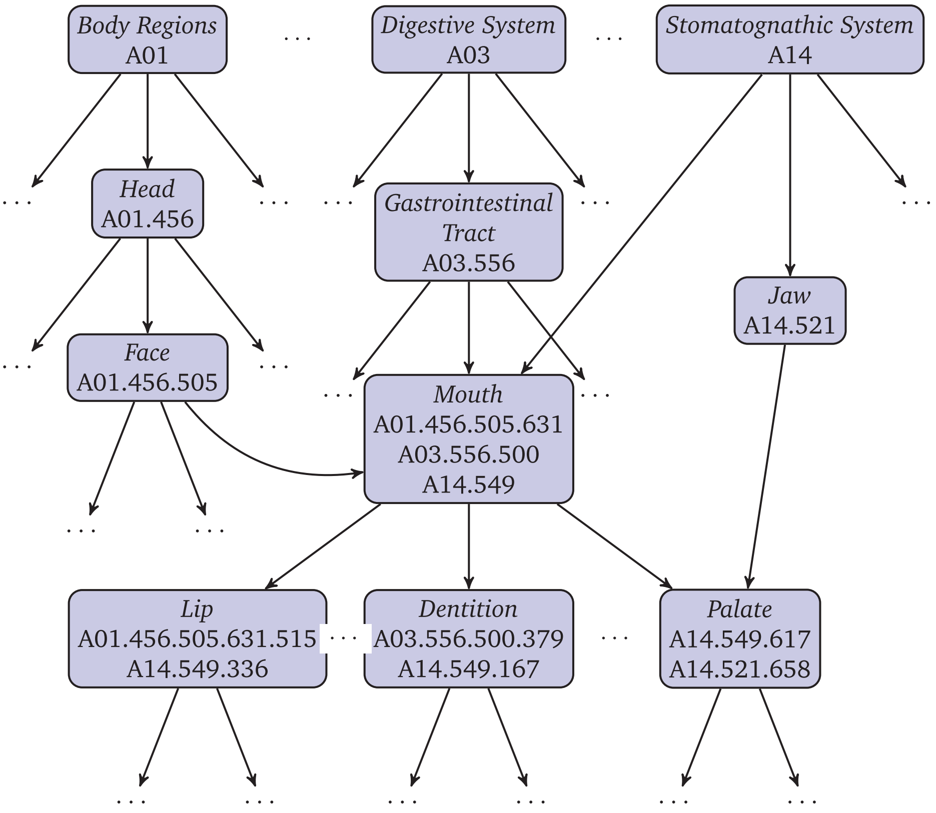


Fig. 6.1: Partial graph of the MeSH hierarchy for descriptor “Mouth”.

Qualifiers

Apart from the descriptors, which form the bulk of the MeSH catalog system, there are 83 *topical qualifiers*, also known as *subheadings*, used for indexing and cataloging in conjunction with descriptors. These subheadings are not required but are used to put emphasis in an specialization of the given descriptor. Furthermore, each descriptor can only be qualified by a small set of subheadings. Suitable subheadings for the descriptor “Endocrine Cells” are, for example, “EN Enzymology” or “ME Metabolism”.

It is worth noting that topical qualifiers have their own Unique Identifier (starting with Q and follow by 6 to 9 digits) and entry terms but do not belong to the hierarchy, therefore they have no tree number attached. As an example, qualifyer “ME Metabolism” has Unique ID Q000378 and entry terms “catabolysm”, “biodegradation”, among others.

MeSH usage in PubMed articles

For the interest of this study we note that almost every article in the PubMed database has been manually annotated by domain experts with a set of MeSH terms [@Natb].[2] Every publication is typically assigned between ten to twelve descriptors [@Natc] and these data can be easily accessed via the PubMed web interface. Moreover, a subset of these descriptors is signaled as representing the major focus of the article, these are called the *major descriptors* of the article.[3] A PubMed article has on average four major descriptors.[4]

As an example, article “The effect of mepyramine and 48/80 on the histamine content of pleural exudates in the rat” (PubMed ID 999393) is paired with the descriptors “Animals”, “Carrageenan”, “Exudates and Transudates”(m), “Histamine”(m), “Pleura”(m), “Pleurisy”, among others.

[2] This denominations is used to loosely refer to MeSH descriptors.

[3] Some literature uses the term “major topics”. We prefer to use the term “major descriptors” to avoid any confusion with LDA topics.

[4] Calculated on a sample of 100 000 articles.

References

[AC12] Kyle H Ambert and Aaron M Cohen. „Text-mining and neuroscience“. In: *Int. Rev. Neurobiol* 103 (2012), pp. 109–132 (cit. on p. 4).

[Aki+11] Huda Akil, Maryann E Martone, and David C Van Essen. „Challenges and opportunities in mining neuroscience data“. In: *Science (New York, NY)* 331.6018 (2011), p. 708 (cit. on p. 2).

[Amb+13] Kyle H Ambert, Aaron M Cohen, Gully APC Burns, Eilis Boudreau, and Kemal Sonmez. „Virk: an active learning-based system for bootstrapping knowledge base development in the neurosciences“. In: *Frontiers in neuroinformatics* 7 (2013) (cit. on p. 95).

[Amu+13] Katrin Amunts, Claude Lepage, Louis Borgeat, et al. „BigBrain: an ultrahigh-resolution 3D human brain model“. In: *Science* 340.6139 (2013), pp. 1472–1475 (cit. on p. 18).

[Ana+10] Sophia Ananiadou, Sampo Pyysalo, Jun'ichi Tsujii, and Douglas B Kell. „Event extraction for systems biology by text mining the literature“. In: *Trends in biotechnology* 28.7 (2010), pp. 381–390 (cit. on p. 9).

[And+00] Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinos, George Paliouras, and Constantine D Spyropoulos. „An evaluation of naive bayesian anti-spam filtering“. In: *arXiv* (2000) (cit. on p. 3).

[And07] R.K. Ando. „BioCreative II gene mention tagging system at IBM Watson“. In: *Proceedings of the Second BioCreative Challenge Evaluation Workshop*. Vol. 23. 2007, pp. 101 –103 (cit. on pp. 20, 35).

[Asc+08] Giorgio A Ascoli, Lidia Alonso-Nanclares, Stewart A Anderson, et al. „Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex“. In: *Nature Reviews Neuroscience* 9.7 (2008), pp. 557–568 (cit. on pp. 6, 92, 93, 95).

[Bad+12] Michael Bada, Miriam Eckert, Donald Evans, et al. „Concept annotation in the CRAFT corpus“. In: 13.1 (July 2012), p. 161 (cit. on pp. 16, 32).

[Bai+05] Amos Bairoch, Rolf Apweiler, Cathy H. Wu, et al. „The universal protein resource (UniProt)“. In: *Nucleic acids research* 33.suppl 1 (2005), D154–D159 (cit. on p. 34).

[Bak+12a] Anton Bakalov, Andrew McCallum, Hanna Wallach, and David Mimno. „Topic models for taxonomies“. In: *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*. ACM. 2012, pp. 237–240 (cit. on p. 47).

[Bak+12b] Rembrandt Bakker, Thomas Wachtler, and Markus Diesmann. „CoCoMac 2.0 and the future of tract-tracing databases“. In: *Frontiers in neuroinformatics* 6 (2012) (cit. on p. 18).

[Bar+05] Jonathan Bard, Seung Y Rhee, and Michael Ashburner. „An ontology for cell types“. In: *Genome Biology* 6.2 (2005) (cit. on pp. 16, 34, 92, 93).

[Bar+10] Garni Barkhoudarian, Tony Klochkov, Mark Sedrak, et al. „A role of diffusion tensor imaging in movement disorder surgery“. In: *Acta neurochirurgica* 152.12 (2010), pp. 2089–2095 (cit. on pp. 80, 89).

[BB01] M Banko and E Brill. „Mitigating the Paucity-of-Data Problem: Exploring the Effect of Training Corpus“. In: *Proceedings of HLT 2001*. 2001 (cit. on p. 8).

[BD03] Douglas Bowden and Mark Dubach. „NeuroNames 2002“. In: *Neuroinformatics* 1.1 (2003), pp. 43–59 (cit. on pp. 17, 34, 68).

[Bir06] Steven Bird. „NLTK: the natural language toolkit“. In: *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics. 2006, pp. 69–72 (cit. on p. 22).

[Bjo+11] J. Bjorne et al. „Extracting Complex Biological Events with Rich Graph-Based Feature Sets“. In: 27.4 (Nov. 2011), pp. 541 –557 (cit. on pp. 4, 21).

[Bjö+10] Jari Björne, Filip Ginter, Sampo Pyysalo, Jun'ichi Tsujii, and Tapio Salakoski. „Complex event extraction at PubMed scale“. In: *Bioinformatics* 26.12 (2010), pp. i382–i390 (cit. on p. 22).

[Ble+03] David M Blei, Andrew Y Ng, and Michael I Jordan. „Latent dirichlet allocation“. In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022 (cit. on p. 47).

[Ble12] David M Blei. „Probabilistic topic models“. In: *Communications of the ACM* 55.4 (2012), pp. 77–84 (cit. on p. 46).

[Boh+09a] Jason W Bohland, Caizhi Wu, Helen Barbas, et al. „A Proposal for a Coordinated Effort for the Determination of Brainwide Neuroanatomical Connectivity in Model Organisms at a Mesoscopic Scale“. In: *PLoS Computational Biology* 5.3 (2009) (cit. on p. 90).

[Boh+09b] Jason W Bohland, Hemant Bokil, Cara B Allen, and Partha P Mitra. „The brain atlas concordance problem: quantitative comparison of anatomical parcellations“. In: *PLoS One* 4.9 (2009), e7200 (cit. on p. 65).

[Bot+12] Mihail Bota, Hong-Wei Dong, and Larry W Swanson. „Combining collation and annotation efforts toward completion of the rat and mouse connectomes in BAMS“. In: *Frontiers in neuroinformatics* 6 (2012) (cit. on p. 18).

[BS08] M. Bota and L. W. Swanson. „BAMS neuroanatomical ontology: design and implementation“. In: *Frontiers in neuroinformatics* 2 (2008) (cit. on pp. 18, 64, 68).

[BS11] Quoc-Chinh Bui and Peter Sloot. „Extracting biological events from text using simple syntactic patterns“. In: *Proceedings of the BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics. 2011, pp. 143–146 (cit. on p. 21).

[Bug+08] W. J. Bug, G. A. Ascoli, J. S. Grethe, et al. „The NIFSTD and BIRNLex vocabularies: building comprehensive ontologies for neuroscience“. In: *Neuroinformatics* 6.3 (2008), 175–194 (cit. on pp. 17, 34, 93).

[Bui+13] Quoc-Chinh Bui, David Campos, Erik van Mulligen, and Jan Kors. „A fast rule-based approach for biomedical event extraction“. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013, pp. 104–108 (cit. on p. 21).

[Bun09a] Wray Buntine. *DCA 0.202: Discrete Component Analysis Software*. Software Documentation. Canberra, Australia: Statistical Machine Learning Group, NICTA, July 2009 (cit. on pp. 51, 52).

[Bun09b] Wray Buntine. *DCA 0.202 User Guide*. User Guide. Canberra, Australia: Statistical Machine Learning Group, NICTA, July 2009 (cit. on p. 52).

[Bun09c] Wray Buntine. „Estimating Likelihoods for Topic Models“. In: *Proceedings of the 1st Asian Conference on Machine Learning: Advances in Machine Learning*. ACML ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 51–64 (cit. on p. 51).

[Bur+08] G. Burns, D. Feng, and E. Hovy. „Intelligent Approaches to Mining the Primary Research Literature: Techniques, Systems, and Examples“. In: *Computational Intelligence in Medical Informatics*. Vol. 85. Springer Berlin Heidelberg, 2008, pp. 17–50 (cit. on pp. 4, 16, 20).

[Cam+12] David Campos, José Luís Oliveira, and Sérgio Matos. *Biomedical named entity recognition: a survey of machine-learning tools*. INTECH Open Access Publisher, 2012 (cit. on p. 3).

[Cam+13] David Campos, Sérgio Matos, and José Luís Oliveira. „Gimli: open source and high-performance biomedical name recognition“. In: *BMC Bioinformatics* 14.1 (Feb. 2013), p. 54 (cit. on pp. 20, 35).

[Cha13] Jean-Cédric Chappelier. „Topic-based generative models for text information access“. In: *Textual Information Access: Statistical Models*. Ed. by Eric Gaussier and François Yvon. John Wiley & Sons, 2013, pp. 129–178 (cit. on p. 47).

[Cob14] Erick Cobos. *Topic Models applied to Neuroscientific Literature*. Semester Project Report. Lausanne, Switzerland: Ecole Polytechnique Fédérale de Lausanne, School of Computer and Communication Sciences, 2014 (cit. on p. 46).

[Con+08] UniProt Consortium et al. „The universal protein resource (UniProt)“. In: *Nucleic acids research* 36.suppl 1 (2008), pp. D190–D195 (cit. on pp. 6, 96).

[Cor+15] James Cormack, Chinmoy Nath, David Milward, Kalpana Raja, and Siddhartha R Jonnalagadda. „Agile text mining for the 2014 i2b2/UTHealth Cardiac risk factors challenge“. In: *Journal of biomedical informatics* (2015) (cit. on p. 21).

[Cun+11] Hamish Cunningham, Diana Maynard, and Kalina Bontcheva. *Text processing with gate*. Gateway Press CA, 2011 (cit. on p. 22).

[D’A+13] Egidio D’Angelo, Sergio Solinas, Jesus Garrido, et al. „Realistic modeling of neurons and networks: towards brain simulation“. In: *Functional neurology* 28.3 (2013), p. 153 (cit. on p. 2).

[DCG09] Richard Eckart De Castilho and Iryna Gurevych. „DKPro-UGD: A Flexible Data-Cleansing Approach to Processing User-Generated Discourse“. In: *Onlineproceedings of the First French-speaking meeting around the framework Apache UIMA, LINA CNRS UMR*. 2009 (cit. on p. 33).

[DeF+13] Javier DeFelipe, Pedro L López-Cruz, Ruth Benavides-Piccione, et al. „New insights into the classification and nomenclature of cortical GABAergic interneurons“. In: *Nature Reviews Neuroscience* 14.3 (2013), pp. 202–216 (cit. on pp. 92, 93).

[Dem+13] Janez Demšar, Tomaž Curk, Aleš Erjavec, et al. „Orange: data mining toolbox in Python“. In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 2349–2353 (cit. on p. 22).

[Dim+11] E. Dimmer et al. „The UniProt-GO Annotation database in 2011“. In: *Nucl. Acids Res.* (Nov. 2011) (cit. on p. 16).

[DM+06] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. „Generating typed dependency parses from phrase structure parses“. In: *Proceedings of LREC*. Vol. 6. 2006. 2006, pp. 449–454 (cit. on p. 19).

[Doğ+14] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. „NCBI disease corpus: a resource for disease name recognition and concept normalization“. In: *Journal of biomedical informatics* 47 (2014), pp. 1–10 (cit. on p. 16).

[Dru11] G. Druck. „Generalized Expectation Criteria for Lightly Supervised Learning“. PhD thesis. University of Massachusetts Amherst, 2011 (cit. on p. 22).

[Fel10] C. Fellbaum. „WordNet“. In: *Theory and Applications of Ontology: Computer Applications* (2010), 231–243 (cit. on p. 34).

[Fer+10] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, et al. „Building Watson: An overview of the DeepQA project“. In: *AI magazine* 31.3 (2010), pp. 59–79 (cit. on pp. 3, 23).

[Fer+13] David A Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, and Erik T Mueller. „Watson: beyond jeopardy!“ In: *Artif. Intell.* 199 (2013), pp. 93–105 (cit. on p. 23).

[FL04] David Ferrucci and Adam Lally. „UIMA: an architectural approach to unstructured information processing in the corporate research environment“. In: *Natural Language Engineering* 10.3-4 (2004), pp. 327–348 (cit. on pp. 23, 33).

[FP12] L. French and P. Pavlidis. „Using text mining to link journal articles to neuroanatomical databases“. In: *The Journal of Comparative Neurology* 520.8 (2012), 1772–1783 (cit. on pp. 66, 67, 88, 97).

[Fre+09] L. French, Suzanne Lane, Lydia Xu, and Paul Pavlidis. „Automated Recognition of Brain Region Mentions in Neuroscience Literature“. In: *Front Neuroinformatics* 3 (Sept. 2009) (cit. on pp. 34, 67–70, 88).

[Fre+12] L. French, S. Lane, L. Xu, et al. „Application and evaluation of automated methods to extract neuroanatomical connectivity statements from free text“. In: *Bioinformatics* 28.22 (2012), 2963–2970 (cit. on pp. 20, 66, 67, 69, 71, 88).

[Fre+15] Leon French, Po Liu, Olivia Marais, et al. „Text mining for neuroanatomy using WhiteText with an updated corpus and a new web application“. In: *Frontiers in Neuroinformatics* 9 (2015), p. 13 (cit. on pp. 16, 20, 71).

[FT02] Roy T Fielding and Richard N Taylor. „Principled design of the modern Web architecture“. In: *ACM Transactions on Internet Technology (TOIT)* 2.2 (2002), pp. 115–150 (cit. on pp. 25, 41).

[GB08] C. Gasperin and T. Briscoe. „Statistical anaphora resolution in biomedical texts“. In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. 2008, 257 – 264 (cit. on p. 18).

[Ger+10] M. Gerner, G. Nenadic, and C. Bergman. „Linnaeus: A species name identification system for biomedical literature“. In: *BMC Bioinformatics* 11.1 (2010), p. 85 (cit. on pp. 20, 35, 69, 86, 97).

[Ger+12] Martin Gerner, Farzaneh Sarafraz, Casey M Bergman, and Goran Nenadic. „BioContext: an integrated text mining system for large-scale extraction and contextualization of biomolecular events“. In: *Bioinformatics* 28.16 (2012), pp. 2154–2161 (cit. on p. 22).

[Giu+06] C. Giuliano, A. Lavelli, and L. Romano. „Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature.“ In: *EACL*. Vol. 2006. 2006, 98–113 (cit. on p. 69).

[GS04] Thomas L Griffiths and Mark Steyvers. „Finding scientific topics“. In: *Proceedings of the National Academy of Sciences* 101.suppl 1 (2004), pp. 5228–5235 (cit. on p. 47).

[Hah+07] U. Hahn, Ekaterina Buyko, Katrin Tomanek, et al. *An Annotation Type System for a Data-Driven NLP Pipeline*. 2007 (cit. on p. 34).

[Hah+08] Udo Hahn, Ekaterina Buyko, Rico Landefeld, et al. „An overview of JCoRe, the JULIE lab UIMA component repository“. In: *Proceedings of the LREC*. Vol. 8. 2008, 1–7 (cit. on p. 33).

[Ham+12] David J. Hamilton, Gordon M. Shepherd, Maryann E. Martone, and Giorgio A. Ascoli. „An ontological approach to describing neurons and their relationships“. In: *Front. Neuroinform.* 6 (2012), p. 15 (cit. on p. 95).

[Ham+13] DJ Hamilton, DW Wheeler, C White, et al. „Machine-readable representations of hippocampal neuron properties to facilitate investigative analytics“. In: *Front. Neuroinform. Conference Abstract: Neuroinformatics 2013*. 2013 (cit. on p. 93).

[HD14] Orit Hazzan and Yael Dubinsky. „The Agile Manifesto“. In: *Agile Anywhere*. Springer, 2014, pp. 9–14 (cit. on p. 11).

[Hin+04] Michael L Hines, Thomas Morse, Michele Migliore, Nicholas T Carnevale, and Gordon M Shepherd. „ModelDB: a database to support computational neuroscience“. In: *Journal of computational neuroscience* 17.1 (2004), pp. 7–11 (cit. on p. 93).

[Hof+00] P. R. Hof, W. G. Young, F. E. Bloom, P. V. Belichenko, and M. R. Celio. *Mouse Brains. Comparative Cytoarchitectonic Atlas of the C57BL/6 and 129/SV*. Elsevier Science, Amsterdam, 2000 (cit. on pp. 18, 68).

[Hof+10] Matthew D. Hoffman, David M. Blei, and Francis Bach. „Online learning for latent dirichlet allocation“. In: *In NIPS*. 2010 (cit. on pp. 52, 53).

[Ima+11] F. T. Imam, S. D. Larson, J. S. Grethe, et al. „NIFSTD and NeuroLex: A Comprehensive Neuroscience Ontology Development Based on Multiple Biomedical Ontologies and Community Involvement“. In: (2011) (cit. on pp. 17, 34).

[Ima+12] Fahim T Imam, Stephen Larson, Jeffery S Grethe, et al. „Development and use of ontologies inside the neuroscience information framework: a practical approach“. In: *Frontiers in genetics* 3 (2012), p. 111 (cit. on p. 17).

[Jes+11] D. Jessop et al. „OSCAR4: a flexible architecture for chemical text-mining“. In: *Journal of Cheminformatics* 3.1 (Oct. 2011), p. 41 (cit. on pp. 20, 35).

[JG10] S. Jonnalagadda and G. Gonzalez. „Sentence simplification aids protein-protein interaction extraction“. In: (2010) (cit. on p. 21).

[JJB11] Saad Jbabdi and Heidi Johansen-Berg. „Tractography: where do we go from here?“ In: *Brain connectivity* 1.3 (2011), pp. 169–183 (cit. on p. 80).

[Kae+14] Suwisa Kaewphan, Kai Hakala, and Filip Ginter. „UTU: Disease Mention Recognition and Normalization with CRFs and Vector Space Representations“. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, 2014, pp. 807–811 (cit. on pp. 3, 20).

[Kan+11a] N. Kang, E. van Mulligen, and J. Kors. „Comparing and combining chunkers of biomedical text“. In: 44.2 (Apr. 2011), pp. 354 –360 (cit. on p. 19).

[Kan+11b] Yoshinobu Kano, Jari Björne, Filip Ginter, et al. „U-Compare bio-event meta-service: compatible BioNLP event extraction services“. In: *BMC bioinformatics* 12.1 (2011), p. 481 (cit. on p. 22).

[KB09] Halil Kilicoglu and Sabine Bergler. „Syntactic dependency based heuristics for biological event extraction“. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics. 2009, pp. 119–127 (cit. on p. 21).

[Kim+03] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. „GENIA corpus–a semantically annotated corpus for bio-textmining“. In: *Bioinformatics* 19 (July 2003), pp. i180–i182 (cit. on pp. 4, 34).

[Kim+12] Jin-Dong Kim, Ngan Nguyen, Yue Wang, et al. „The genia event and protein coreference tasks of the BioNLP shared task 2011“. In: *BMC bioinformatics* 13.Suppl 11 (2012), S1 (cit. on pp. 16, 21).

[Kim+13] Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. „The genia event extraction shared task, 2013 edition-overview“. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013, pp. 8–15 (cit. on p. 21).

[Klu+09] Peter Kluegl, Martin Atzmueller, and Frank Puppe. „Textmarker: A tool for rule-based information extraction“. In: *Proceedings of the Biennial GSCL Conference*. 2009, pp. 233–240 (cit. on pp. 23, 40, 69).

[Klu+14a] P. Kluegl, M. Toepfer, P.-D. Beck, G. Fette, and F. Puppe. „UIMA ruta: Rapid development of rule-based information extraction applications“. In: *Natural Language Engineering* (2014), pp. 1–40 (cit. on pp. 23, 40).

[Klu+14b] P. Kluegl, M. Toepfer, P.-D. Beck, G. Fette, and F. Puppe. „UIMA Ruta: Rapid development of rule-based information extraction applications“. In: *Natural Language Engineering* (2014) (cit. on p. 69).

[Kon+13] Georgios Kontonatsios, Ioannis Korkontzelos, BalaKrishna Kolluru, Paul Thompson, and Sophia Ananiadou. „Deploying and sharing U-Compare workflows as web services.“ In: *J. Biomedical Semantics* 4 (2013), p. 7 (cit. on pp. 22, 33).

[Kos10] R. N. Kostoff. „Expanded information retrieval using full-text searching“. In: *Journal of Information Science* 36.1 (2010), pp. 104–113 (cit. on p. 26).

[KP13] Jin-Dong Kim and Sampo Pyysalo. „Bionlp shared task“. In: *Encyclopedia of Systems Biology*. Springer, 2013, pp. 138–141 (cit. on p. 4).

[Kra+08] Martin Krallinger, Alexander Morgan, Larry Smith, et al. „Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge“. In: *Genome Biology* 9.Suppl 2 (2008), S1 (cit. on p. 34).

[Kra+11] M. Krallinger, M Vazquez, F. Leitner, et al. „The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full-text“. In: *BMC bioinformatics* 12.Suppl 8 (2011), S3 (cit. on p. 67).

[Kuo+09] C.-J. Kuo et al. „BioAdi: a machine learning approach to identifying abbreviations and definitions in biological literature“. In: *BMC Bioinformatics* 10.Suppl 15 (Dec. 2009), S7 (cit. on pp. 18, 32).

[KW12] Jin-Dong Kim and Yue Wang. „PubAnnotation: a persistent and sharable corpus and annotation repository“. In: *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*. Association for Computational Linguistics. 2012, pp. 202–205 (cit. on p. 16).

[Lar+07] Stephen D Larson, Lisa L Fong, Amarnath Gupta, et al. „A formal ontology of subcellular neuroanatomy“. In: *Frontiers in neuroinformatics* 1 (2007) (cit. on pp. 17, 93).

[Lea+13] Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. „DNorm: disease name normalization with pairwise learning to rank“. In: *Bioinformatics* (2013), btt474 (cit. on p. 20).

[Lei+07] Ed S Lein, Michael J Hawrylycz, Nancy Ao, et al. „Genome-wide atlas of gene expression in the adult mouse brain“. In: *Nature* 445.7124 (2007), pp. 168–176 (cit. on p. 105).

[LG+08] R. Leaman, G. Gonzalez, et al. „BANNER: An executable survey of advances in biomedical named entity recognition“. In: *Pacific Symposium on Biocomputing*. Vol. 13. 2008, 652–663 (cit. on pp. 35, 97).

[Liu+12] H. Liu et al. „BioLemmatizer: a lemmatization tool for morphological processing of biomedical text“. In: *Journal of Biomedical Semantics* 3.1 (Apr. 2012), p. 3 (cit. on pp. 18, 32).

[LM09] Stephen D Larson and Maryann E Martone. „Ontologies for neuroscience: what are they and what are they good for?“ In: *Frontiers in neuroscience* 3.1 (2009), p. 60 (cit. on p. 16).

[LM13] Stephen D Larson and Maryann E Martone. „NeuroLex.org: an online framework for neuroscience knowledge“. In: *Frontiers in neuroinformatics* 7 (2013) (cit. on pp. 2, 17, 92, 93).

[Lu+11] Zhiyong Lu, Hung-Yu Kao, Chih-Hsuan Wei, et al. „The gene normalization task in BioCreative III“. In: *BMC bioinformatics* 12.Suppl 8 (2011), S2 (cit. on pp. 4, 16).

[LW09] D. Lin and X. Wu. „Phrase clustering for discriminative learning“. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. 2009, pp. 1030–1038 (cit. on p. 47).

[MA13] Maryann E Martone and Giorgio A Ascoli. „Connecting connectomes“. In: *Neuroinformatics* 11.4 (2013), pp. 389–392 (cit. on pp. 6, 92, 93).

[MA15] Makoto Miwa and Sophia Ananiadou. „Adaptable, high recall, event extraction system with minimal configuration“. In: *BMC Bioinformatics* 16.Suppl 10 (2015), S7 (cit. on p. 22).

[MAC12] D. Movshovitz-Attias and W. Cohen. „Alignment-HMM-based Extraction of Abbreviations from Biomedical Text“. In: *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, 47–55 (cit. on pp. 29, 68).

[Mar+15] Henry Markram et al. „Reconstruction and Simulation of Neocortical Microcircuitry“. In: *Cell* 163.2 (2015), pp. 456–492 (cit. on p. 108).

[Mar06] Henry Markram. „The blue brain project“. In: *Nature Reviews Neuroscience* 7.2 (2006), pp. 153–160 (cit. on p. 2).

[Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. „Efficient estimation of word representations in vector space“. In: *arXiv preprint arXiv:1301.3781* (2013) (cit. on pp. 3, 20).

[Miy+09] Y. Miyao, K. Sagae, R. S\aetre, T. Matsuzaki, and J. Tsujii. „Evaluating contributions of natural language parsers to protein–protein interaction extraction“. In: *Bioinformatics* 25.3 (2009), 394–400 (cit. on p. 19).

[MS99] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999 (cit. on p. 3).

[MT08] Yusuke Miyao and Jun'ichi Tsujii. „Feature forest models for probabilistic HPSG parsing“. In: *Computational Linguistics* 34.1 (2008), pp. 35–80 (cit. on p. 19).

[Mul+04] Hans-Michael Muller, Eimear E Kenny, and Paul W Sternberg. „Textpresso: an ontology-based information retrieval and extraction system for biological literature“. In: *PLoS Biol* 2.11 (2004), e309 (cit. on p. 3).

[Nat+11] Darren A Natale, Cecilia N Arighi, Winona C Barker, et al. „The Protein Ontology: a structured representation of protein forms and complexes“. In: *Nucleic Acids Res.* 39.Database issue (Jan. 2011), pp. D539–545 (cit. on p. 34).

[New+09] David Newman, Sarvnaz Karimi, and Lawrence Cavedon. „Using Topic Models to Interpret MEDLINE's Medical Subject Headings“. In: *AI 2009: Advances in Artificial Intelligence*. Ed. by Ann Nicholson and Xiaodong Li. Vol. 5866. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 270–279 (cit. on p. 55).

[Néd+13] Claire Nédellec, Robert Bossy, Jin-Dong Kim, et al. „Overview of BioNLP shared task 2013“. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013, pp. 1–7 (cit. on p. 16).

[OB09] P. V. Ogren and S. J. Bethard. „Building test suites for UIMA components“. In: *NAACL HLT 2009* (2009), p. 1 (cit. on p. 33).

[Ogr+08] P. V. Ogren, P. G. Wetzler, and S. J. Bethard. „ClearTK: A UIMA toolkit for statistical natural language processing“. In: *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP* (2008), p. 32 (cit. on pp. 18, 33, 35).

[Oh+14] S. W. Oh, J. A. Harris, L. Ng, et al. „A mesoscale connectome of the mouse brain“. In: *Nature* 508.7495 (2014), 207–214 (cit. on pp. 64, 73, 75).

[Oht+13] T. Ohta, S. Pyysalo, R. Rak, et al. „Overview of the pathway curation (PC) task of bioNLP shared task 2013“. In: *ACL*. 2013, pp. 67–75 (cit. on p. 67).

[Oka+10] Naoaki Okazaki, Sophia Ananiadou, and Jun'ichi Tsujii. „Building a high-quality sense inventory for improved abbreviation disambiguation“. In: *Bioinformatics* 26.9 (2010), pp. 1246–1253 (cit. on p. 19).

[Osb+09] J. Osborne, Jared Flatow, Michelle Holko, et al. „Annotating the human genome with Disease Ontology“. In: *BMC Genomics* 10.Suppl 1 (July 2009), S6 (cit. on p. 34).

[PA13] Ruchi Parekh and Giorgio A Ascoli. „Neuronal morphology goes digital: a research hub for cellular and system neuroscience“. In: *Neuron* 77.6 (2013), pp. 1017–1038 (cit. on p. 95).

[PA14] Sampo Pyysalo and Sophia Ananiadou. „Anatomical entity mention recognition at literature scale“. In: *Bioinformatics* 30.6 (2014), pp. 868–875 (cit. on p. 20).

[PM09] Rob Phillips and Ron Milo. „A feeling for the numbers in biology“. In: *Proceedings of the National Academy of Sciences* 106.51 (2009), pp. 21465–21471 (cit. on p. 38).

[PW06] G. Paxinos and C. Watson. *The rat brain in stereotaxic coordinates: hard cover edition*. Elsevier, 2006 (cit. on pp. 65, 68).

[Pyy+12] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, et al. „Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011“. In: *BMC Bioinformatics* 13.Suppl 11 (June 2012), S2 (cit. on p. 34).

[Raj07] Martin Rajman. *Speech and language engineering*. EPFL Press, 2007 (cit. on p. 3).

[Rak+12] Rafal Rak, Andrew Rowley, William Black, and Sophia Ananiadou. „Argo: an integrative, interactive, text mining-based workbench supporting curation“. In: *Database: the journal of biological databases and curation* 2012 (2012) (cit. on pp. 23, 33).

[Ram+10] Cartic Ramakrishnan, William A. Baumgartner Jr, Judith A. Blake, et al. „Building the Scientific Knowledge Mine (SciKnowMine1): a Community-driven Framework for Text Mining Tools in Direct Service to Biocuration. Malta“. In: *Language Resources and Evaluation* (2010) (cit. on p. 33).

[Ram+12] Cartic Ramakrishnan, Abhishek Patnia, Eduard H Hovy, Gully APC Burns, et al. „Layout-aware text extraction from full-text PDF of scientific articles.“ In: *Source code for biology and medicine* 7.1 (2012), p. 7 (cit. on p. 28).

[Ran+11] Rajnish Ranjan, Georges Khazen, Luca Gambazzi, et al. „Channelpedia: an integrative and interactive database for ion channels“. In: *Frontiers in neuroinformatics* 5 (2011) (cit. on p. 34).

[Ric+13] R. Richardet, J.-C. Chappelier, and M. Telefont. „bluima: a UIMA-based NLP Toolkit for Neuroscience“. In: *Proceedings of the 3rd Workshop on Unstructured Information Management Architecture, Darmstadt, Germany, September 23, 2013*. 2013, pp. 34–41 (cit. on p. 33).

[Ric+15] R. Richardet, J.-C. Chappelier, M. Telefont, and S. Hill. „Large-scale extraction of brain connectivity from the neuroscientific literature“. In: *Bioinformatics* 31.10 (2015), pp. 1640–1647 (cit. on pp. 3, 9, 64).

[Rol13] Orianne Rollier. *Content Extraction from PDF Scientific Articles*. Semester Project Report. Lausanne, Switzerland: Ecole Polytechnique Fédérale de Lausanne, School of Computer and Communication Sciences, 2013 (cit. on p. 29).

[RS+10] Dietrich Rebholz-Schuhmann, Antonio José Jimeno-Yepes, Erik M van Mulligen, et al. „The CALBC Silver Standard Corpus for Biomedical Named Entities-A Study in Harmonizing the Contributions from Four Independent Named Entity Taggers.“ In: *LREC*. 2010 (cit. on p. 16).

[Sas+08] Yutaka Sasaki, Simonetta Montemagni, Piotr Pezik, et al. „Biolexicon: A lexical resource for the biology domain“. In: *Proc. of the third international symposium on semantic mining in biomedicine (SMBM 2008)*. Vol. 3. 2008, pp. 109–116 (cit. on p. 17).

[Sas+09] Yutaka Sasaki, Paul Thompson, John McNaught, and Sophia Ananiadou. „Three BioNLP Tools Powered by a Biological Lexicon“. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*. EACL ’09. Athens, Greece: Association for Computational Linguistics, 2009, pp. 61–64 (cit. on p. 17).

[Sav+10] Guergana K. Savova, James J. Masanz, Philip V. Ogren, et al. „Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications“. In: *Journal of the American Medical Informatics Association* 17.5 (2010), 507–513 (cit. on p. 33).

[Set10] B. Settles. *Active Learning Literature Survey*. Technical Report 1648. University of Wisconsin–Madison, Sept. 2010 (cit. on p. 22).

[Set11] B. Settles. „Closing the Loop: Fast, Interactive Semi-Supervised Annotation With Queries on Features and Instances“. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Edinburgh, Scotland, UK: ACL Press, 2011 (cit. on p. 22).

[Smi+07] B. Smith et al. „The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration“. In: *Nature Biotechnology* 25.11 (2007), pp. 1251–1255 (cit. on p. 16).

[Son+75] U. Sonnhof, P. Grafe, J. Krumnikl, M. Linder, and Lucia Schindler. „Inhibitory postsynaptic actions of taurine, {GABA} and other amino acids on motoneurons of the isolated frog spinal cord“. In: *Brain Research* 100.2 (1975), pp. 327 –341 (cit. on pp. 46–50).

[Spo11] Olaf Sporns. „The human connectome: a complex network“. In: *Annals of the New York Academy of Sciences* 1224.1 (2011), pp. 109–125 (cit. on p. 80).

[Ste+01] Klass E Stephan, Lars Kamper, Ahmet Bozkurt, et al. „Advanced database methodology for the Collation of Connectivity data on the Macaque brain (CoCoMac)“. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 356.1412 (2001), pp. 1159–1186 (cit. on p. 18).

[Swa04] L. W. Swanson. *Brain maps: Structure of the rat brain*. Gulf Professional Publishing, 2004 (cit. on pp. 18, 65, 68).

[Tan+10] Michael A. Tanenblatt, Anni Coden, and Igor L. Sominsky. „The ConceptMapper Approach to Named Entity Recognition.“ In: *LREC*. 2010 (cit. on p. 34).

[Tho+11] P. Thompson et al. „The BioLexicon: a large-scale terminological resource for biomedical text mining“. In: *BMC Bioinformatics* 12.1 (2011), p. 397 (cit. on pp. 17, 34).

[Tom+06] Katrin Tomanek, Joachim Wermter, and Udo Hahn. „A reappraisal of sentence and token splitting for life sciences documents.“ In: *Studies in health technology and informatics* 129.Pt 1 (2006), 524–528 (cit. on p. 30).

[Tri+14] Shreejoy J Tripathy, Judith Savitskaya, Shawn D Burton, Nathaniel N Urban, and Richard C Gerkin. „NeuroElectro: a window to the world's neuron electrophysiology data“. In: *Frontiers in neuroinformatics* 8 (2014) (cit. on pp. 4, 9, 93, 95).

[Vas+15] Xavier Vasques, Renaud Richardet, Sean L Hill, et al. „Automatic target validation based on neuroscientific literature mining for tractography“. In: *Frontiers in Neuroanatomy* 9.66 (2015) (cit. on pp. 80–82).

[Vla+09] Andreas Vlachos, Paula Buttery, Diarmuid O Séaghdha, and Ted Briscoe. „Biomedical event extraction without training data“. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Association for Computational Linguistics. 2009, pp. 37–40 (cit. on p. 21).

[Wan+09a] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y Chang. *Plda: Parallel latent dirichlet allocation for large-scale applications*. 2009 (cit. on pp. 51, 52).

[Wan+09b] Zhiping Wang, Seongho Kim, Sara K Quinney, et al. „Literature mining on pharmacokinetics numerical data: a feasibility study“. In: *Journal of biomedical informatics* 42.4 (2009), pp. 726–735 (cit. on p. 20).

[Wan+10a] Wei Wang, Payam Barnaghi, and Andrzej Bargiela. „Probabilistic topic models for learning terminological ontologies“. In: *Knowledge and Data Engineering, IEEE Transactions on* 22.7 (2010), pp. 1028–1040 (cit. on p. 47).

[Wan+10b] X. Wang, J. Tsujii, and S. Ananiadou. „Disambiguating the species of biomedical named entities using natural language parsers“. en. In: *Bioinformatics* 26.5 (Mar. 2010), pp. 661–667 (cit. on p. 19).

[Wan+10c] Xinglong Wang, Jun'ichi Tsujii, and Sophia Ananiadou. „Disambiguating the species of biomedical named entities using natural language parsers“. In: *Bioinformatics* 26.5 (2010), pp. 661–667 (cit. on p. 20).

[Wer+09] Joachim Wermter, Katrin Tomanek, and Udo Hahn. „High-performance gene name normalization with GeNo“. In: *Bioinformatics* 25.6 (2009), pp. 815–821 (cit. on p. 20).

[Wil+07] J. Wilbur, L. Smith, and L. Tanabe. „Biocreative 2. gene mention task“. In: *Proceedings of the second biocreative challenge evaluation workshop*. Vol. 23. 2007, 7–16 (cit. on pp. 16, 35).

[Wil03] Geoffrey Williams. „From meaning to words and back: Corpus linguistics and specialised lexicography“. In: *ASp. la revue du GERAS* 39-40 (2003), pp. 91–106 (cit. on p. 46).

[Xan14] Aris Xanthos. „Textable: programmation visuelle pour l'analyse de données textuelles“. In: *Actes des 12èmes Journées internationales d'analyse statistique des données textuelles* (2014) (cit. on p. 22).

[Zei+15] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, et al. „Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq". In: *Science* 347.6226 (2015), pp. 1138–1142 (cit. on p. 105).

[Zha+12] Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad L. Alkhouja. „Mr. LDA: a flexible large scale topic modeling package using variational inference in MapReduce". In: *Proceedings of the 21st international conference on World Wide Web*. WWW '12. Lyon, France: ACM, 2012, pp. 879–888 (cit. on p. 52).

[Zim13] Marc Zimmermann. *UIMA Integration of Topic Models*. Semester Project Report. Lausanne, Switzerland: Ecole Polytechnique Fédérale de Lausanne, School of Computer and Communication Sciences, 2013 (cit. on pp. 46, 51).

Online Resources

[@All] *Allen Mouse Brain Atlas, Technical White Paper, Allen Reference Atlas*. 2008. URL: http://help.brain-map.org/download/attachments/2818169/AllenReferenceAtlas_v1_2008_102011.pdf (visited on Aug. 1, 2015) (cit. on p. 18).

[@Had] *Hadoop LDA: A Hadoop MapReduce-based tool for training Latent Dirichlet Allocation models*. 2012. URL: http://code.google.com/p/hadoop-lda (visited on Aug. 10, 2015) (cit. on pp. 52, 53).

[@HK14] R. Hodgson and P. J. Keller. *QUDT-quantities, units, dimensions and data types in OWL and XML*. 2014. URL: http://www.qudt.org (visited on Aug. 10, 2015) (cit. on p. 38).

[@Hof] Matthew D. Hoffman. *Online VB for LDA in VW*. URL: https://github.com/JohnLangford/vowpal_wabbit/wiki/lda.pdf (visited on Aug. 10, 2015) (cit. on pp. 52, 53).

[@Stm] *A statement of commitment by scientific, technical and medical (STM) publishers to a roadmap to enable text and data mining (TDM) for non commercial scientific research in the European Union*. 2013. URL: http://www.stm-assoc.org/2013_11_11_Text_and_Data_Mining_Declaration.pdf (visited on July 21, 2015) (cit. on p. 7).

[@McC02] A. McCallum. *Mallet: A machine learning for language toolkit*. 2002. URL: http://mallet.cs.umass.edu (visited on Aug. 10, 2015) (cit. on pp. 51, 52, 68).

[@Nata] U.S. National Library of Medicine. *Fact Sheet Medical Subject Headings (MeSH)*. URL: https://www.nlm.nih.gov/pubs/factsheets/mesh.html (visited on Aug. 10, 2015) (cit. on p. 109).

[@Natb] U.S. National Library of Medicine. *NLM Medical Text Indexer*. URL: http://ii.nlm.nih.gov/MTI/index.shtml (visited on Aug. 10, 2015) (cit. on p. 111).

[@Natc] U.S. National Library of Medicine. *Searching PubMed with MeSH*. URL: http://ii.nlm.nih.gov/MTI/index.shtml (visited on Aug. 10, 2015) (cit. on p. 111).

Abbreviations

ATMA agile text mining application
BBP Blue Brain Project
CAS common analysis structure
DCA discrete component analysis
BBP Human Brain Project
LDA latent dirichlet allocation
MeSH medical subject headings
NER named entity recognizer
NLP natural language processing
PMID PubMed identifier or PubMed unique identifier
PubMed PubMed provides free access to the MEDLINE database of citations and abstracts in the biomedical domain
TMA text mining application

List of Figures

List of Tables

Colophon

This thesis was typeset with $\LaTeX\,2_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner.

Renaud Richardet

CH-1865 Les Diablerets
+41 78 675 9501
renaud@apache.org

Swiss citizen
Married, two children

EDUCATION

PhD Computational Neuroscience Sept 2011 – Sept 2015
Swiss Federal Institute of Technology, Lausanne, Switzerland
Switzerland's premier university and top European university
- Natural language processing (NLP) for biomedical scientific articles, information retrieval (IR)
- Big data, distributed computing
- Machine learning (e.g. CRF, LDA)

MSc Engineering Oct 1996 - June 2002
Swiss Federal Institute of Technology, Zürich, Switzerland
Bachelors and Masters degrees in Agricultural Economics, major in business economics and marketing
- BA Thesis on algebraic modelization with AMPL (modeling language for mathematical programming)
- Awarded 3rd place prize from Swiss Market Research Association for MS Thesis on product introduction and selection strategies in retail, 2003

PUBLICATIONS

Agile text mining with Sherlok
Richardet R, Chappelier J-C, Tripathy SJ, and Hill S
Special Session on Intelligent Mining, 2015 IEEE International Conference on Big Data

An automated approach for identifying and normalizing mentions of diverse neuron types from the biomedical literature (abstract)
Richardet R*, Tripathy SJ*, Pavlidis P, Hill S
2015 International Conference on Brain Informatics and Health, London.

Automatic target validation based on neuroscientific literature mining for tractography
Vasques X*, Richardet R*, Hill S, Slater D, Chappelier J-C, Pralong E, Bloch J, Draganski B and Cif L
Front. Neuroanat., 2015 (journal impact factor 4.2)

Large-scale extraction of brain connectivity from the neuroscientific literature
Richardet R, Chappelier J-C, Telefont M and Hill S
Oxford Bioinformatics, 2015 (journal impact factor 4.6)

Bluima: a UIMA-based NLP Toolkit for Neuroscience (abstract)
Richardet R, Chappelier JC, Telefont M
UIMA@GSCL 2013, volume 1038 of CEUR Workshop Proceedings

WORK

Research Scientist and Lecturer January 2009 – Sept 2011
University of Applied Sciences Nordwestern Switzerland (FHNW)
- Managed several research projects (technology transfer) and MSc & BSc student projects
- Developed a recommendation system for Swiss startup Amazee.com (project manager)
- Designed and implemented an ontology platform for Swiss startup x28.ch (semantic matching of job vacancies and resumes), using semantic technologies (OWL) and semi-automated pattern matching

Senior R&D Scientist Sept 2010 – Feb 2011
Small Rivers, Lausanne-EPFL, Switzerland
- Developed content and semantic analysis tools for paper.li, using machine learning.
- Large scale distributed data processing (>5M content items per day)

Co-Founder Jan 2010 – Aug 2010
Braindrop, Geneva, Switzerland
- Developed algorithm combining full-text search (content) and social network position (context)
- Awarded funding from Venture Kick's startup promotion program

Software engineer and Teacher April 2008 – Sept 2008
Institute for Indian Mother and Child, NGO, Calcutta, India
- Managed, developed and implemented a microfinance software suite for >20k members bank
- Taught computer skills to students four lessons a week

CTO Dec 2006 – Oct 2007
GalaxyAdvisors LLC, social network analysis, Cambridge MA, USA
- Engineered social network analysis algorithm for large graph computations
- Developed full text search capabilities, content analysis and social network crawlers

COO United States May 2005 – Nov 2006
Wyona, open source software development, Cambridge MA, USA
- Set up and managed new US office of Wyona
- Lead team of developers on various software development projects; clients include Lycos Inc

Project Manager, Market Research March 2003 – Sept 2004
Coop, major retailer and largest Swiss employer, Basel, Switzerland
- Created study design and managed > 30 research studies. Analyzed and presented study results to clients and top management
- Proficient in many research methodologies (SPSS certification), including focus groups, telephone interviews, face-to-face interviews, online questionnaires, eye tracking, and data mining
- Managed 2 MSc Theses on data mining in cooperation with Swiss Federal Institute of Technology

IT SKILLS

Senior J2EE developer with 10 years working experience, Sun Certified Java Programmer
Open source developer (Apache committer in the UIMA project)

Expert Knowledge:
- Java J2EE, Spring, Scala, Hibernate, JUnit, Travis CI
- Information retrieval (Lucene, Nutch, Solr, LingPipe)
- MySQL, MongoDB, RabbitMQ
- Hadoop, Pig
- EC2, S3, Elastic Map Reduce
- Eclipse IDE, Maven, Ant, Git, Linux

Good Working Knowledge:
- Web development (HTML5, AngularJS, Django, Ruby on Rails)
- Scripting languages (Python, Javascript, Ruby, Bash)

LANGUAGES
- *French*: native
- *English*: fluent (3 years in USA)
- *German*: fluent (9 years in German speaking Switzerland)
- *Italian, Spanish*: intermediate

EXTRACURRICULAR
- President of ASSIIME, Swiss support branch of the IIMC, a NGO in Kolkata, India
- Co-founder and treasurer of Geneva Net Dialogue, a NPO focusing on international Internet governance and human rights
- Avid snowboarder and back-country hiker
- Frequent traveler, including extended trips to India, South America and throughout Europe