A Multi API approach for Natural Language Processing in Unstructured Clinical Documents

Author:
Hugo van Krimpen

Supervisors & Co-authors:
Dr. Marco Spruit
MSc. Shengru (Ian) Shen

A thesis presented for the degree of
Master of Science

Universiteit Utrecht

Faculty of Information & Computing Science
Utrecht University
The Netherlands
August 28, 2017

Contents

1 Introduction **5**
- 1.1 Background . . . 5
- 1.2 Relevance . . . 8
- 1.3 Problem Statement . . . 9
- 1.4 Research Questions . . . 9

2 Research Method **11**
- 2.1 Design Science Research . . . 11
  - 2.1.1 Design Science Research Framework . . . 12
  - 2.1.2 Design Science Research Guidelines . . . 13
- 2.2 Literature Research . . . 13
  - 2.2.1 Approach . . . 13
  - 2.2.2 Scope . . . 15
- 2.3 Data Ethics . . . 15
- 2.4 Evaluation . . . 17
  - 2.4.1 Framework Evaluation . . . 18

3 Theoretical Background **20**
- 3.1 Existing Frameworks and Architectures . . . 20
  - 3.1.1 GATE . . . 20
  - 3.1.2 UIMA . . . 22
  - 3.1.3 NERD Framework . . . 23
  - 3.1.4 General clinical NLP system . . . 25
  - 3.1.5 HITEx . . . 26
  - 3.1.6 cTakes . . . 26
  - 3.1.7 ClearTK . . . 28
  - 3.1.8 Health-CPS . . . 28
- 3.2 NLP in Information Extraction . . . 29
  - 3.2.1 NLP Tasks . . . 30
  - 3.2.2 Rule based and Statistical based NLP . . . 31
  - 3.2.3 Natural Language Processing Output . . . 32
  - 3.2.4 Negation . . . 33
- 3.3 Interoperability & Standardizations . . . 34
  - 3.3.1 Syntactic Interoperability . . . 35
  - 3.3.2 Semantic Interoperability . . . 35
- 3.4 Information Fusion . . . 38
- 3.5 Multi-API based NLP . . . 39

4 Multi-API based NLP Framework 42
4.1 Framework . . . 42
4.2 Implementation . . . 44
4.2.1 Remote NLP Tasks . . . 44
4.2.2 Local NLP Tasks . . . 48
4.2.3 Knowledge Database . . . 49
4.2.4 Information Fusion . . . 49
4.3 Prototype . . . 50
4.3.1 Requirements . . . 50
4.3.2 Early Versions . . . 51
4.3.3 Final Version . . . 51

5 Results 58
5.1 Results - Overall . . . 59
5.2 Results - Obesity Challenge . . . 60
5.3 Results - Medication Challenge Data Set . . . 61
5.4 Results - STRIPA . . . 64

6 Conclusion 67
6.1 Research Questions . . . 67
6.2 Discussion and limitations . . . 68
6.3 Future Work . . . 69

References 71

Preface

In this master thesis of Hugo van Krimpen for the master Business informatics at Utrecht University, we aim to provide insight on the current state of natural language processing. We will propose a framework for improvement of natural language processing and evaluate it by implementing a prototype.

The document will first focus on the background of the topic at hand, followed by the problem statement. The next chapters will describe the methodology, relevance and the approach for the literature review of the research. Next, the document will focus on the proposed framework and go into more depth on all the elements defined in the framework. A prototype was developed to test the framework which will be presented and evaluated in this thesis. This prototype will allow evaluation of the framework.

This research is a sub research of the STRIPA project (Meulendijk, Spruit, Jansen, Numans, & Brinkkemper, 2015). A small data set from the actual STRIPA will be used as evaluation for the proposed framework. More about the STRIPA application will be described in the chapter "Background".

Acknowledgements

Special thanks go to Marco Spruit en Shengru (Ian) Shen for allowing me to do this research and assisting me with whatever was necessary. From the beginning to end, from research to practical context, they were there to assist me and trying to improve the current status of the project. Whenever it was needed, I could just walk by and they would find time to assist me.

I would also like to thanks my parents, Dirk en Marijke van Krimpen, for always supporting me no matter the situation, no matter the setback. They also assisted by thinking along with the project and coming up with new ideas, additional information or interesting directions I could take into making this thesis better and more complete.

Last but not least I would like to mention the group of friends who were willing to proofread the thesis. Even though it was a last minute request, they freed time to help me out perfecting the document. Without them the syntactical quality would not have been as high as it is now.

Glossary

Term	Definition
Natural Language Processing	Natural Language Processing (NLP) is a field of computer science and linguistics concerned with the interactions between computers and human natural languages. (Kumar, 2011) We use the term natural language processing to refer to the set of text manipulation tasks which aim to transform unstructured text to structured data. NLP is the means for achieving the task / goal of information extraction.
Information Extraction	The task of extracting structured information out of unstructured or semi-structured data. We define information extraction as the task or goal, whereas natural language processing is the means of achieving the task.
Information Retrieval	Information retrieval is a subfield of computer science that deals with the automated storage and retrieval of data or information, specifically documents. (Croft & Lafferty, 2013; Frakes & Baeza-Yates, 1992)
API	Application Programming Interface is a clear predefined set of methods for communication between two systems.
Extractor	An extractor is a reference to one of the NLP services we used.
Extraction	An extraction is the tangible result of performing information extraction with an extractor. An extraction can come in various forms, however our prototype only extracts "Entities", a data object containing a term and a category.
Accuracy	An evaluation metric on which we base the utility of our framework. Accuracy is measured by calculating the number of true positives, false positives, true negatives and false negatives of an NLP task and is always a value between 0 and 100.
F-Measure / F-Score	An evaluation metric on which we base the utility of our framework. F-Measure is a value between 0 and 1. When talking about Accuracy of a NLP process in general, this term may be interchanged with Accuracy.
Biomedical informatics	Biomedical informatics is the application of the science of information as data plus meaning to problems of biomedical interest.(Bernstam, Smith, & Johnson, 2010)
Concept	We refer to a concept as a notion within a meta-terminology database.

1. Introduction

1.1 Background

In a world with an ever increasing amount of data, the need to organize this data and extract relevant information or knowledge from it becomes an increasing necessity for every discipline. Big data intensifies the need for sophisticated statistics and analytic skills (Wixom et al., 2014), mainly because of the fact that data are stored in various different structures or formats. (Zhang, Qiu, Tsai, Hassan, & Alamri, 2015) This data can be either structured (database records), semi-structured (HTML text) or unstructured (text, speech, video, audio). (Das & Kumar, 2013; Kambatla, Kollias, Kumar, & Grama, 2014; Zhang et al., 2015) While every type of data brings challenges when it comes to being handled by computers (Chen, Mao, & Liu, 2014), unstructured data is still the hardest to work with. Figures show that a staggering 80% of all data is unstructured. (Grimes, 2008; Tan et al., 1999) As the amount of data is increasing exponentially fast, the amount of people that understands this data decreases. Potential useful information lies hidden within this data. (M. Hall et al., 2009) Therefore getting and understanding of this data will unlock a large amount of information. (Das & Kumar, 2013; Marr, 2016)

Over the past years, the amount and use of data has seen a large increase in various fields of application. (Chen et al., 2014) A major application of Big Data and Data Mining is gathering medical and biomedical information from large amounts of clinical data. (Koh & Tan, 2011) Many organizations and governments have recognized the importance of handling and using this data to gain benefit within their organization. (Chen et al., 2014; Demirkan & Delen, 2013; Koh & Tan, 2011; Labrinidis & Jagadish, 2012)

Data mining is the process of extracting implicit, previously unknown, and potentially useful patterns and knowledge from large databases. (M. Hall et al., 2009; Han, Haihong, Le, & Du, 2011) Text mining is a process that can be compared to data mining, but focuses on the handling of unstructured or semi-structured data, like text or HTML files. (Tan et al., 1999; Gupta, Lehal, et al., 2009) Aggarwal and Zhai (2012) leave the definition of text mining vague, arguing it envelops a large set of various topics like Information Extraction and Natural Language Processing (NLP).

Text mining uses NLP tasks to perform IE. (Feldman & Sanger, 2007; Gupta et al., 2009; Weiss, Indurkhya, Zhang, & Damerau, 2010) Not everyone sees NLP as a technique of IE but another technique of Text Mining. (Abbott, 2013; Feldman & Sanger, 2007) We define IE as the goal where NLP is the method to achieve the goal: transforming unstructured or semi-structured data to a structured form. (Weiss et al., 2010; Abbott, 2013) Figure 1.1 shows the taxonomy of data mining terms.

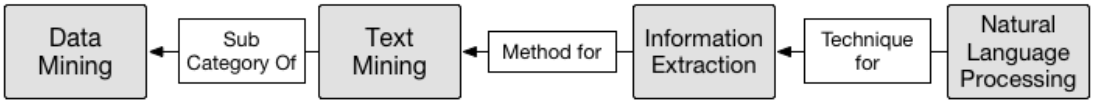

Figure 1.1: The taxonomy of Data Mining terms.

Natural Language Processing Natural Language Processing (NLP) is the process for understanding natural language. NLP began in the 1950's as an intersection between artificial intelligence and linguistics. In the following years, NLP went from a process nearly identical to Information Retrieval towards its own definition of analyzing semantics of natural language. (Nadkarni, Ohno-Machado, & Chapman, 2011)

Nowadays, NLP envelops a large amount of tasks to understand the meaning of a natural language. (Collobert et al., 2011) The ultimate goal of NLP is to build computer systems that perform as well at using natural languages as humans do, but an immediate goal of NLP is to build a system that can process text and speech more intelligently. (Ng, 2008) The NLP system must at least be able to answer the question "Who or what did what to whom or what?". (Manning, Schütze, et al., 1999) The problem of natural language understanding, and thus the problem NLP tries to solve, can be split into two (Kumar, 2011):

- The problem of understanding natural language text using syntactical, semantical and lexical information.
- The problem of understanding natural language speech, using all information needed for the first problem plus additional information on how to tackle verbal challenges like phonetics.

Both problems have their challenges and both know real world applications like the Google search engine and Apple's Siri.

But what is it that makes natural language processing of a text so difficult? A NLP system needs to figure out the structure of a text before it can do anything with it. Multiple challenges like ambiguity in syntax, semantics, discourse and morphology all have to be overcome before accurate NLP can be applied. (Ng, 2008; Kumar, 2011) As an example, the sentence *"Our company is training workers"* may already be interpreted in multiple different ways by conventional NLP systems: either "our company is training workers so they can function correctly" or "our company is 'training workers' ", in which case training workers is the name or status of the company. This simple example of ambiguity presents one of the major difficulties in NLP systems. A good NLP system must be able to make disambiguation decisions of word sense, word category, syntactic structure and semantic scope. (Manning et al., 1999)

Currently there are dozens of businesses that offer Text Mining software either as on premise software or as a service in the cloud. (Harris, 2017) These pieces of software offer a bundle of functions, often identical to Natural Language Processing techniques like entity extraction, sentiment analysis and classification. All these businesses claim to be able to perform correct text analysis, including texts containing clinical data like the texts which are found in Electronic Health Records.

Electronic Health Records Electronic Health Records (EHR) describes the concept of a comprehensive and cross institutional collection of a patient's health and health care data. (Hoerbst & Ammenwerth, 2010) It contains clinical and administrative data about the complete treatment of a patient by a medical expert. (Ambinder, 2005) This includes data that is not only particularly relevant to a subject's medical treatment but also to a subject's health in general. (Hoerbst & Ammenwerth, 2010) EHR's contain both structured (name, age, etc) and unstructured data.

As with other types of data, a staggering 80% of medical and clinical information is stored as unstructured data such as written physician notes, consultant notes, radiology notes, pathology results, discharge notes from a hospital, etc. (Marr, 2016; Das & Kumar, 2013) This data may lead to discoveries that improve the understanding of patient conditions, diagnosis and treatment of diseases. (Hripcsak & Albers, 2013)

As Jensen, Jensen, and Brunak (2012) depicted, applying text mining to EHR's will yield immense amounts of value as this will decrease time, money and energy needed for the diagnosis and treatment of patients. This data can then be imported in Clinical Decision Support Software (Stewart, Shah, Selna, Paulus, & Walker, 2007) to gain a clinical advantage.

Clinical Decision Support Software Clinical Decision Support Software (CDSS) is a sub type of Decision Support Software(DSS), providing computerized support in the making of decisions in the clinical setting by using information and communication technologies. (Greenes, 2011; Haynes, Hayward, & Lomas, 1995; Saverno et al., 2010) CDSS may assist medical experts in their day to day practices or it can automate the process of decision making, assisting or even leaving out the professional. (Greenes, 2011)

The Systematic Tool to Reduce Inappropriate Prescribing Assistant (STRIPA) (Meulendijk et al., 2015) is a CDSS attempting to improve the medicine prescription and medicine usage of elderly people. Stichting Farmaceutische Kengetallen (2005) shows that 17% of the chronically ill patients use over 5 types of medicine on a regular basis. Moreover, half of these patients are over 70 years old. This problem is known as Polypharmacy. (Hajjar, Cafiero, & Hanlon, 2007; Leendertse, Egberts, Stoker, & van den Bemt, 2008) The chronic use of multiple medications by a patient often leads to severe clinical problems, including adverse drug effects, under prescribing, overtreatment, low patient compliance and decreased drug adherence. (Björkman et al., 2002; Claxton, Cramer, & Pierce, 2001; Munger, 2010; Steinman et al., 2006; Wright et al., 2009)

As the usage of the CDSS tools like STRIPA is increasing, the need for an easy way to import the data into the system increases with it. Like other CDSS, STRIPA works with structured and standardized data. Therefore, all unstructured data has to be transformed into structured information and standardized, before it can be used by the CDSS.

MAPI-NLP In this work, we will introduce a new framework, utilizing a Multi Application Programming Interface approach to process natural language. This Multi Application Programming Interface Natural Language Processing (MAPI-NLP) Framework will provide a better solution in the context of clinical NLP, aiming

to improve the use of EHR's and CDSS. More about the MAPI-NLP will be discussed in chapter 4.

1.2 Relevance

Scientific Relevance

This project attempts to fill in the research gap on how to adequately and efficiently extract clinical information from unstructured clinical data and evolve it into structured and standardized data for secondary use. A lot of research has been performed in the area of natural language processing (Chowdhury, 2003; Liddy, 2001; Collobert et al., 2011) and a multitude of applications performing NLP tasks like the Stanford CoreNLP Toolkit.(Manning et al., 2014)

However, the existing research and applications all have a similar approach and/or focus on the individual NLP tasks, based on e.g. the GATE (Cunningham, Wilks, & Gaizauskas, 1996) or UIMA framework. (Ferrucci & Lally, 2004) We propose a new approach to the NLP process, utilizing elements from older architectures and frameworks to create a better solution for NLP.

In fact, this project is the first to attempt NLP through this new approach. We identify this new approach, explore the positives and the negatives and finally put this approach to the test. From this we will gain new knowledge which can help improve either this NLP solution in future work, or other NLP solutions that are looking for new insights.

Business Relevance

There are a lot of organizations that hit a wall when it comes to working with large amounts of (unstructured) data. Projects may stand or fall with the availability of accurate information extraction and natural language processing. The incentive for this thesis, the need of data input into the STRIPA system, is a good example for this. However, there are many more projects which might be able to use this technology.

This project is a step towards a better view on the current approaches and techniques on natural language processing, the necessities for these techniques and the overall best approach on the extraction of entities from large amounts of unstructured clinical data. If it succeeds, clinical data can be shared and used in a structured format which will benefit the complete clinical work flow.

Another struggle current NLP approaches have is that it can take a long time for a NLP system to analyze the unstructured data. NLP is a relatively heavy process that can take a lot of power to run. Businesses want to reduce the money they have to spend while increasing the output of their business. A major consideration in this is to either make (insource) or buy (outsource) a specific product or service. (M. Lacity, Yan, & Khan, 2017) In the world of NLP, many businesses are working on making a better NLP algorithm. We feel, however, that much knowledge, time and other forms of profit are to be gained when instead of developing an even more complex NLP algorithm, an outsource approach is used.

Time is not the only factor that is important to improve upon. Scalability of domain (Bates & Weischedel, 2006) and language (like arabic (Habash, 2010)) offer

a wide variety of challenges and opportunities for improvement.

1.3 Problem Statement

Figures show that 80% of all clinical data is unstructured. (Das & Kumar, 2013; Gharehchopogh & Khalifelu, 2011; Grimes, 2008; Tan et al., 1999) This unstructured data contains information that can be used to gain advantages in the clinical domain, like better medical support and automation. However, before we can use the data is has to be transformed to a structured format using Text Mining techniques. The current state of text mining still has room for improvement, as both the F-score and accuracy is only around 80%-90% when using optimal data sets.

Raghupathi and Raghupathi (2014) state that health care data is rarely standardized, often fragmented or in an incompatible format due to legacy systems. Due to incompatibility, many information systems can not be combined and thus do not function properly when information is gained from an outside source. This may limit improving treatments significantly.

It has long been recognized that Clinical Decision Support Systems are of significant importance when it comes to assisting healthcare workers and medical experts browse through the vast amount of medical conditions and medications. (Greenes, 2011; Musen, Middleton, & Greenes, 2014) The incompatibility between the unstructured data found in EHR's and the structured and standardized information CDSS requires poses a problem that has not yet been adequately solved.

We must conclude that previously proposed solutions are not adequate enough for solving the problem of extracting information from unstructured clinical data. The problem is twofold:

- Existing Frameworks and techniques for Information Extraction are considered inadequate: accuracy improvement is needed.
- Health care data is fragmented, in an incompatible format and rarely standardized: extracted clinical information is hard to fuse and standardize for secondary use.

1.4 Research Questions

Based on the problem statement, a main research question and multiple sub research questions are developed:

MRQ: How can unstructured clinical data be evolved to structured and standardized information utilizing existing NLP API's?

SRQ 1: How can we reuse existing API-based NLP solutions?
SRQ 2: What are the characteristics of the multi-API based NLP framework?
SRQ 3: How to implement the framework?
SRQ 4: Does the concept of information fusion work on entity extraction?
SRQ 5: How does the implementation perform with clinical test data?

This thesis will focus on developing a framework that presents a way to extract information from clinical data, to fuses and standardizes this information and to disseminate it for secondary use. Along with the framework, an implementation artifact will be developed. This implementation artifact, in the form of an online application, will function as an evaluation for the proposed framework.

Creating such a framework in combination with a tool has multiple objectives:

- Providing a better scientific understanding and thus accelerating innovations in the area of natural language processing and information extraction.
- Creating a stable framework on which companies can base further research or application development.
- Increase the stability of a product and the speed at which a product can be developed.

2. Research Method

The entire thesis project exists out of two main phases: A research phase and an execution phase. The research phase is planned to take up to 3 months. The execution phase may take up to another 6 months. These phases will be executed using the Design Science Research Methodology.

During the first phase the focus will be on the preliminary research steps:

- Becoming aware of the problem statement and incentive.
- Researching the current knowledge on the topic at hand.
- Designing the proposed framework which will be one of the deliverables of the project.

These steps include exploring all the relevant sub processes, related research on these sub processes, and the first steps towards the implementation of these processes. The final deliverable will be an artifact presenting a framework for multi-API based NLP.

During the second phase, a prototype to evaluate the proposed framework will be developed. This implementation will be able to test whether the presented framework is able to adequately perform NLP.

2.1 Design Science Research

This project uses the design science research methodology. Therefore it has a lot in common with the design science research cycle (DSR). (Vaishnavi & Kuechler, 2015) This cycle, presented in figure 2.1, states that there are five steps in the design science research cycle, ultimately leading to a certain knowledge contribution and creating opportunities for theory development and refinement. This project will follow the steps of the DSR cycle to create a solution which adds new knowledge to the knowledge base.

The first step is becoming aware of the actual problem the research is trying to solve. This does not only help to define a scope for the project, but also provides a good foundation of existing work and their potential improvements. The problem we try to solve in this project can be found in section 1.3 and the theoretical background can be found in 3.1

The second step is creating a 'suggestion' to solve to problem. This artifact is a model, method, framework or architecture which might solve the defined problem. This step is similar to phase one of this thesis, as we will research previous work to propose a framework as solution to the problem. The final 'suggestion', the MABNLP framework, will be discussed in chapter 4.

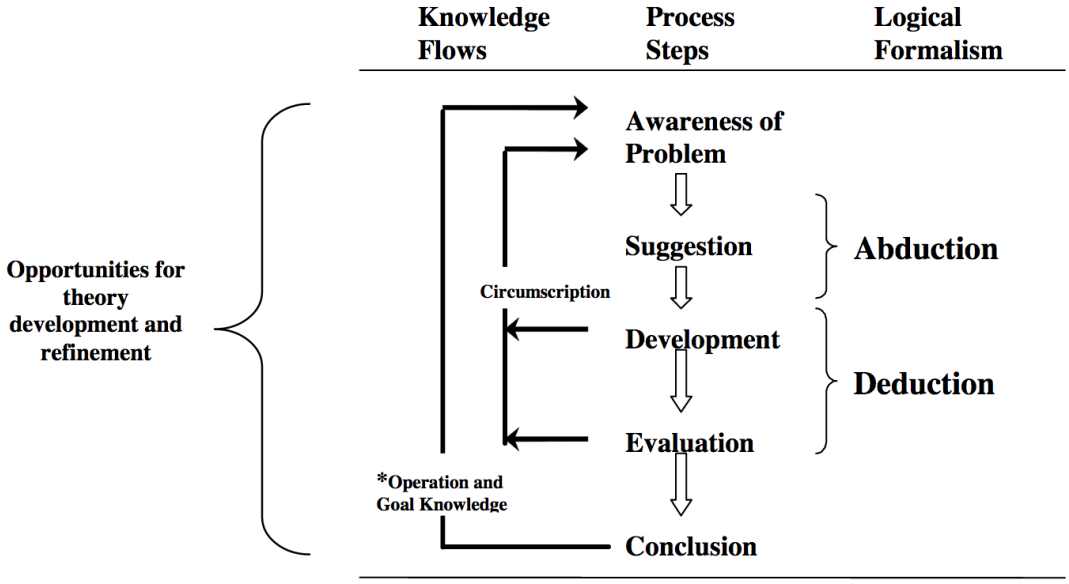

Figure 2.1: The Design Science Research Cycle (Vaishnavi & Kuechler, 2015).

Once this suggestion is completed, an instantiation artifact will be developed to evaluate the framework. This is step three of the DSR and the start of phase two of the thesis. The instantiation artifact will be a prototype that will function as an evaluation tool for the proposed design artifact. We will use the prototype to process several clinical documents, extract entities out of them and perform binary classification to measure the accuracy and f-score. The implementation will be discussed in section 4.2 and the evaluation of the prototype in 2.4.

In the end, conclusions will be drawn on whether the proposed design artifact will be any success, based on the results gathered by the instantiation artifact. We will discuss the results, look back on what could have gone better and suggest ideas and improvements for future work. The conclusion can be found in chapter 6.

We must mention that there are slight differences between the DSR cycle and the approach of this research. The DSR cycle has a large focus on the iterative improvement of the solution while at the same time contributing to the knowledge database. This thesis project will also include an iterative approach. However, the approach will be to acquire the most knowledge during the first phase and to put this into practice in the second. However, as getting this right in one attempt is most likely unrealistic, iterations will still occur. Also, the most knowledge contribution will be provided at the end, when the evaluation and conclusions are inferred.

2.1.1 Design Science Research Framework

When placing the DSR cycle in a more abstract view of the design science paradigm, we get the DSR framework. (Von Alan, March, Park, & Ram, 2004; Hevner, 2007) This framework is also applicable to this project. It can be seen as three closely related cycles of activities. The relevance cycle connects the environmental context with the DSR activities. The rigor cycle connects the previous found work with the DSR activities. The final cycle is the design cycle itself, which includes the development of the artifacts followed by the evaluation of the artifact.(Hevner, 2007)

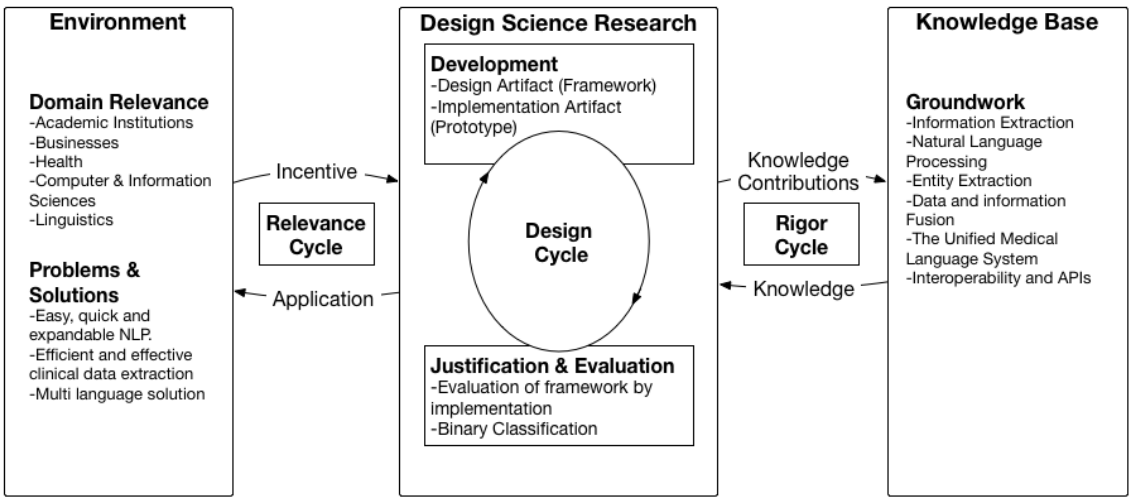

Figure 2.2: The adapted Design Science Research Framework

Figure 2.2 shows what we get once we adapt the framework to this project.

2.1.2 Design Science Research Guidelines

The DSR framework states that for a design science research to be implemented correctly, it has to satisfy seven guidelines. The guidelines are presented and explained in table 2.1.

2.2 Literature Research

This section will describe the approach and scope of the literature research. As there is a near infinite set of research done in this and similar domains, it is necessary that this is scoped down to a workable amount.

2.2.1 Approach

The research started out by finding existing key terms in the area of natural language processing, analytics and information extraction. By applying the snowball method to the found literature, we got more in depth knowledge of the previously mentioned topics. We did not only snowball through the references, but also through relevant terms used in the literature. We saved these terms together with one or multiple definitions. These terms form the very base of the research and are relatively abstract terms. The list can be found in appendix A.

This initial step was executed for multiple reasons:

- To have a concrete starting point of the literature research
- To avoid using specific terms interchangeably, while they are actually slightly different (or not being aware of the differences between two terms)
- To get an overview of all the areas within the research scope, specifically the for us relatively unknown clinical area.

From this base, the research continues using the snowball method. Also, when a new term is found in one of the papers, this term is used to search relevant papers.

Design Science Research - Seven guidelines to create satisfactory artifacts	
Guideline	Justification
Design as an Artifact	The thesis project will develop a design artifact and an implementation artifact.
Problem Relevance	The challenge of evolving unstructured data to structured information is a major one, spanning multiple disciplines. Currently there is a major focus on this process in the clinical field of work.
Design Evaluation	The implementation artifact will evaluate the framework artifact.
Research Contribution	The framework presents a new approach in the context of NLP. It contributes both the knowledge gained while developing the framework and the results of the framework.
Research Rigor	The artifacts will be produced with scientific rigor in design, development and evaluation.
Design as a Search Process	The research will be done in iterations of gathering knowledge, adding knowledge and evaluating this knowledge.
Communication of Research	The research will be communicated through the thesis document and presentations during the MBI Colloquium.

Table 2.1: The seven guidelines to satisfy for design science research.

This approach is found appropriate because the goal of this thesis is not to provide an in depth review of what has been done, but instead aims to build upon ideas that have been research in the past.

2.2.2 Scope

The work done in the area of clinical natural language processing is rapidly increasing, and new ideas and innovations are released on a daily basis. While there might be older fundamental techniques or knowledge on which the research might be based, it will be unnecessary to put extra focus on older work as this work will mostly be outdated already. Therefore the earliest date we search literature on is 2010 (7 years at the time of writing). However, some work is still relevant even though it was published before the year 2010. Those papers are mostly found through the use of the snowball method and are included in the work. When it is necessary to explain more about a specific topic, references to older literature are made.

Due to the limitations of language, we are only able to include English or Dutch literature. As it turned out, all literature we found is English, even papers by Dutch authors. We feel, however, that this language limitation still gives a good view of the current literature.

The main focus is on finding literature that combines the area of natural language processing with the clinical domain. Many documents we use combine these topics. However, we found some of the terms or methods had their roots in other domains like law, risk management and military. We did not exclude this literature as we found the knowledge they provide on the history of some topics relevant. An example of this is the process of Information Fusion. (Section 3.4)

We use two sources to find scientific literature: PubMed and Google Scholar.

- PubMed: to find clinical related scientific literature
- Google Scholar: to find other related scientific literature

Other, non scientific literature was also found through the use of normal search engines like Google, web pages, blogs, and articles. They mostly provided particular good information on some less scientific or complicated topics like Binary Classification and Unstructured Data.

This also implies the quality of literature we accepted. For a definition of terms or important statements we always use literature that is well accepted by the academic community, was cited by many other authors and/or was published in a well respected journal. When using non scientific literature, we make sure the source is credible, like an article or blog from a large (academic) institution or well known company.

2.3 Data Ethics

Data Ethics is of importance for every framework, architecture or system that handles personal or confidential data. As this thesis proposes a framework that does exactly that, attention should be paid to the area of data ethics. However, as the safe handling of data is not a major focus area of this thesis, we will not propose any

additions to our framework specifically for the safe handling of data. It will however be vital to include during future research or projects based upon this framework.

Because the data is frequently data about people and their characteristics and behavior, the potential use and abuse of this acquired data extends in a great many directions.

- Zwitter, (2014)

The area of (big) data ethics is a very difficult one. The ethical discussion is about what is right and what is wrong, what is good and what is bad. The lack of a common vision on what is right and what is wrong can create obstacles finding a consensus on how to handle big data. (Zwitter, 2014).

On the one hand, the use of personal data can lead to new discoveries and solutions to problems we are currently unable to tackle. On the other hand, concerns people have about the consequences of having personal data captured, aggregated, sold, mined, re-sold, and linked to other data are starting to become really large. People worry that large companies might negatively use personal data to gain value or that data might fall into the wrong hands.

We are at a critical point in time, where we must balance the use of big data with the protecting of human values like privacy and confidentiality. These values are what we as a society, in different degrees, believe in. If society fails to respect these values, organizations that use big data might trade these values for new innovations, putting our human values even more at risk. (Richards & King, 2014; Zwitter, 2014)

To respect society's values, Richards and King (2014) define four high-level principles that we should recognize while handling data within an information society. These principles are:

- **We must recognize "privacy" as information rules.** While the amount of personal information that is being recorded is certainly increasing, so too is the need for rules to govern this social transformation. Richards and King argue that "privacy" in today's information economy should be understood as a set of rules that handle the data within organizations in an ethical way.

- **We must recognize that shared private information can remain "confidential."** Much of the tension in privacy law over the past few decades has come from the simplistic idea that privacy is a binary. It is either private or public. The thought is that once information is shared and consent given, it can no longer be private. Understanding that shared private information can remain confidential better helps us to see how to align our expectations of privacy with the rapidly growing secondary uses of big data analytics.

- **We must recognize that big data requires transparency.** Transparency can help prevent abuses of institutional power while also encouraging individuals to feel safe in sharing more relevant data to improve big data predictions for our society.

- **We must recognize that big data can compromise identity.** Identity is referred to as the ability of individuals to define who they are. Big data predictions risk compromising our identity since organizations can use these

predictions to identify, categorize and even determine who we are before we have made up our own minds. Richards and King therefore believe that we should regulate or even prohibit certain predictions to protect the people.

Personal data should be handled with care. Several governments even created laws to enforce this. In the Netherlands, the "Law for the protection of personal information" was created as an instance of the European guidelines. The law cites that:

- Personal data should be handled "adequately".
- Personal data should be stored in a safe manner.
- The person who the data belongs to should be informed about:
  - The goal of using the data
  - The identity of the organization that handles the data

This project makes use of data that belongs to individuals. By using this data, we have to respect the above described human values and laws.

This project will make use of only anonymized and de-identified data. This will ensure that no data can be linked to a single human being. This project does not aim to tackle the problem of data ethics nor does it aim to build an application or framework which does.

However, it is worth noting that in the near future, rules concerning data protection might change with the introduction of the General Data Protection Regulation (GDPR). (Flint, 2017) The changes include a shift from the data controller having the sole responsibility to a combined responsibility for both the data controller and the data processors. This results not only in our framework (the data controller) needing to handle the data securely, but also any remote NLP services (the data processors) to obey the data protecting rules, possibly forwarding these rules to the data controllers. These rules will only apply from May 2018 onwards, and thus will not be considered during this project.

2.4 Evaluation

One of the goals of the implementation of the framework is the capability to evaluate the framework itself. Evaluation is what puts the "Science" in "Design Science". Without evaluation, we only have an unsubstantiated design theory or hypothesis that some developed artifact will be useful for solving some problem or making some improvement .(Venable, Pries-Heje, & Baskerville, 2012) In their third guideline for Design Science in Information Science Research, Hevner, March, Park, and Ram (2004) state that "The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods".

Venable et al. worked on this by proposing a four step method to find the correct evaluation method for one's design science research. The four steps function around the DSR Evaluation Strategy Selection Framework: a table containing four possible evaluation strategies. The strategy set selected is based upon criteria you set for the evaluation of your framework or architecture.

Based on the DSR Evaluation Strategy Selection Framework by Venable et al., we can conclude that the evaluation will be Naturalistic (in the real environment) -

Ex Post (evaluation using an actual implementation). From this we get the following possible techniques of evaluation:

- Action Research
- Case Study
- Focus Group
- Participant Observation
- Ethnography
- Phenomenology
- Survey

Not all of these techniques are useful for this project. However, we find that by using a case study we will be able to test our implementation and thus the framework. We are currently in possession of anonymous clinical data which can be used for a case study on the implementation.

The framework will be tested for utility. While utility can not be measured in exact numbers or by calculations, it can be measured in relative terms. Venable et al. states there are five purposes to evaluate in DSR. The one applicable to this project:

"Evaluate an instantiation of a designed artifact to establish its utility and efficacy (or lack thereof) for achieving its stated purpose"

It states that its central purpose is to demonstrate the utility of the artifact being evaluated. Therefore we have to demonstrate that the artifact adds value compared to not having the artifact and to what degree the artifact achieves its purpose. We will achieve this by passing the clinical data we have through the prototype to see if it fulfills the purpose of the design artifact.

2.4.1 Framework Evaluation

For the evaluation of the implementation we will be performing binary classification. Binary classification divides a data set in two, labeling all entities in the dataset with either a positive or a negative classification. Comparing this with the ground truth results in four possible outcomes:

- True Positive (TP): The entity was labeled correctly positive.
- False Positive (FP): The entity was labeled incorrectly positive.
- True Negative (TN): The entity was labeled correctly negative.
- False Negative (FN): The entity was labeled incorrectly negative.

The most commonly used measurements in binary classification, are the Recall, Precision, F-Score and the Accuracy. (Sokolova, Japkowicz, & Szpakowicz, 2006) F-score is calculated by measuring the Recall and Precision. Accuracy is calculated by using the amount of TP, FP, TN and FN directly. F-Score is a better measurement than Accuracy (Cisneros, 2013), however Accuracy is far more common and the normal term to use when talking about Binary Classification. Therefore we will calculate and present both.

- Recall is the percentage of true positives in the set of true positives and false negatives. In other words, it is the percentage of correctly extracted true data. It is calculated as follows
  R = TP / TP + FN

- Precision is the percentage of true positives in the complete set of positives. In other words, it is the percentage of correctly extracted data in the collection of data we know is correct plus the data that was flagged to be correct, but was actually incorrect. It is calculated as follows
  P = TP / TP + FP

- F-measure (or F-Score) describes the balance, or harmonic mean, between the Precision and Recall. It is a value between 0 (bad) and 1 (good). It is calculated as follows:
  F = 2 * ((P * R) / (P + R))

- Accuracy is the ratio between the correctly classified entities compared to the complete set of entities. It is calculated as follows:
  Acc = (TP + TN) / (TP + FP + TN + FN)

The TP, FP, TN and FP values will be the output of the prototype. With that we calculate the Recall, Precision, F-Score and Accuracy, which will provide a good view of how the framework performs.

3. Theoretical Background

In this chapter we will go into more detail about the work that has already been done in this field of work. It is important to build a good foundation of knowledge to be able to correctly develop a complete framework.

3.1 Existing Frameworks and Architectures

This section will discuss some of the currently existing frameworks and architectures used for natural language processing, specifically those who focus on extracting entities. It is an important step to first identify the current work done in the area of NLP frameworks, architectures and tools. Not only is it necessary to identify what has already been done so we do not propose something identical, it also makes us aware of the current NLP solutions including advantages and disadvantages they bring.

Table 3.1 presents a small summary of all the frameworks, architectures and systems. We can divide the artifacts in two different categories: General and based upon. The General artifacts are mostly frameworks which depict an approach for NLP. The based upon artifacts are implementations which adhere to the artifact they are based upon. For more information about NLP tasks, see section 3.2.

3.1.1 GATE

The General Architecture for Text Engineering (Cunningham et al., 1996; Cunningham, Bontcheva, Peters, & Wilks, 2000) is a family of artifacts providing various services and pieces of knowledge. The major elements of the GATE family are (Cunningham, Tablan, Roberts, & Bontcheva, 2013):

- **GATE Architecture**: A high-level organizational picture of language processing software composition.
- **GATE Developer**: An integrated development environment (IDE) for language processing components.
- **GATE Cloud**: A cloud solution for hosted large-scale text processing.
- **GATE Embedded**: An object library optimized for inclusion in diverse applications giving access to all the services used by GATE Developer and others. Also referred to as the GATE Framework. See figure 3.1 for the composition of the framework.

Name	Type	Description
General Architecture for Text Engineering (GATE)	General	A family of artifacts describing and implementing the GATE approach to NLP.
Unstructured Information Management Architecture (UIMA)	General	An architecture describing how software systems can analyze large volumes of unstructured information in order to discover knowledge that is relevant to an end user.
Named Entity Recognition and Disambiguation Framework (NERD)	General	A NLP framework unifying 10 different NLP services publicly available on the web.
Health Information Text Extraction (HITEx)	GATE	An open-source natural language processing (NLP) software application, consisting of the collection of Gate plugins that were developed to solve problems in medical domain
clinical Text Analysis and Knowledge Extraction System (cTAKES)	UIMA	An open-source natural language processing system for information extraction from electronic health record clinical free-text
ClearTK	UIMA	ClearTK is a framework for developing machine learning and natural language processing components within the Apache UIMA.
Health Cyber-Physical System (Health-CPS)	General	A cloud-based system depicting the extraction of data from multiple sources through adapters, storing and managing them in the cloud, and disseminate them through an API and/or interface.

Table 3.1: The analyzed NLP frameworks, architectures and systems.

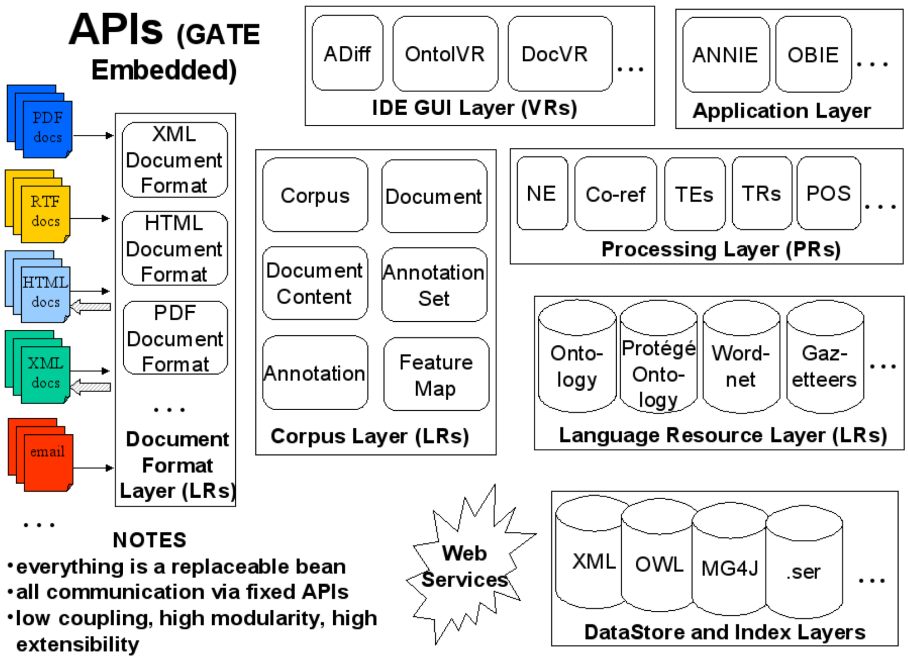

Figure 3.1: The elements of the GATE Framework. (Cunningham et al., 2013)

The collection of tools GATE embeds use Java as their main programming language.

The rationale of the GATE framework is that researchers should not be focusing on data storage and display, loading data into the processes, initiating and administering the processes and passing data between processes and machines. Instead, the focus should be more on the process itself. (Cunningham et al., 2000).

GATE Embedded, the framework a lot of other NLP solutions use, does adhere to various architectural principles. These are (Cunningham et al., 2013):

- **Neutrality**: No NLP approach is excluded
- **Re-use**: Emphasizing reuse and interoperation with related systems, and avoiding reimplementation wherever possible.
- **Componentisation**: Almost everything in GATE is modeled as a component, and the various component sets are all userextendable.
- **Multiple usage modes**: Allowing access to the GATE components through various means, like the GATE Developer IDE and the GATE Framework.

3.1.2 UIMA

The Unstructured Information Management Architecture(UIMA) (Ferrucci & Lally, 2004) was researched and developed by IBM back in the early 2000's. The primary focus of the research was Natural Language Processing, engaging in activities like natural language dialog, named entity recognition, document classification and bioinformatics. UIMA's scope goes beyond NLP: one could integrate structured-format databases, images, and multimedia, and any arbitrary technology. (Nadkarni

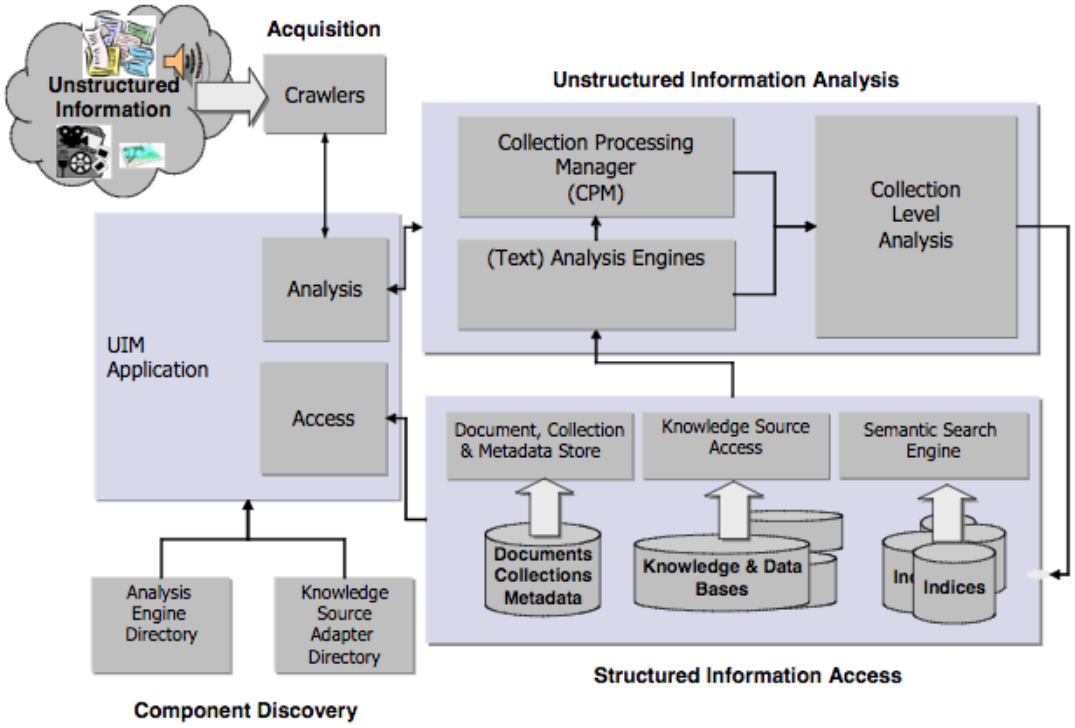

Figure 3.2: The high level architecture of UIMA (Ferrucci & Lally, 2004)

et al., 2011) The UIMA is the basis for the natural language processing capabilities of IBM Watson. Figure 3.2 presents the UIMA framework.

The UIMA framework was adopted by the Apache Foundation (Apache, 2011) and renamed Apache UIMA. This adoption is an implementation of the UIMA framework. It includes runtime frameworks in Java and C++, API's and tools for implementation of UIMA components. The Apache adopted UIMA framework can be seen in figure 3.3.

3.1.3 NERD Framework

The Named Entity Recognition and Disambiguation (NERD) framework (Rizzo & Troncy, 2012) is a framework that unifies the output of 10 different NLP extractors publicly available on the web.

NERD is a web application plugged on top of various NLP tools. The dissemination of the information follows the REST principles by exchanging the information through a JSON or XML format. The framework can be seen in figure 3.4.

The framework defined its own ontology for the data. It extracts concepts from different sources which all use different schema types (e.g.: ontology for DBpedia Spotlight, taxonomy for AlchemyAPI, flat types for OpenCalais). A concept is included in the NERD ontology as soon as at least two sources report this concept.

Figure 3.5 presents the results of the tests of the NERD Framework on a 1000 news articles of The New York Times. It shows the amount of entities per service provider. While it is very interesting to see there is a large difference in numbers between the various service providers, it is even more interesting to note that some categories are recognized by some, but not all service providers. This is one of the

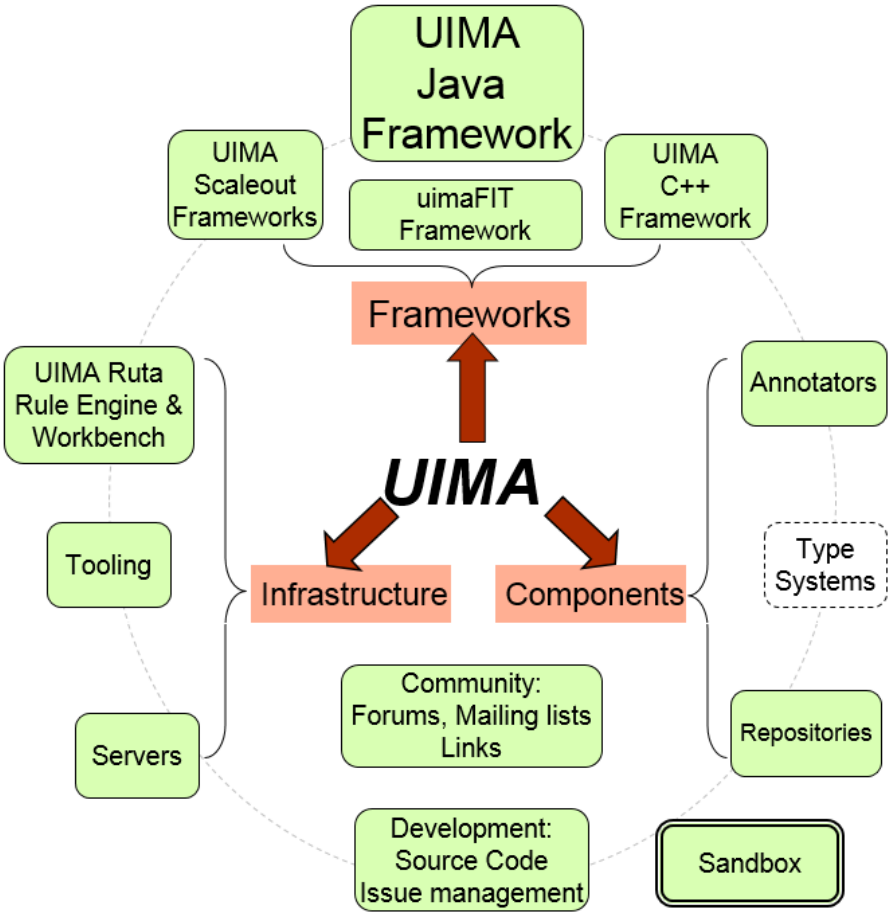

Figure 3.3: The high level architecture of the Apache implementation of UIMA (Apache, 2011)

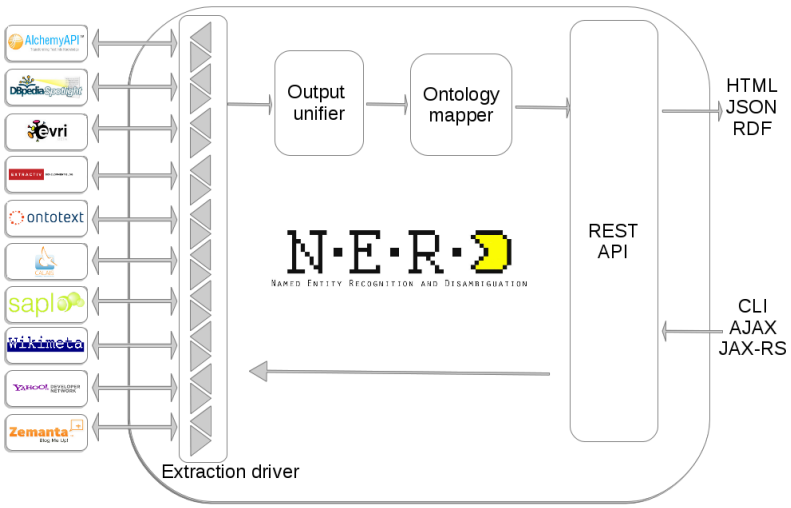

Figure 3.4: The NERD Framework (Rizzo & Troncy, 2012)

strengths of the NERD framework: using multiple sources of information extraction algorithms to get a more complete final result. This method is called information fusion. More information about this can be found in chapter 3.4

	AlchemyAPI	DBpedia Spotlight	Evri	Extractiv	OpenCalais	Zemanta
Person	6,246	14	2,698	5,648	5,615	1,069
Organization	2,479	-	900	81	2,538	180
Country	1,727	2	1,382	2,676	1,707	720
City	2,133	-	845	2,046	1,863	-
Time	-	-	-	123	1	-
Number	-	-	-	3,940	-	-

Figure 3.5: The NERD Framework (Rizzo & Troncy, 2012)

3.1.4 General clinical NLP system

Doan, Conway, Phuong, and Ohno-Machado (2014) designed an architecture for a general clinical NLP system. This system includes two main components: background knowledge and the NLP framework. The background component contains ontologies, domain models, domain knowledge and trained corpora for the natural language processing. The framework includes processes like tokenization and Part of Speech tagging (low-level processes) and NER and relation extraction (high-level processes). More information of some of these techniques can be found in section 3.2. The architecture system is presented in figure 3.6.

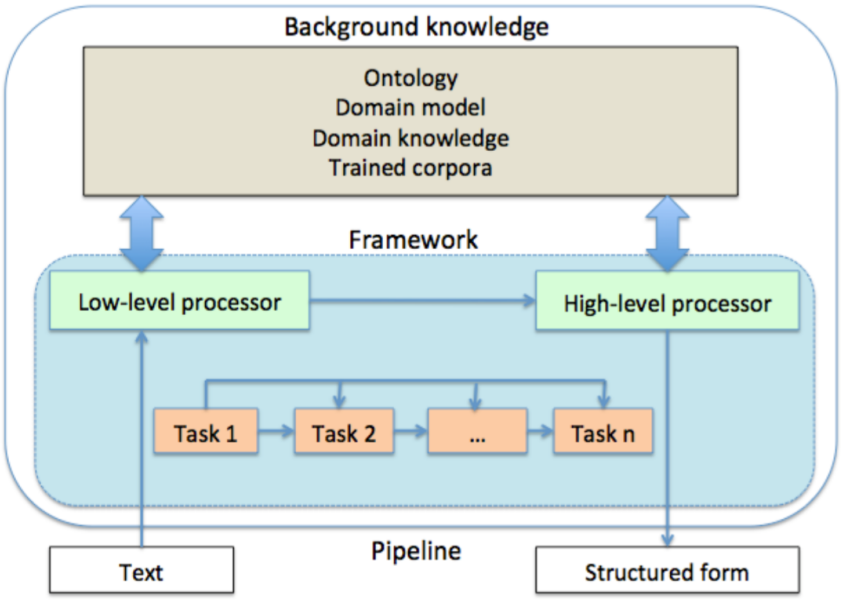

Figure 3.6: A general clinical NLP system as described by Doan et al. (2014)

The system is a summary of multiple other NLP frameworks. Frameworks include but are not limited to HITEx (section 3.1.5) and cTAKES (section 3.1.6).

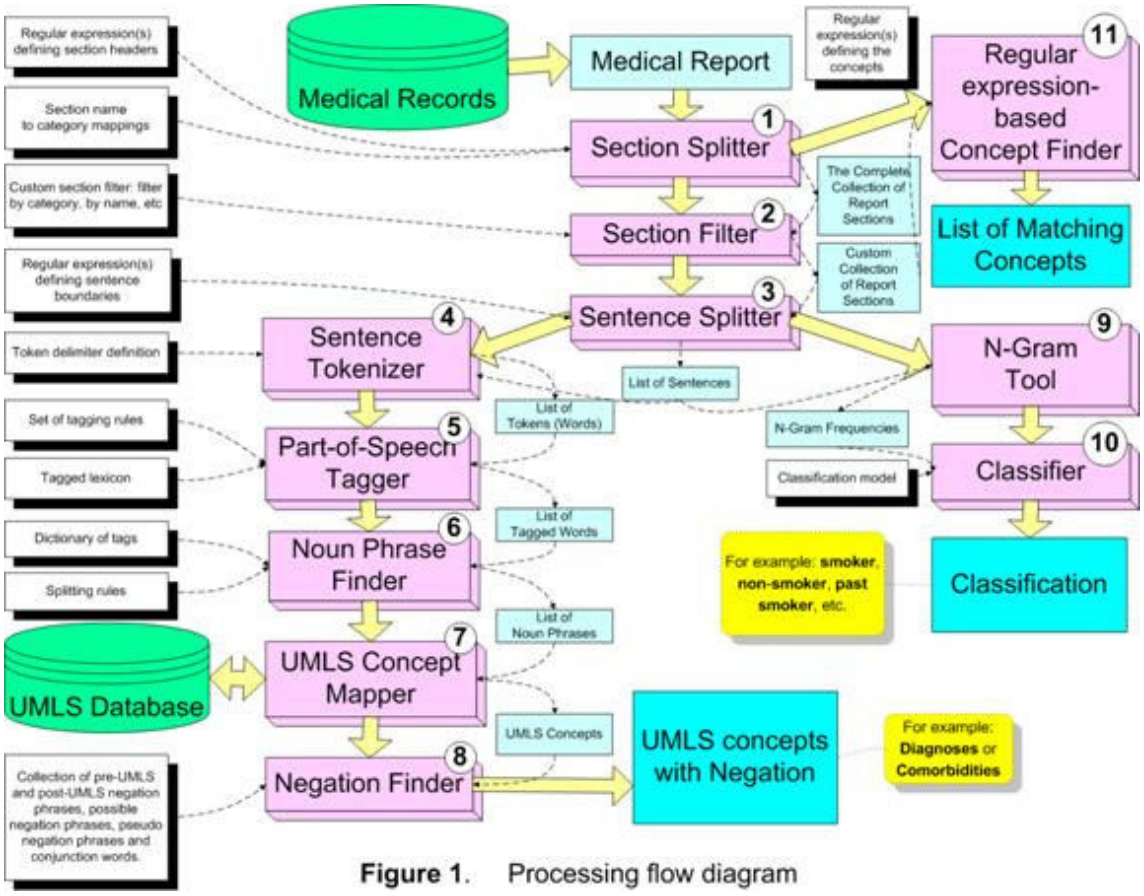

Figure 3.7: The HITEx system architecture

3.1.5 HITEx

The Health Information Text Extraction (HITEx) is an open source NLP system. The system uses a set of NLP components known as CREOLE (a Collection of REusable Objects for Language Engineering) such as sentence splitting and Part of Speech tagging. Other components, like the UMLS Mapper, are plug-ins to the HITEx system.

3.1.6 cTakes

Another system is the clinical Text Analysis and Knowledge Extraction System (cTAKES) Savova et al. (2010) initiated by a Mayo-IBM collaboration in 2000 (Doan et al., 2014).

Background knowledge: cTAKES uses trained corpora from Mayo clinic data, utilizing the UMLS as main background knowledge. Where the trained corpora are used for sentence splitting and tokenizing, the UMLS is used for Named Entity Recognition (NER).

Framework: cTAKES applies a number of rule based and machine learning NLP tasks. Tasks included are: Tokenization, Normalization, POS Tagging and Named Entity Recognition. The complete cTAKES NLP pipeline can be seen in figure 3.8.

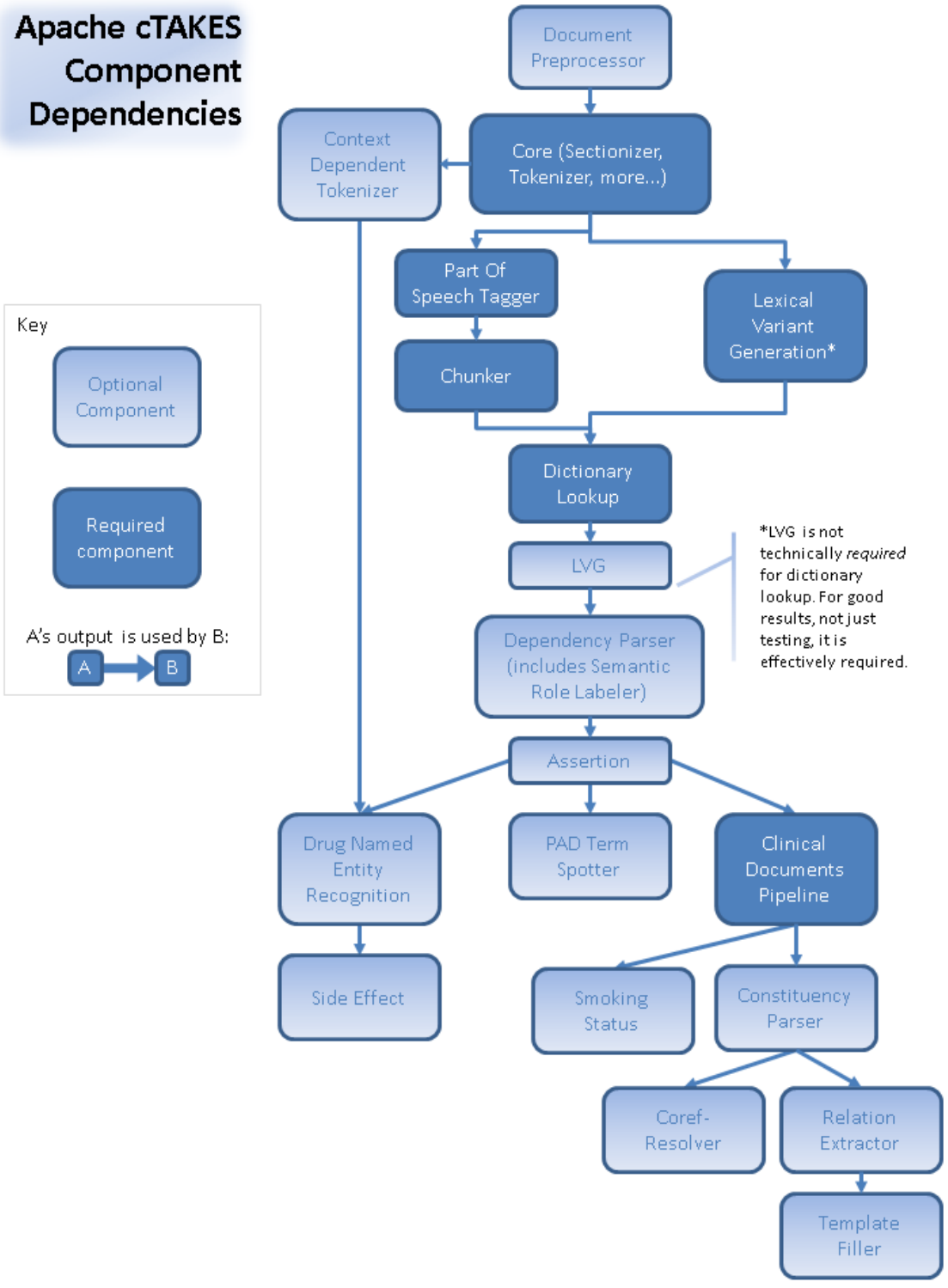

Figure 3.8: The cTAKES pipeline

3.1.7 ClearTK

ClearTK is a framework for developing machine learning and natural language processing components within the UIMA framework. (Ogren, Wetzler, & Bethard, 2008) ClearTK was developed by the Center for Computational Language and Education Research (CLEAR) at the University of Colorado at Boulder. The software is a framework that supports statistical NLP by providing a rich feature extraction library, interfaces to popular machine learning libraries, and a set of components for tackling NLP tasks.

The framework has a high focus on the technical implementation of NLP tasks, especially NLP tasks which are applied using machine learning techniques. It provides a common interface and wrappers for popular machine learning libraries such as SVMlight, LIBSVM, LIB-LINEAR, OpenNLP MaxEnt, and Mallet.

3.1.8 Health-CPS

Zhang et al. (2015) propose the Healthcare Cyber-Physical System (Health-CPS). This cloud based data oriented system suggests the extraction from multiple sources through adapters, storage and management of data in the cloud and a user interface / API to make the data accessible. Zhang et al. propose a single system architecture (Figure 3.9) which can achieve this, consisting out of three layers:

Data Collection Layer: The layer responsible for the collection of data from various "nodes".

Data Management Layer: The layer responsible for the storage and scaling.

Application Service Layer: The layer containing the software to access / retrieve the data, management platforms and development platforms.

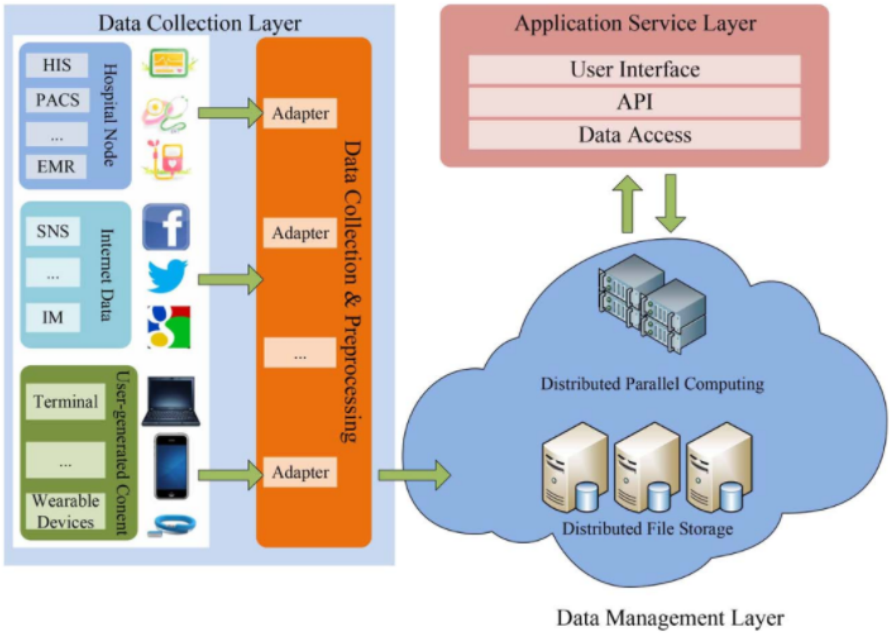

Figure 3.9: Health-CPS Architecture (Zhang et al., 2015)

This architecture provides a structured fundamental approach on how to tackle the problem of information extraction and making it available to the public. While it is not a framework that focuses specifically on NLP, it does utilize in the cloud deployment.

3.2 NLP in Information Extraction

Information Extraction is one of the most important tasks of Text Mining (Aggarwal & Zhai, 2012). Back in 1997, Cardie already defined information extraction as:

"An information-extraction system takes as input an unrestricted text and 'summarizes' the text with respect to a predefined topic or domain of interest: It finds useful information about the domain and encodes the information in a structured form suitable for populating databases."

Cardie defined the output as structured information suitable for databases. Abbott(2013) defined information extraction as:

"The Identification and extraction of relevant facts and relationships from unstructured text; the process of making structured data from unstructured and semi-structured text."

Both definitions describe the process of getting something useful out of a large body of unstructured data which potentially contains a lot of useful data. The general aim of information extraction is to discover structured information in unstructured or semi-structured text (Aggarwal & Zhai, 2012), most commonly to improve processes or decision making. Applying it to unstructured or semi-structured data may however:

- Decrease the amount of data
- Decrease the amount of useless data
- Structure the unstructured/semi-structured data
- Restructure it to a form which can easily be analyzed for secondary use.

There are various perspectives on what unstructured data is exactly. While many will agree that a document full of text is in fact unstructured data, others suggest that from a linguistic perspective all documents and text bear some kind of semantic or syntactical structure. (Feldman & Sanger, 2007) This thesis will consider any data not in fact or relational format and which can not be used for direct analysis to be either unstructured or semi structured.

Extraction of information out of unstructured data is done through NLP. NLP envelops a set of techniques to get meaning out of a text. These techniques are described in section 3.2.1. We define NLP as the practical approach for the goal of extracting information.

3.2.1 NLP Tasks

The process of NLP is one that knows several tasks. Different algorithms use a different collection of tasks to extract information from unstructured data. All discussed frameworks in section 3.1 use one or more NLP techniques, either in order or parallel. The usage of specific techniques may vary with the type of unstructured data. NLP tasks can be divided into two categories: Syntactical and Semantical. Some of the core NLP tasks are: (Abbott, 2013; Collobert et al., 2011; Hotho, Nürnberger, & Paaß, 2005; Nadkarni et al., 2011):

Part of Speech (PoS) Tagging: Tag all words in the data with the correct type. Tag all verbs as verbs, nouns as nouns, etc. The amount of word types vary per technique. While 89% of the English words only have one part of speech (unambiguous), the words which have two or more parts of speech can be disambiguated by rules or probability. A frequently used model which does this is the Hidden Markov Model. (Martin & Jurafsky, 2000; Stolcke & Omohundro, 1993)

Named Entity Recognition: Classifying words into predefined categories. For example, classifying the words "Mr. Smith" as a person or "in October 2016" as a Time Indication.

Tokenization: The conversion of a range of characters into tokens. Many words are equal to one token. However, some words include special characters making this process more difficult. For example, a dash in a word because the line ended splits up a word but should be treated as one token.

Stemming: Transforming words into their "stem" version. Removing plurals, all verbs go to their stem, etc. Stemming does not look at the part of speech tag of a word.

Lemmatization: Combining all the different forms of a word and identifying them as one single item, called the Lemma. While similar to stemming, lemmatization does look at the part of speech tag and only includes the forms which apply to that PoS tag. The lemma corresponds to the singular form in the case of a noun, the infinitive form in the case of a verb, and the positive form in the case of an adjective or adverb. (H. Liu, Christiansen, Baumgartner, & Verspoor, 2012) For example, "walk" is the lemmatization of the words "walk", "walking", "walked", etc.

Removing stop words: Removing stop words like "a", "an","then" and "the" as they rarely provide useful information. This should be treated with care, as some words do add value, for example by negation. (Barbantan & Potolea, 2014)

Dictionaries and Lexicons: By using a dictionary or lexicon, one can find the correct word from an abbreviation, synonym or foreign language.

Word Sense Disambiguation (WSD): The technique for trying to solve the ambiguity between two exactly the same words with the same part of speech, but a different meaning. For example, the word "bank" can be either a financial institution, the side of a river/lake or furniture one sits on. "Bank" is also used a verb, e.g. to put money in a bank. It is essential to disambiguate these words to discover the semantics of the text.

Sentiment Analysis: This technique uses other techniques to define the sentiment of a text. Sentiment Analysis usually return a value between -1 (negative

sentiment) and 1 (positive sentiment). The major indicators for sentiment are sentiment words. (B. Liu, 2012) Words like "good", "wonderful" and "amazing" will positively affect the sentiment, while words like "bad", "evil" or "poor" clearly contain a negative load and thus will negatively affect the sentiment.

While this list might not include every technique available, these are the most fundamental and mostly used techniques in most IE methods. Another division of NLP tasks can be low level NLP tasks and high level NLP tasks. Low level NLP tasks include tasks like tokenization and part of speech tagging. High level NLP tasks are tasks that require several low level NLP tasks to function, like sentiment analysis and named entity recognition.

3.2.2 Rule based and Statistical based NLP

There are two major approaches for natural language processing: the rule based approach and the statistical approach.

The rule based approach is quicker and cheaper to implement but far older and less accurate approach to natural language processing compared to the statistical approach. (Brill, 1992; Nadkarni et al., 2011) The rule base takes a set of rules and simply applies it to the text. A rule can be as simple as a translation of a single token, or a regular expression. (Nadkarni et al., 2011; Friedl, 2002) Terms matching that specific word or regular expression trigger an action which defines the output. One can think of exporting the term combined with a category, increasing or decreasing the sentiment of the document, etc.

Rules may also apply to multiple words in a sentence, defining the interpretation of a specific term in a sentence based on previous occurrences of specific terms. A good example of this is NegEx (see section 3.2.4), which interprets a term as negated or not depending on occurrences of specific "que" words in the whole sentence.

Rule based approaches are, however, not always adequate. NLP must ultimately extract meaning (semantics) from a text: PoS tags, relations, etc. Rule based approaches can be extended to cover these kinds of functions, but unfortunately this quickly leads to unmanageable numbers of rules, conflicting rules, unexpected results or even ambiguous results. (Nadkarni et al., 2011)

A far more complex method of NLP is by using a statistical based approach. The statistical approach calculates the probability for specific parts of the text, trying to find the meaning of that part with the highest probability. The probability of a text is calculated using a language model. (Ng, 2008)

One of the more used models is the Hidden Markov Model (HMM). It has become the method of choice for modeling stochastic processes and sequences in applications such as speech and handwriting recognition. (Fine, Singer, & Tishby, 1998; Nag, Wong, & Fallside, 1986; Rabiner & Juang, 1986) The theory of the HMM is a difficult one and not the focus of this thesis. However, one can quickly describe it as the probability of a specific word or tag based on the previous words in that specific sentence. An easy example would be the sentence "The patient died". The HMM would calculate the probability of the term "patient" after the word "The", and the probability of the word "died" after the words "The patient". This can not only be applied to finding the probability of the semantics of a sentence, but also to finding part of speech tags in sentences. For example, based on the previous word "The",

what is the probability of the word "patient" being a noun versus the probability of it being a verb.

Supervised vs. Unsupervised Learning One of the major time consuming tasks of applying statistical models is that it requires an annotated training data set before it can function correctly. Using some sort of training data classifies it as supervised learning. There is a major difference between NLP tasks that apply supervised learning (SL) and those that apply unsupervised learning (UL). (Kotsiantis, Zaharakis, & Pintelas, 2007)

Supervised learning uses a set of training data. Before the NLP task is applied to the actual data set, it is first applied to a similar data set which has been correctly tagged before. By using this correctly tagged data, the NLP task "learns" about the data, finding the probabilities of specific words, tags, etc. Next, it can apply this to another data set resulting in a much more accurate extraction of information.

Unsupervised learning is the opposite of supervised learning. The target data set is directly presented to the tasks, and the tasks has to learn itself what the concepts, PoS tags, etc are. The aim is that unsupervised learning gets more accurate as it is applied to more data, as it learns more about the data and thus can apply what it learned to other, new data sets.

3.2.3 Natural Language Processing Output

We established that NLP takes unstructured data as an input. We've taken a look at some of the NLP tasks. There are multiple types of outputs a IE method can deliver:

Term: The most basic form of information extraction, the extraction of key terms. These terms contain the major information components of the unstructured data and can be very useful when extracting information like names, dates, values, or anything with a default format. This type of output can be accomplished with pattern matching and does not require any form of machine learning. However, machine learning techniques may improve the accuracy.

Entities: As a result of Named Entity Recognition (NER), an entity can best be described as a term annotated by a category. An entity category can be anything ranging from a something simple as a date to something more complex as a company name. As an example, lets take the sentence:

Mary had no history of diabetes or family history of cardiac disease

Varying by extractor, this sentence may be annotated as follow:

$Mary_{[Person]}$ *had no history of* $diabetes_{[MedicalCondition]}$ *or family history of cardiac* $disease_{[MedicalCondition]}$

Entities are found by either comparing it to a large database or by using machine learning. One of these databases is DBpedia. (*DBpedia*, 2017) Note that entities are not the same as part of speech tags. While part of speech

tagging is applied to all words in a sentence and limits itself to the tags defined by the human language, Entities are only matched on specific words in a sentence and are limited to the categories defined by the developer of the specific Entity Recognition process.

Relations: A (binary) relation between two words or concepts in the same sentence. Aggarwal and Zhai (2012) provide an example with the sentence:

"Mark Zuckerberg is the co-founder of Facebook"

The relation expected to be extracted from this sentence is:

"CofounderOf(Mark Zuckerburg, Facebook)"

A relation can also be between two words from different sentences, however finding these is much more difficult.

Negations: Negations are a special kind of relation. A word in a sentence can negate another word in a sentence. Lets take the following sentence as an example:

Mary had no history of diabetes or family history of cardiac disease

When no negations are taken into account, the IE process on this sentence may conclude that Mary has diabetes or a family history of cardiac disease. However, these statements are negated by the word "no". Therefore, the information extracted from this sentence should be perceived as negated. For more information about Negation, see chapter 3.2.4.

3.2.4 Negation

Negation is an important aspect of Natural Language Processing and Information Extraction. It makes the major difference in semantics in sentences and individual words. A study of negation has shown that clinical observations are frequently negated in clinical narratives. (Chapman, Bridewell, Hanbury, Cooper, & Buchanan, 2001a; Mehrabi et al., 2015)

Detecting a negation in a single sentences like *"Mary has no Diabetes"* is relatively simple. A simple trigger word like "no" might result in the next word being interpreted as negated. This can already be achieved by matching regular expressions. However, it quickly becomes more difficult to detect negations once the complexity of the sentences increases. Lets take a look at a more complex version of the above sentence: *"Mary had no history of diabetes or family history of cardiac disease"*. A human will quickly conclude that Mary had no diabetes, nor a family member who ever had a cardiac disease. A simplistic approach, as described above, will not work on this sentence as that would result in that "Mary had no history". This sentence can however be negated by using a little more complex approach.

One of the more commonly used algorithm is NegEx. (Chapman, Bridewell, Hanbury, Cooper, & Buchanan, 2001b) NegEx is based on the "baseline" IPS (Identifying Patient Subgroups) system, developed by the University of Pittsburgh.

(Aronis, Cooper, Kayaalp, & Buchanan, 1999) This baseline algorithm searches for six phrases that indicate a negation coming up and negates all UMLS terms (See section 3.3.2 for more information on the UMLS) after that negation .(Chapman et al., 2001b) NegEx expands on this baseline by adding another 35 negation phrases These 35 contain phrases that can identify double negations, modified meanings and ambiguous phrasing. The IPS System marks all words after the negation term as negated. NegEx first searches for any negation term, then looks for any UMLS term within a range of 6 words. Depending on the negation term, this UMLS term lookup may be either only forward, backwards or both.

3.3 Interoperability & Standardizations

Interoperability concerns the exchange of data between two different systems. (Benson, 2012a; HIMSS, 2017) Interoperability is an important element in the entire framework as there is a lot of communication between internal and external systems. As Mead (2006) states:

"If each system-to-system connection requires a separate, non-standard interface, the number of interfaces required to connect n systems is roughly $(n^2)/2$*. Thus, full connectivity of 20 information systems requires approximately 200 interfaces; for 40 information systems, the number jumps to around 800."*

Large health care organizations can easily have over a 100 different information systems, which even when they do not all have to be connected result in an immense amount of interfaces required. To reduce the amount of interfaces required, data interchange standards are essential. Multiple standards are already formed for the transferring (and storing) of data in modern day applications.

Three different types of interoperability are recognized (Geraci et al., 1991; HIMSS, 2017):

- **Foundational:** When two systems can exchange data without explicitly having the ability to do so. This happens mostly when two systems use identical data.
- **Syntactic:** When two systems can exchange data in a specific predefined format.
- **Semantic:** When two systems can exchange data, understand the data and use the data. Semantic interoperability takes advantage of both the structuring of the data exchange and the codification of the data including vocabulary so that the receiving information technology systems can interpret the data. Often these standardizations are context specific.

Our prototype will mostly make use of syntactic and semantic types of interoperability. However, the framework also has various internal elements which require communication. We identify these custom data object as foundational interoperability.

3.3.1 Syntactic Interoperability

JSON and XML are the most used formats for Syntactic Interoperability. While the syntaxes are not the same, they do have the same approach on the storage. They both implement an hierarchical approach, where you start with a list of keys which all have a value. These values can be primitive types like integers or strings, or can be another list of key values. As all kinds of unstructured and structured data can be serialized to text, almost every form of data can be used as a value. E.g.: an image can also be serialized into a string.

Because JSON and XML follow a rather strict markup, they can easily be serialized into a string. This string can then be sent to another system, where it can be easily transformed into the systems model. While used by a lot of applications currently on the market, they do not provide any advantages or disadvantages for clinical data.

3.3.2 Semantic Interoperability

There are dozens of standardizations for semantic interoperability. In this section we depict only a small amount of standardizations which we feel are important for this project, either because we will use them in our prototype or because it is important for the overall goal. As mentioned before, semantic standardizations are often context specific. All standardizations discussed here are clinical context specific.

Every system uses its own terminology for clinical concepts which can almost never be exchanged with other systems. Meulendijk, Spruit, Lefebvre, and Brinkkemper (2017) identified two approaches to share concept between two systems using different terminologies:

- Merge the concepts to form one single meta-terminology.
- Map the concepts from one terminology to another.

Both approaches have been used by a large amount of systems to improve semantic interoperability. Some systems even implement both approaches, serving both as a meta-terminology and a mapping between terminologies.

UMLS "The Unified Medical Language System is a repository of biomedical vocabularies developed by the US National Library of Medicine. The UMLS integrates over 2 million names for some 900 000 concepts from more than 60 families of biomedical vocabularies, as well as 12 million relations among these concepts." (Bodenreider, 2007) The technical implementation of UMLS is a set of files and software bridging various biomedical vocabularies and standards. These files can be imported into a database for easy access in information systems.

The UMLS consists out of three tools (Bodenreider, 2007):

- Metathesaurus: Terms and codes from many vocabularies, including SNOMED CT (section 3.3.2), ICD-10-CM (section 3.3.2) and RxNorm (section 3.3.2).
- Semantic Network: Broad categories (semantic types) and their relationships (semantic relations).

- SPECIALIST Lexicon and Lexical Tools:Natural language processing tools.

The UMLS has its own meta-terminology which integrates mappings to countless other terminologies. Each UMLS concept has an unique identifier (CUI) and relation with one or more atoms. An atom is a building block for the UMLS concept, being data from various other sources. For example, an atom can be the ICD-10 data about "headache", which is linked to the UMLS concept of "headache".

The UMLS also implements an online API. This API is accessible after being granted access by the U.S. National Library of Medicine, Department of Health and Human Services. This API, while relatively slow, allows easy access to all the information the UMLS yields.

HL7 HL7 knows a version 2 and 3 of which only the former is commonly used. HL7 Version 2 is the most widely used health care interoperability standard in the world. It is used in over 90% of all hospitals in the USA and is widely supported by health care IT suppliers worldwide. (Benson, 2012b)

Health Level 7 Clinical Document Architecture (HL7 CDA) is a document markup standard that specifies the structure and semantics of clinical documents. A CDA document is a defined and complete information object that can include text, images, sounds, and other multimedia content.

This standardization is kind of special, as HL7 covers both syntactics and semantics. It envelops both a data structure and a semantic meaning behind the data in the structure. However, as version 3 of the HL7 interoperability standard uses XML to build its data structure, we consider HL7 as the semantic standard and not the syntactic.

SNOMED Clinical Terminology SNOMED Clinical Terminology (SNOMED CT) or simply SNOMED is the most comprehensive and precise, multilingual health terminology database in the world. (*Overview of SNOMED CT*, 2016) Every entry consists out of four primary core components:

- Concepts: A combination of a name with an unique identifier. These are the actual medical terms.
- Descriptions: The description of the concept, divided into "Fully Specified Names", "Preferred Terms" and "Synonyms"
- Relationships: A concept's relationship to another concept.
- Reference sets: A group of Concepts or Descriptions.

ICD-10 The International Classification of Diseases and Related Health Problems (ICD-10) is the international standard for reporting diseases and health conditions. It defines the universe of diseases, disorders, injuries and other health conditions in a comprehensive, hierarchical list. (*ICD purpose and uses*, 2016) ICD-10 is especially useful for sharing and using the clinical data between systems, as the code is language/country independent.

The format of ICD-10 can be found in table 3.2. (*ICD-10 Basics*, 2015)

Two example codes can be found in table 3.3.

Character Location	Description	Format
1	Disease Category	Alphabetic
2-3	Disease Category	Numeric
Between 3 and 4	A seperator dot	A seperator dot
4-6	Detail about etiology, anatomical site and severity	Any combination of numeric and/or alphabetic
7	Detail about etiology, anatomical site and severity	Any combination of numeric and/or alphabetic

Table 3.2: The structure of the ICD-10 Code.

Code	Description
I25.110	Atherosclerotic heart disease of native coronary artery with unstable angina pectoris
S72.044G	Non-displaced fracture of base of neck of right femur, subsequent encounter for closed fracture with delayed healing

Table 3.3: ICD-10 Examples

Using this code standard allows for an in detail description about every known disease or injury which computers are easily able to store, retrieve and transfer. Combining SNOMED with ICD-10 allows the transfer of universally standardized clinical data. The previously described STRIPA system (section 1.1) uses ICD-10 coding internally.

RxNorm RxNorm is two things (*RxNorm Overview: What is RxNorm?*, 2017)

- A normalized naming system for generic and branded drugs. (RxNorm)
- A tool for supporting semantic interoperation between drug terminologies and pharmacy knowledge base systems. (SAB=RXNORM)

One could see RxNorm as another standard similar to ICD-10 or SNOMED. However, RxNorm also has focus on linking different medical terms or standards to each other to improve communication between systems handling clinical data. It receives drug names and descriptions from 14 different sources, including SNOMED, UMLS and ICD-10. For example, various data sources have product names like:

- Naproxen Tab 250 MG
- Naproxen 250mg tablet (product)
- NAPROXEN@250 mg@ORAL@TABLET
- Naproxen 250 MILLIGRAM In 1 TABLET ORAL TABLET
- NAPROXEN 250MG TAB,UD [VA Product]

They describe exactly the same drug, but with various names and completeness. RxNorm combines these concepts by assigning an RxNorm normalized name, with the structure ingredient, strength and dose form: ”Naproxen 250 MG Oral Tablet”.

3.4 Information Fusion

Multi-sensor Data / information fusion is about combining information of multiple sensors by using related information from large databases, into one single better representing set of data. As Waltz and Llinas (1990) defined:

Information Fusion encompasses theory, techniques and tools conceived and employed for exploiting the synergy in information acquired from multiple sources (sensor, databases, information gathered by human, etc.)

The concept of multi-sensor information fusion is based on what both human and animals have been doing for centuries: combining sensors to increase the chance of survival or to get a better quality of life. E.g. figuring out if something is edible can be done by using vision, smell and taste which combined, generally, gives a more accurate result than using any of these sensors alone. (D. L. Hall & Llinas, 1997) As Waltz and Llinas describe it:

The objective is that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness etc.) than it would be possible if any of these sources were used individually).

Multi-sensor information fusion is widely applied in IT solutions, stretching from manufacturing processes to autonomous vehicles. In our tool it will be used to fuse the entities that were extracted by various NLP services.

There are a number of issues that make data fusion a challenging task. The majority of these issues arise from the data to be fused, imperfection and diversity of the technologies, and the nature of the application environment (Khaleghi, Khamis, Karray, & Razavi, 2013), like.

- **Data alignment/registration:** sensor data must be transformed from each sensor's local frame into a common frame before fusion occurs.
- **Outliers and spurious data:** the uncertainties in sensors arise not only from the impreciseness and noise in the measurements, but are also caused by the ambiguities and inconsistencies present in the environment, and from the inability to distinguish between them.
- **Data imperfection:** data provided by sensors is always affected by some level of impreciseness as well as uncertainty in the measurements. Data fusion algorithms should be able to express such imperfections effectively, and to exploit the data redundancy to reduce their effects.

While these challenges talk about sensor data fusion, these challenges apply to data fusion of other data as well.

One of the major information fusion approaches is that of the Joint Directors of Laboratories (JDL). (Liggins II, Hall, & Llinas, 2017; Snidaro, García, & Llinas, 2015) JDL's model depicts different levels of information fusion processing (figure 3.10):

- **Level 0:** Information fusion at the sensor level. This level concerns fusion on syntactical level, not semantical).

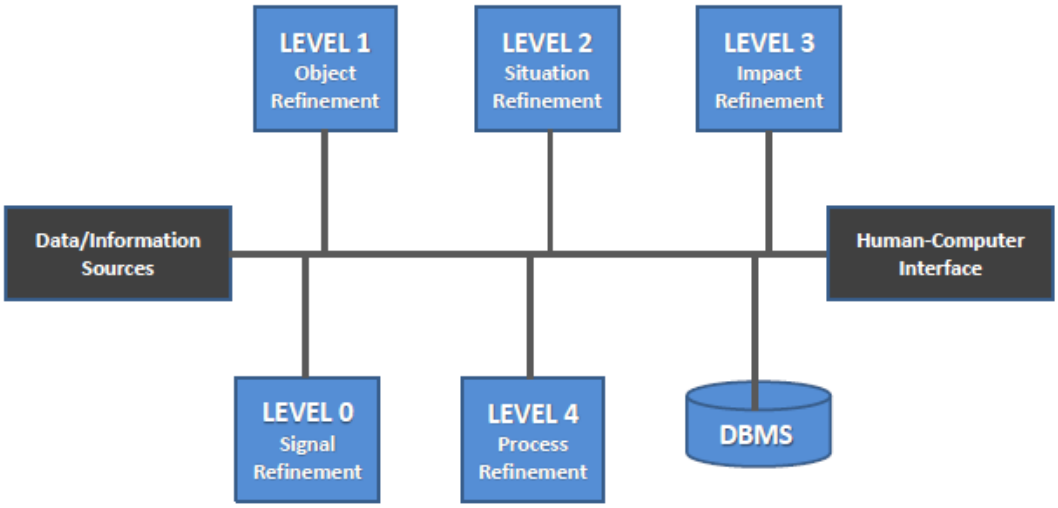

Figure 3.10: The model presented by the Joint Directors of Laboratories. (Liggins II et al., 2017; Snidaro et al., 2015)

- **Level 1:** Concerns about estimating what individual entities are on a sensor level.
- **Level 2:** This level parses entities and estimates, based on their attributes and relationships with other entities.
- **Level 3:** Estimation of the impact of a reactive environment on the entities.
- **Level 4:** The ability to adapt data acquisition by using the previous levels. This level is part of the Resource Management.

An entity, in the context of information fusion, is a single 'measurement' in a single point of time. Level 0 and level 1 will be done by our NLP services, providing us with unique entities. Level 2 is to be of significance, as we are trying to combine two entities based on the relation between attributes. Level 4 will be implemented in our prototype.

Several methods, with each method having several techniques for each level, exist: Sensor Characterization using Weighting (Level 0), Data Association techniques using Confidence-based association (Level 1) and Knowledge Representation using ontologies (Level 2). (Snidaro et al., 2015) For our prototype, we will use

- Weighting to characterize and set a weight for our individual NLP services.
- Ontologies combined with weighting to perform level two information fusion.

3.5 Multi-API based NLP

Multi-API based NLP (MAPI-NLP) is a natural language processing approach by using multiple external natural language processing systems, accessed by and referred to as an API. Multi-API based NLP consists out of three major elements:

- ***API:*** An application programming interface is a set of functions or methods made available to the public. In this project, an API is similar to the functions provided by one of the external NLP systems.
- ***Multi:*** Key here is the usage of multi(-API). By using multiple API's at the same time we aim to achieve a better result than just using one single API. This does however imply the use of some sort of information fusion technique.
- ***NLP:*** Natural Language Processing is a general term for a set of methods and techniques to handle unstructured data and transform it into something structured. See section 3.2 for more information about this.

By combining these three concepts we create a new approach to process natural language. Using Multi-API based NLP we can transform unstructured data into structured information based on multiple sources of linguistic knowledge bases, strengthening the overall end result. The aim of a multi-API based NLP is to perform NLP externally to safe money and time, making the NLP processes more easy to implement and the overall NLP task more efficient.

Why do we want to create a (Multi-)API approach? In the last decade, API's transfered from a "Edge Product" to a core concept of many businesses. (Smith, 2017) As Collins and Sisk (2017) state: *"The growth of APIs stems from an elementary need: a better way to encapsulate and share information and enable transaction processing between elements in the solution stack."*. With this project, we embrace this trend in API use and build a new solution based on, as Collins and Sisk stated, enabling transaction between elements in the NLP solution stack.

Insourcing vs. Outsourcing The NLP process varies on many aspects when taking the Multi-API approach instead of the Single-API approach. Another difference is between implementing NLP approaches locally or using remote NLP services through means of an API.

Local NLP are often libraries of code downloaded from the internet which can then be used by a developer in his or her favorite IDE. Most local NLP solutions are either open source or provided by academic institutes. NLP solutions created as an open source initiative or from a academic institute benefit from distributing their code as a local code, since that way others can make additions to their algorithm to improve it.

Remote algorithms, on the contrary, is an IE Algorithm library that is hosted on a remote server and is only accessible through an specific interface. This interface is referred to as an Application Programming Interface (API) , and most of the times this can be accessed by using a RESTful (Representational State Transfer) web service.

As mentioned before in section 1.2, businesses have to consider both making (insourcing) and buying (outsourcing) for developing a product or service (M. Lacity et al., 2017) to reduce costs and increase profits. Using a remote NLP service is a form of outsourcing, thus yielding the same benefits and downsides as outsourcing. An approach where outsourcing to multiple NLP services to develop one single NLP solution has not yet been seen in larger NLP projects thus far.

Most remote NLP service providers are large commercial IT companies. It is easier, safer and more profitable for commercial companies to provide a an online

service instead of distributing a local version, as they can control access to their service without their intellectual property ever leaving the building.

Advantages of using outsourcing information technology, like NLP, which are also applicable to this project are (*IT Outsourcing: The Reasons, Risks and Rewards*, 2017; M. Lacity et al., 2017):

- **Cost Reduction:** Outsourcing one or more information technology show a reduction of over 40% in costs. (Chang & Gurbaxani, 2012; M. C. Lacity, Willcocks, & Feeny, 1996; Saunders, Gebelt, & Hu, 1997) Cost reductions can primarily be found in the reduction of employees needed and the reduction of research and implementation time.

- **Access to expertise/skills:** Outsourcing specific information technologies so these functions are processed better and/or quicker than it would have been when developed in-house.

- **Lower level of technical skills required:** Using an external/remote technology or service and connecting this service to the clients IT system is a much less complex approach than developing the technology.

- **Risk reduction:** This project does not involve large amounts of risks, it is only a small scale research project. However, outsourced IT providers are always up to date of the newest technologies in their field and thus are much more capable of handling risk in their area of expertise, removing the need for us to concern ourselves about things that might go wrong implementing an NLP service.

As great as using a remote NLP service sounds, looking at these advantages, outsourcing IT services also yield some disadvantages. The two most important ones are:

- **Control is lost:** As a remote service is used, the functionality of the service is limited to what the developer of the service provides. The functionality is not owned by the client, limiting the influence the client has on it and reducing the flexibility clients sometime seek. Also, in the relatively new area of natural language processing, this means that outsourcing NLP limits one to using the available functions offered by the various NLP services.

- **Confidentiality:** When using data connected to one single person, it has to be treated carefully and securely. A major property of the security of data is the amount of people that have access to it or handle it. Sending it to other parties increase the amount of people / companies that handle the data and thus, naturally, decreases security.

As mentioned before, outsourcing NLP tasks to multiple NLP services and combine them into one large NLP extraction, is something that has not been tried before. We feel that using the strengths of outsourcing will result in lower costs, less development time and hopefully better results in the context op NLP.

4. Multi-API based NLP Framework

4.1 Framework

In chapter 3 we focused on existing frameworks, architectures and techniques. Based on a combination of previous work, we propose a new framework for natural language processing, the Multi Application Programming Interface Natural Language Programming (MAPI-NLP) framework. The framework can be seen in figure 4.1.

The framework is a combination of the NERD framework (see section 3.1.3) and the General Clinical NLP architecture (see section 3.1.4). It consists out of four major internal elements: API Processing, Structuring and standardization, Evaluation and Act. Four external elements can be found in the framework: The input of unstructured data, external NLP API's, the knowledge database and the output of structured information. The overall aim of the framework is to outsource the important NLP tasks using multiple external NLP services to transform unstructured data to structured information and disseminate it to third parties.

- *API Processing:* The function of API processing is to take the unstructured data and transform it into data that can used for data fusion and analysis. The element consists, for the context of text analysis, out of two sub-elements: Remote NLP systems and Local NLP systems. The remote NLP systems take the unstructured data, processes it and provides it to the local NLP systems.

  Once this data has passed through the remote NLP systems, it is post processed by local NLP systems. In a perfect world the processes would be integrated in the remote systems, we concluded that this is currently and most likely will always be a necessity for a good performance of this framework. In our evaluation tool, it turned out that local NLP systems had to include a negation handler. More about our implementation of the API processing element can be found in section 4.2.1

- *Structuring and Standardization:* The step of classification tries to classify all the information that was extracted during the sensors processing element. It attempts to combine, classify and structure the information. The most important task of the classification is to fuse the information gathered in the sensor processing element. As the input is information gathered from a variable amount of NLP services which theoretically can all output something different with a different certainty, it is of the utmost importance to correctly combine all this information before it is used for analysis. More about this process can be found in section 3.4. Data fusion requires a knowledge base, providing more

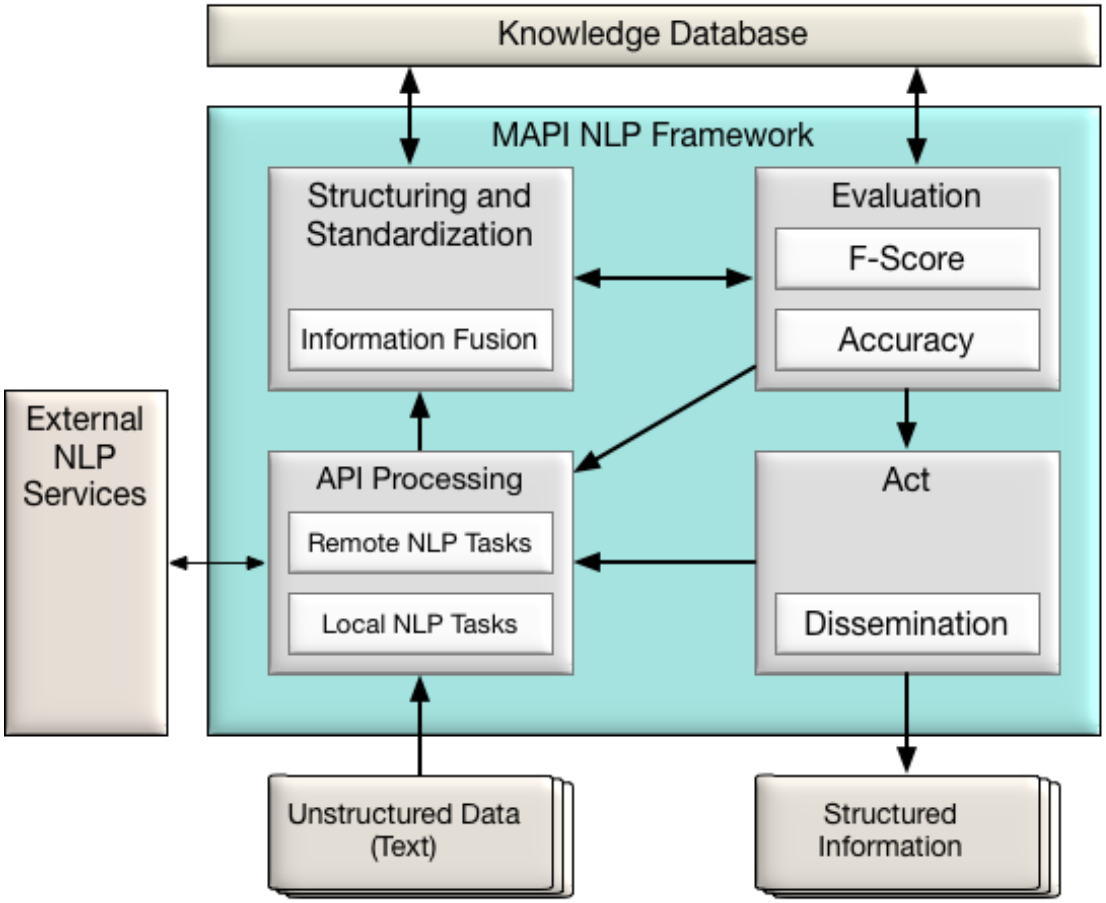

Figure 4.1: The proposed framework: MAPI-NLP

information about the extracted data / information. In this project we use a clinical knowledge based named the unified medical language system (see chapter 4.2.3 for more information).

- *Evaluation:* Once the data has been gathered, structured and fused, it is required to evaluate this data. This evaluation is performed to provide feedback about the extraction and what changes might improve the extraction. The changes may include adjusting parameters of the data fusion or changing external NLP API's. Evaluation can be done manually, but can also be deduced by the information it gets and information is received in the past.

  During this project evaluation will be accomplished by comparing the received information to ground truths of the text the extraction was done on. We build a prototype which enables to evaluate test data with this framework.

- *Act:* After evaluation of the information and possible adjustments to it are done, it is time to act! We propose to disseminate the now structured information to others for further analysis. While this step internally does not include that much, it is necessary to include in the framework as it shows the aim of the framework: transforming unstructured data to structured information and disseminating it!

This framework uses the concept of outsourcing as discussed in section 3.5. The advantages of this framework come with the concepts it includes: lower development time, lower costs and higher NLP specialization due to the fact that the NLP tasks are outsourced to professional companies working with NLP day and night.

4.2 Implementation

The next step is to actually implement the solution, starting with describing in more detail what choices we will make concerning the implementation. As the work is partially based on previous concepts and methods, some of the decisions are defined beforehand. Others will remain up until the actual implementation later in this project.

This section will first describe all the necessities to allow the creation of the implementation artifact. In chapter 3 we discussed the theory and options for the implementation of the proposition. In this section we will depict the concrete implementation decisions.

Next, the implementation we used for testing the architecture will be depicted. The implementation artifact will be an online tool that will be used for the evaluation described in section 2.4 - Evaluation.

4.2.1 Remote NLP Tasks

The world of NLP services has seen some major improvements during the last few years. Both the academic and the business community found the importance of good information extraction for both their personal interests and external interests. Many organizations provide services that allow others to use their natural language processing algorithms.

For the Extraction of Information we decided on using six different online NLP services. The six different systems are the following:

- **Watson** by IBM
- **MeaningCloud** by MeaningCloud LLC
- **Open Calais** by Thomas Reuters
- **Haven OnDemand** by Hewlett Packard
- **TextRazor** by TextRazor Ltd
- **Dandelion API** by Spaziodati

All six offer an online service for natural language processing through an API. Some offer functions that others do not. They do all include the function of entity extraction, albeit sometimes called differently. The entity extraction function might also be part of a bigger function within the NLP service.

IBM Watson - Natural Language Understanding IBM is a major player in the context of artificial intelligence. With their Watson system they are able to gather information and even knowledge out of seemingly unusable sets of data.

Natural Language Understanding (NLU) is one of the services of Watson, accessible on the IBM developer cloud called "Bluemix". IBM describes Natural Language Understanding as *"With Natural Language Understanding, developers can analyze semantic features of text input, including - categories, concepts, emotion, entities, keywords, metadata, relations, semantic roles, and sentiment."* (*IBM Watson - Natural Language Understanding*, 2017)

NLU takes any random text as input. By using a collection of NLP algorithms, it extracts terms, relations and entities from the text that are considered relevant and important. NLU can handle the following languages: English, French, German, Italian, Portuguese, Russian, Spanish and Swedish. However, it states that for some languages additional modeling in Watsons Knowledge studio might be required to get a good result. It can extract both a "Medical Condition" and a "Pharmaceutical Drug" entity type.

Technical Details NLU can be accessed through the NLU service on IBM's Bluemix platform. NLU is a collection of URL's which can be called through a piece of software that can transfer data through a HTTP protocol. One of the more common examples of such software is the cURL software package.

The URL consists out of a base URL and a method URL. The base URL for the NLU API is /citeNLU2017API
"http://gateway.watsonplatform.net/natural-language-understanding/api/v1".
The method URL varies, depending on what method of AlchemyLanguage you want to use. One of these methods, the one we are using for this project, is "/analyze". By concatenating these URL's, we get the final URL for the cURL call: "http://gateway.watsonplatform.net/natural-language-understanding/api/v1/analyze".

This API endpoint requires several parameters to function. First of all, it requires authentication parameters. As NLU is commercial software, and thus IBM wants

to make profit out of it, they have to track who is using their products. Once an account has been created, one gets a secret key to send with every API call which functions as an authentication.

Secondly, it needs the type of the response data. This can be either JSON or XML. For this project we chose JSON as a response.

Third, it requires the input as a parameter. The API can handle three types of input: Plain Text, A text in HTML format or a website URL.

Fourth, and the most important parameter, is a list of things to extract from the provided input. This list varies per API method and type of input (E.G.: A piece of plain text does not have an author). Some of the more common types are:

- authors
- concepts
- dates
- doc-emotion
- entities
- feeds
- keywords
- pub-date
- relations
- typed-rels
- doc-sentiment
- taxonomy
- title

The extraction types parameter is optional. It defaults to "Concepts, Entities, Taxonomy, Keywords".

Meaningcloud MeaningCloud is an online API which allows a developer to send a text to one of the API endpoints. MeaningCloud will then transform this data by applying one of the several services. It can applied, for example, to NER to create Entities (called "Topics" by MeaningCloud), Part of Speech tagging or sentiment analysis.

MeaningCloud can handle the following languages: English, Spanish, French, Italian, Portuguese and Catalan. It can extract both "Medical Conditions" and "Pharmaceutical Drugs".

Technical Details (Free Version) As MeaningCloud is an online API, it can be accessed through a REST API call. The base url is "http://api.meaningcloud.com/". The endpoint you connect to is equal to the MeaningCloud service you wish to use. If a developer would like to use the PoS tagging service, a connection should be made to "http://api.meaningcloud.com/parser-2.0". If a developer would like to use the Topic Extraction service, a connection should be made to "http://api.meaningcloud.com/topics-2.0".

Each requests requires the addition of an authentication key. This key is bound to one's account and can therefore be used for both billing and authentication. Each request requires additional parameters to be send, including but not limited to the text (plain text, URL to webpage, document), return format (xml / json), and service specific parameters.

The NER service returns a list of concepts combined with an entity type. The entity types are structured hierarchically. The highest entity type is called "Top", containing all recognized entities. Entities can then contain a variable amount of sub types. A type with two sub types is then representated as type > sub-type 1 > sub-type 2. For example, when a name is recognized as an entity, the type will be presented as "Top > FullName > Person"

OpenCalais *"With a market-leading ontology linked to Thomson Reuters' authorities and products, Thomson Reuters Open Calais offers the easiest and most accurate way to tag the people, places, companies, facts, and events in your content to increase its value, accessibility and interoperability."*

OpenCalais can be accessed through a free API. OpenCalais features a "premium" version, lifting several limitations the free access has and opening up more features like other types of entities the NER function can recognize. With the free version, "Medical Conditions" can be found, "Pharmaceutical Drugs" can not be found. OpenCalais can handle English, Spanish and French text.

Technical Details (Free) As OpenCalais is an online API, it can be accessed through a REST API call. OpenCalais uses just one URL: "https://api.thomson reuters.com/permid/calais". This URL requires 2 parameters: Content type and Access Token. The content type can be either text/raw, text/html, text/xml or application/pdf. The Access Token can be retrieved from the personal web portal on the OpenCalais website and is used for both authentication and billing.

The OpenCalais API extracts entity types. The free version has a major limitation: it does not recognize the "Pharmaceutical Drug" type, which is a major addition to this project. It does however recognize the types "MedicalCondition" and "Medical Treatment".

Haven OnDemand *"Haven OnDemand simplifies how you or your customers can turn virtually any data into an asset anytime and anywhere. If you are looking for a faster and easier way to tap into big data for delivering comprehensive and actionable insights, now is the time to take advantage of this cloud services platform."* (*Haven OnDemand*, 2017)

This platform, developed by Hewlett Packard, allows developers to access API's designed for rapid development, high performance and massive scalability. The platform allows for connection to web storage like Dropbox, offers function like Image Analysis and large scale handling of textual data. One of these functions is Entity Extraction.

Haven Ondemand offers both a free and a paid version. The paid version comes with a higher maximum of API requests per month, a higher API request limit per minute and a higher amount of resource units.

Haven Ondemand is able to perform text analysis services on a long list of languages as it uses its language identification function to recognize the language of the sent text. It does however not recognize all types of entities for all languages. All the major entity types like "Pharmaceutical Drugs" and "Medical Conditions" can only be recognized in English texts.

Technical Details Haven Ondemand provides a RESTful API through which their services are accessible. The requests can be done through a simple cURL request, only requiring an API key distributed by Haven OnDemand itself, the text in either a plain, HTML or file format, and the to be extracted entity types.

The API can be accessed either synchronously or asynchronously. Synchronous can be used for smaller requests, where the requests will only return once the function has finished processing. Asynchronous must be used for larger requests like video or image processing. This request will return a Job ID with which can be used to

poll whether the processing of the request has been finished or not. The endpoint of an API only changes slightly when switching between synchronous or asynchronous, changing from "sync" to "async".

The endpoint for the Entity Extraction service our implementation will be using is *" https://api.havenondemand.com/1/api/sync/extractentities/v2"*.

TextRazor *"The TextRazor API helps you extract and understand the Who, What, Why and How from your documents with unprecedented accuracy and speed."* (*TextRazor*, 2017)

This API, developed by TextRazor Ltd., allows developers to perform text analytics on a variety of textual documents. The TextRazor API offers functions like Entity Extraction, Classification and Relation Extraction. The Entity Extraction function uses a huge knowledge base of entity details extracted from multiple websources like Wikipedia, DBPedia and Wikidata. It analyses the semantic context of the sent text to disambiguate ambiguous entities. TextRazor is capable of analyzing English, Dutch, French, German, Italian, Polish, Portuguese, Russian, Spanish and Swedish.

TextRazor offers both a free and a paid version. The paid version comes with a higher maximum of API requests per day and a higher number of allowed concurrent requests.

Technical Details TextRazor offers software development kits for Python, PHP and Java to allow an ever easier implementation of their text analysis services. By using one of these SDK's, one only has to enter the API key distributed by TextRazor itself, add which functions of the API it wants to use, and send the text. If the programming language used is not supported by a TextRazor SDK, one can still make requests to their RESTful API.

Dandelion Dandelion is software as a service, describing itself as "Semantic Text Analysis as a service". The software, developed by the Italian startup Spaziodati, provides a set of text analysis tools like keyword extraction and sentiment analysis, with a primary focus on entity extraction. This set of tools can be accessed through their online RESTful API or implemented on premise.

Dandelion offers both a free and a paid version. The paid version comes with a higher API request limit per day and a higher amount of custom models allowed.

Dandelion presents itself as blazing fast and multilingual, enabling their text analysis services for English, French, German, Italian, Spanish and Portuguese.

Technical Details Dandelion offers a RESTful API through which their services are accessible. The requests can be done through a simple cURL request, only requiring an API key distributed by Dandelion itself and the text in either a plain, URL or HTML format.

The endpoint for the Entity Extraction service our implementation will be using is *"https://api.dandelion.eu/datatxt/nex/v1"*

4.2.2 Local NLP Tasks

For our prototype we will only implement one local NLP task: negation.

The concept of negation was previously described in section 3.2.4. The addition of negation recognition is huge in the context of text analysis, and therefore it must also be part of our solution.

At first we expected that the remote information extraction algorithms would be capable of detecting negations. However this turned out to be a false expectation. Therefore we decided to implement a negation detector ourselves.

Due to time and resource constraints it was too difficult to implement the newest and best negation algorithm. However, as stated by Mehrabi et al. (2015), a simplistic negation approach, using cue words without considering the semantics of a text, do perform well in many situations. One of these simple approaches is NegEx (Chapman et al., 2001b).

As described in section 3.2.4, NegEx looks for negation cue words in a sentence and negates any entity within six words. This project will implement the NegEx algorithm through the GeneralNegEx JAVA class developed by Imre Solti (*GenNegEx – A JAVA class to implement Wendy Chapman's NegEx algorithm.*, 2008).

To access the JAVA class we setup a Node.js server which allows access to the JAVA class. The server runs the Node.js Express framework, creating an API for us to access the negation functions from the tool.

This tool has the advantage that its quite lightweight, easy to implement while still effective for a large amount of negations. However, it is using a rule based approach (see section 3.2.4), as well as limited to the English language.

4.2.3 Knowledge Database

We previously described the Unified Medical Language System in section 3.3.2. The UMLS will be a major part of the solution as it will function as a knowledge database containing all medical terms, providing semantics to those words.

During evaluation of the solution, the UMLS will be used for comparison between the found entities and the annotations that come with the test data. This is necessary as some medical conditions or pharmaceutical drugs can be properly described by multiple terms. The UMLS can also be used to convert any found entities into other standardizations like SNOMED and ICD-10, the latter which is used for the STRIPA CDSS. However, this conversion is not included in our prototype as the main objective of the prototype is to evaluate the framework and not the applicability to the STRIPA framework.

Therefore the decision was made to use the UMLS search API to find a specific term. For each extracted term, the top 10 results returned by the API are selected and stored with the extracted term. These top results are considered to be equal to the extracted term and therefore should also result in a positive extraction when compared with the annotations. Apart from the concept unique identifier (CUI), we do not store any other information of the top results. The CUI can be send to the UMLS to get the complete information of a term.

4.2.4 Information Fusion

The information fusion challenge is that there are multiple data sources who all have different (types of) information and have it stored in different ways (I.E., Databases, Free text, HTML, etc). Meulendijk et al. (2015) stated that they found it difficult

to extract data because of all the incompatible systems, classifications standards and data formats. This in turn leads to loss of information and ultimately medical errors. (Meulendijk et al., 2017) We attempt to combine the information from multiple sources (multiple NLP services) by using the concept of information fusion.

As discussed in section 3.4, we will be using weighting systems and an ontology: UMLS. The prototype will be using a double weight system to fuse the information. The first weight system determines whether an extracted entity is similar to another extracted entity in that document. We do this by using the UMLS database to find equal terms to an extraction. The top 10 of these equal terms are saved in combination with the extraction. All 10 are then compared with 10 of another extraction. If the comparison results in a value higher than the set Similarity Threshold γ value, we consider it to be an equal entity. The comparison is done as follows:

- Calculate the percentage of equal terms over all 10 (α)
- Calculate the percentage of equal terms over the top 3 of the top 10. (β)
- Sum the multiplication of α by 0.35 and the multiplication of β by 0.65. This weight formula can be written down like: $\alpha \cdot 0.35 + \beta \cdot 0.65 > \gamma$

This will result in a value between 0 and 1, 0 being that the entities are not similar at all and 1 is the entities being exactly the same. We consider the β percentage to be of much more value than the α as when either of the first three are equal, the probability of the entity being exactly the same as another is much higher when the number eight, nine and ten are similar.

The second weight system determines whether an entity was extracted by a high enough weight to be actually considered "extracted". This weighting will determine a weight for each individual extractor by using the f-score calculated for just that single extractor applied to the document on which the extraction is running. This is done for all extractors, resulting in multiple f-scores, which can be normalized to an extractor-weight. For each entity extracted, we sum the weights of the extractors the entity was extracted by. If the weight is over the Extractor Threshold θ value, it is considered to be actually extracted. If its less it is considered to be a false extraction. This weight formula can be written down as: $\sum_{n=1}^{\#extractors} \omega_n$ where ω is the weight of the extractor.

4.3 Prototype

This section will depict more about the prototype. The main objective for developing the prototype was to evaluate the proposed framework as described in chapter 2.4. The evaluation of the prototype has been done with three different data sets, as can be seen in chapter 5

4.3.1 Requirements

At no time during this project, dedicated requirements were set. There were however some thoughts we had for the prototype which we saw necessary for it to become useful. In other words, we specified no "must haves", but some "should haves". These requirements were as follows:

- The prototype should run anywhere without any installation. Therefore the prototype should be developed as a web application.
- The prototype should be able to handle all test data in a "reasonable time", being the time a human normally would want to wait.
- The prototype should be easy to use. It should have a single place where a text can be input, a single place where extractors can be selected, and a simple overview of the results.
- As it is a prototype, the graphical user interface may be minimalistic.

Not all requirements turn out to be as important in every stage of the development. During the first version we got a lot of value out of a structured visual representation of the extracted data. This would make it much quicker for us to estimate what the capabilities and strengths of the extractors were. In at later stages, when only the test data had to be evaluated against the annotation, we decided that a minimalistic approach would suffice.

The decision was made to start developing a web application by using HTML5, CSS and JavaScript for the front-end, and PHP for the back-end. As the aim for the prototype is to only be a processor of various API requests, we expect these programming languages to suffice.

4.3.2 Early Versions

As described in chapter 2, we expected the development process to take several iterations before a decent prototype was developed. This indeed turned out to be the case.

The first version of the prototype was focused on getting an organized visual feedback on the information extraction and transformation process and to discover any challenges we might face when implementing this solution. Screenshots of it can be found in figure 4.2 and 4.3.

This first version only included two out of the six extractors, namely: IBM Watson and Thomson Reuters OpenCalais. Apart from using these two extractors, it was also capable of transforming the extracted entities to a standardized format. However, this process was very slow and turned out to be our first major obstacle in the architecture. It was also unable to detect negations.

The second version was an improved version of the first version. The goals were the same, getting a organized visual feedback on the information extraction and transformation process. The tool was brought back to one single view making it easier to evaluate the intermediate results. All other extractors were implemented, as was the negation handler. The negation handler results were not combined with the extractor results however, and thus not visible in the tool itself. A screenshot of the second version can be found in figure 4.4.

4.3.3 Final Version

With the practical information gathered from the early versions, it was time to create a version that was capable of handling large amounts of data. In contrast to what the early versions were able to handle, namely just the one text, we were

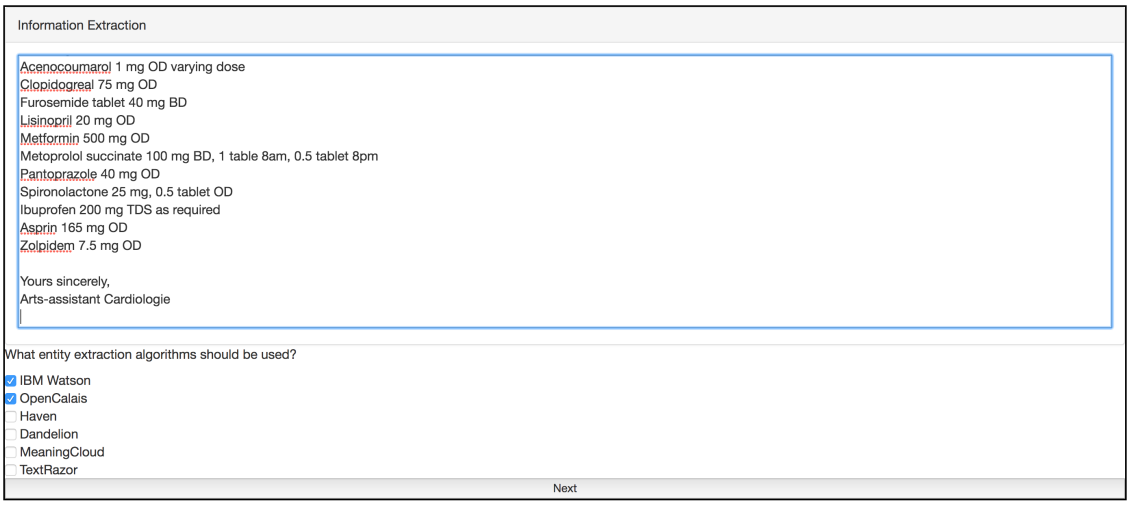

Figure 4.2: Screenshots of the first version of the prototype showing the simplicity of the input. Note that while all extractors are listed, only Watson and OpenCalais were implemented.

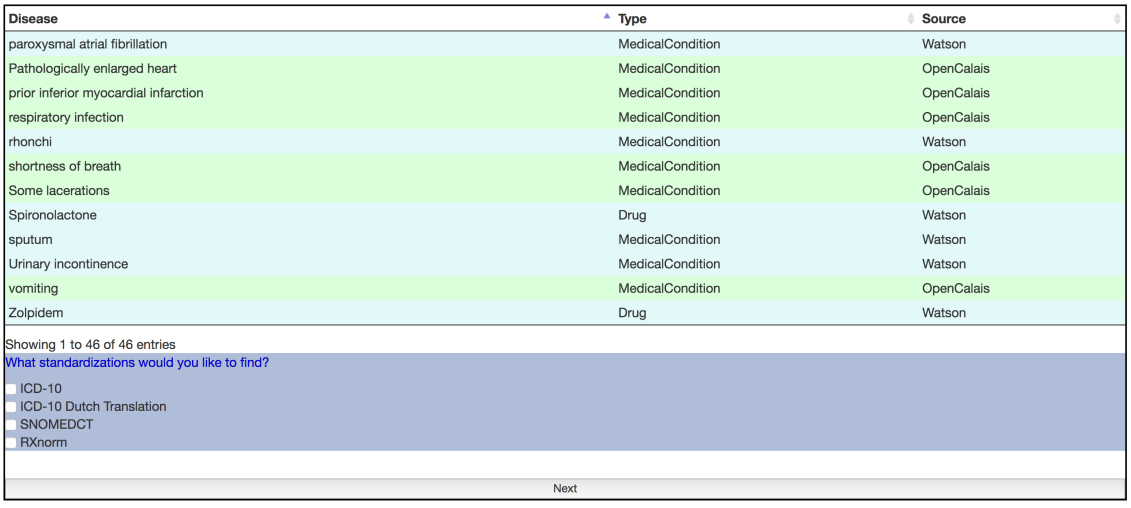

Figure 4.3: Screenshots of the first version of the prototype showing the extractor results from that text in a structured and sortable data table.

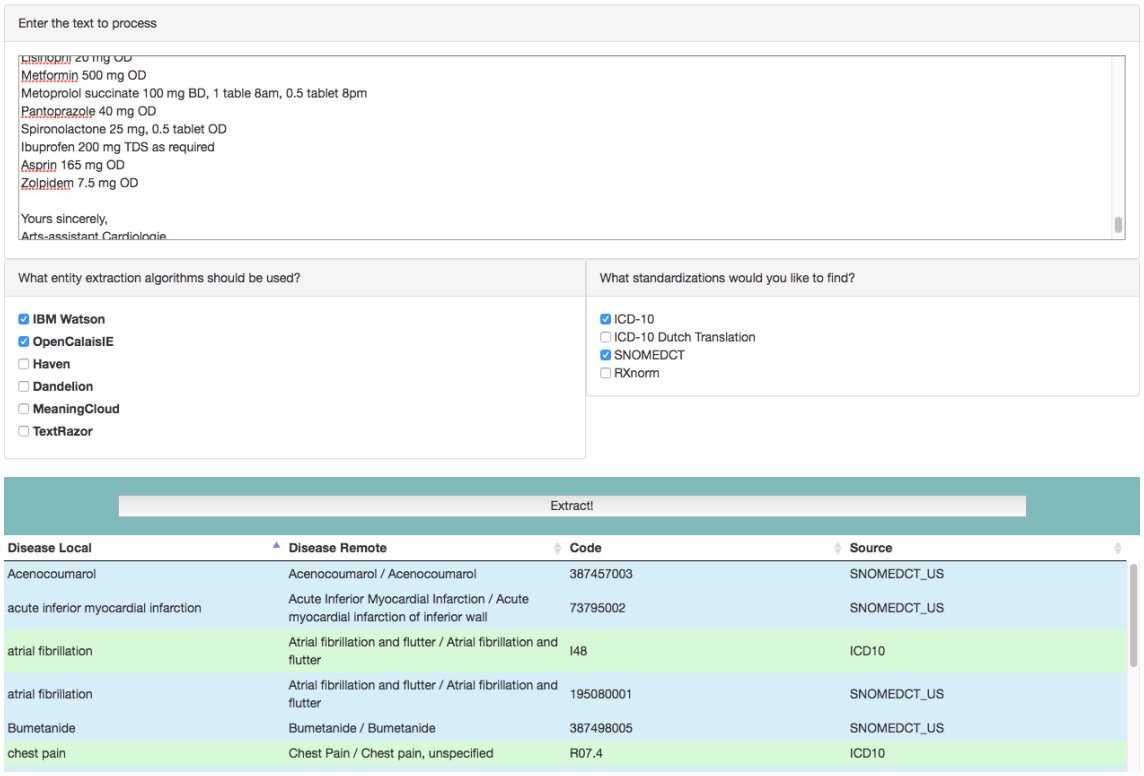

Figure 4.4: Screenshot of the second version of the prototype showing the single view containing the text input, options for extractors and the results.

already sure of over 100 documents in our test data. This, combined with the overall goal of having this architecture work for large amounts of clinical data, created the need for a tool that would be able to import and handle these large amounts of documents.

We therefore had to restructure the approach, going from a structure that took one text and provided us with one result, towards a structure that takes multiple documents and provides us with one result. The change was made from a text input to a document input connected to the document parser. The extractor handler then sends all documents, one by one, to all the extractors. Next, all entities returned by the extractor handler, including the text they were extracted from, are passed to the negation handler. The negation handler processes all entities as discussed in section 3.2.4 and finds any negated entities. All entities, negated or not, are then send to the UMLS handler. This element send all entities to the UMLS and returns the top results found per entity (as described in section 4.2.3). The entity, negation and similar UMLS concepts are then stored in the database. These results can be transformed into any standardization UMLS can offer us, after which it can be disseminated in any form necessary.

The following will just be done for evaluation purposes and would therefore not be necessary in a definitive architecture or process. The annotations are imported in the tool through another custom parser. All annotated terms are then send to the UMLS to get the UMLS concept for it. This UMLS concept is then compared to all UMLS concepts extracted by the remote extractors. The comparison results are then disseminated as a large custom JSON file. These results can be found in chapter 5.

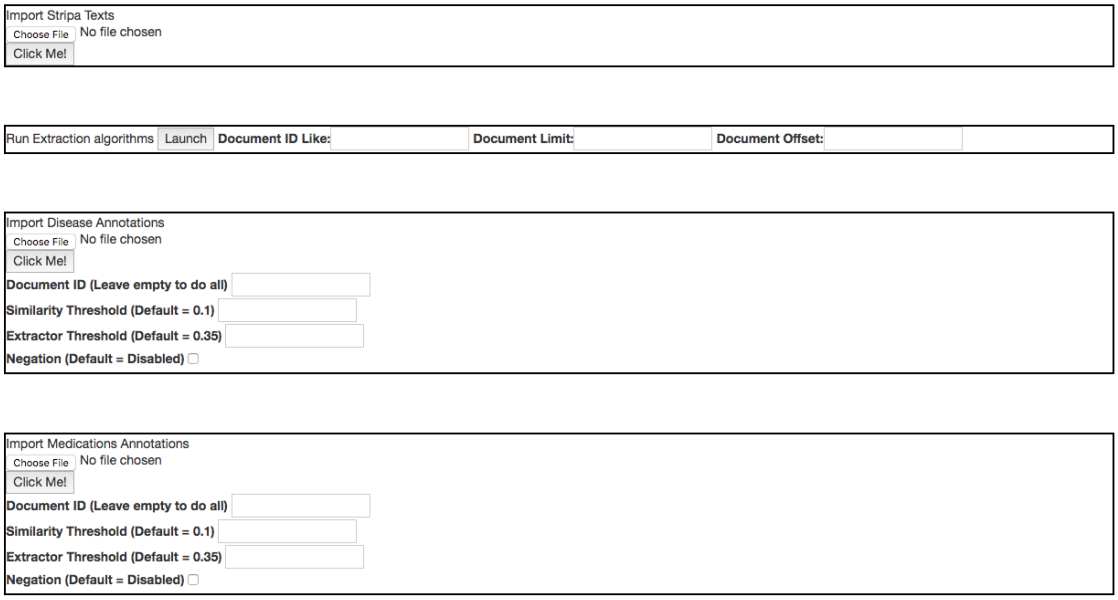

Figure 4.5: A screenshot of the input page of the third version of the prototype.

This version does not contain an interface. It consists of several buttons representing a specific function: importing a document, running the extractor, negation and UMLS handlers and comparing it with annotations. The results of the evaluation are presented in plain JSON. This minimalistic interface is not a problem, as the core aim is to evaluate the framework based on the results of the process and not the process itself. Images of this version can be found in figure 4.5 and figure 4.6.

This entire process can be seen in figure 4.7.

Interoperability As it is vital for the components to communicate the data, we need to set a single format to transfer this data. The decision was made to use the JSON format for this. JSON is capable of delivering large amounts of data in any format while still being easy to work with. Many existing systems, like all extractors, already implement an JSON export feature, and many programming languages and tools are capable to transform data into a JSON string. Most information extraction API's can communicate in both JSON and XML, with JSON being the default.

An exception on the usage of JSON might be the connection between data source and our tool, as the tool will be dependent on how the source disperses its data.

Within the tool the communication will be done through custom objects containing all the data. These object can easily be converted to a JSON string. Figure 4.8 adds the communication formats to the architecture of the tool.

Prototype Example The following example uses a snippet from one of the test data sets. It shows the entire process the text goes through in the prototype.

"Patient is a 79 year old Russian speaking female who had had a history of exertional chest pain and no previous documented coronary artery disease. In September 1995, patient presented to the Sumri Sondoyle Hospital with complaints of exertional chest pain with onset after one flight of stairs. Carotids were without bruits."

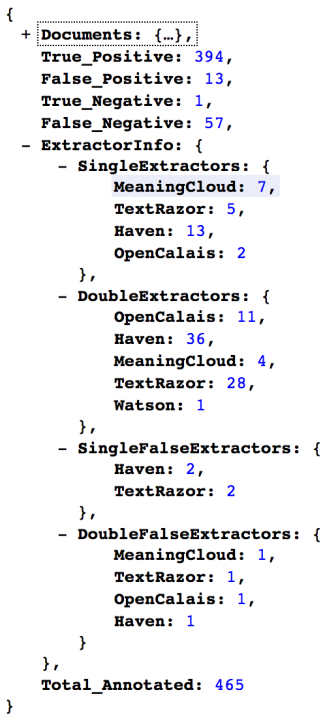

Figure 4.6: A screenshot of a simple JSON result page of the third version of the prototype.

The document is first imported into the system by using a custom parser. This custom parser removes all invisible line endings and creates new line endings after a dot. Next, the text is passed to all extractors, resulting in the following entity extractions:

- ”exertional chest pain” - OpenCalais
- ”chest pain” - Haven, TextRazor
- ”documented coronary artery disease” - OpenCalais
- ”coronary artery disease” - Haven, Dandelion
- ”complaint” - MeaningCloud
- ”bruit” - Haven
- ”bruits” - Dandelion, TextRazor

5 out of 6 extractors extracted one or more concepts. All entities are stored in the database. The text plus extractions are then send to the negation handler, which checks which entities are negated. It returns ”documented coronary artery disease”, ”coronary artery disease”, ”bruit” and ”bruits” as negated, as they are preceded by the que word ”no”. This is then saved with the corresponding entities in the database.

The next step is using the knowledge database UMLS to compare the terms. All concepts are looked up in the UMLS database, and the best 10 responses are saved in the database. The UMLS handler then tries to find similar terms, according to the calculations as described in 4.2.4. The three relevant similarities are between ”exertional chest pain” / ”chest pain”, ”documented coronary artery disease” / ”coronary artery disease” and ”bruit”/”bruits”. The UMLS CUI’s linked to the extractions are as follows:

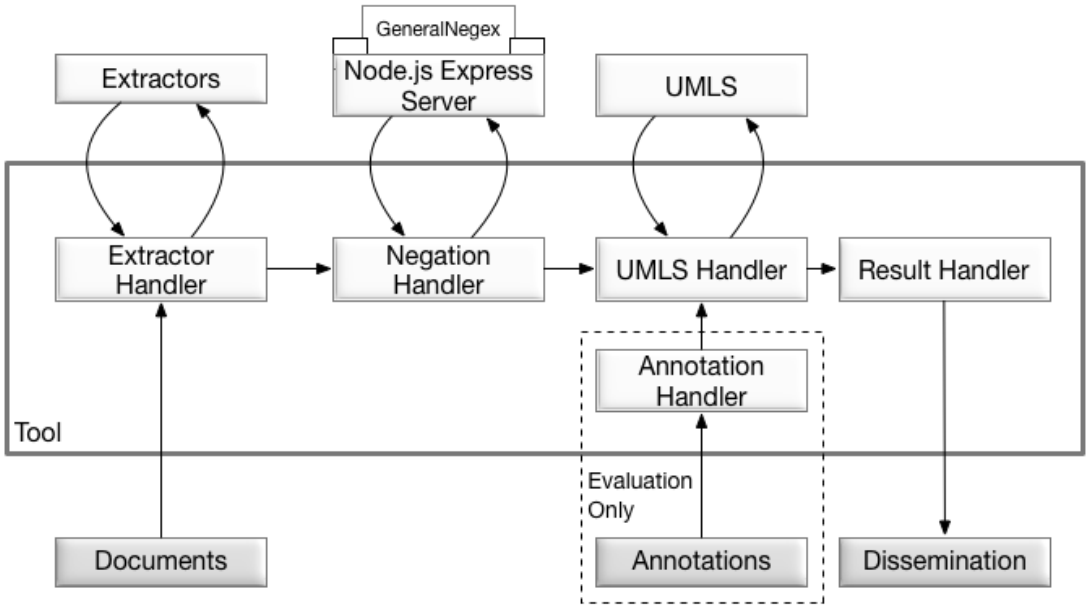

Figure 4.7: An overview of the elements and the process of MAPI-NLP.

- Exertional Chest Pain: C0232288, C0742319
- Chest Pain: C0008031, C2926613, C0008035, C0151826, C0742302,C0742304,C0742305, C0002962, C0002965, C0008033
- Documented Coronary Artery Disease: C2882164
- Coronary Artery Disease: C0010054, C0010068, C0343692, C0742826, C0742827, C0742828, C0742829, C0856171, C1299384, C1384898
- Bruit: C0006318, C0028263, C0007280, C0018820, C0028264, C0221755,C0232112, C0232257, C0238980, C0239614
- Bruits: C0006318, C0007280, C0018820, C0035234, C0221755, C0232112, C0233658, C0277934, C0426625, C0558799

Against what one might expect when looking at the names, none of the CUI's of the first four words match. Therefore, the similarity value will be 0 for each combination and thus no extraction will be considered to be equal, as it does not pass the default Entity Similarity Threshold of $\gamma = 0.1$. (see chapter 5 for more about the thresholds).

The last two entities, bruit and bruits, do have concepts in common. Therefore we will calculate whether the similarity is big enough for us to consider them to be equal, using the method described in section 4.2.4. Out of all 10 concepts, 5 are equal. This gives us an α value of 0.5. Out of the top 3 concepts (in the list above the first three CUI's), 2 are equal. This gives us a β value of 0.667. Considering that the top 3 are much more important, we get a final similarity value γ of

$$\gamma = \alpha \cdot 0.35 + \beta \cdot 0.65 = 0.5 \cdot 0.35 + 0.667 \cdot 0.65 = 0.608.$$

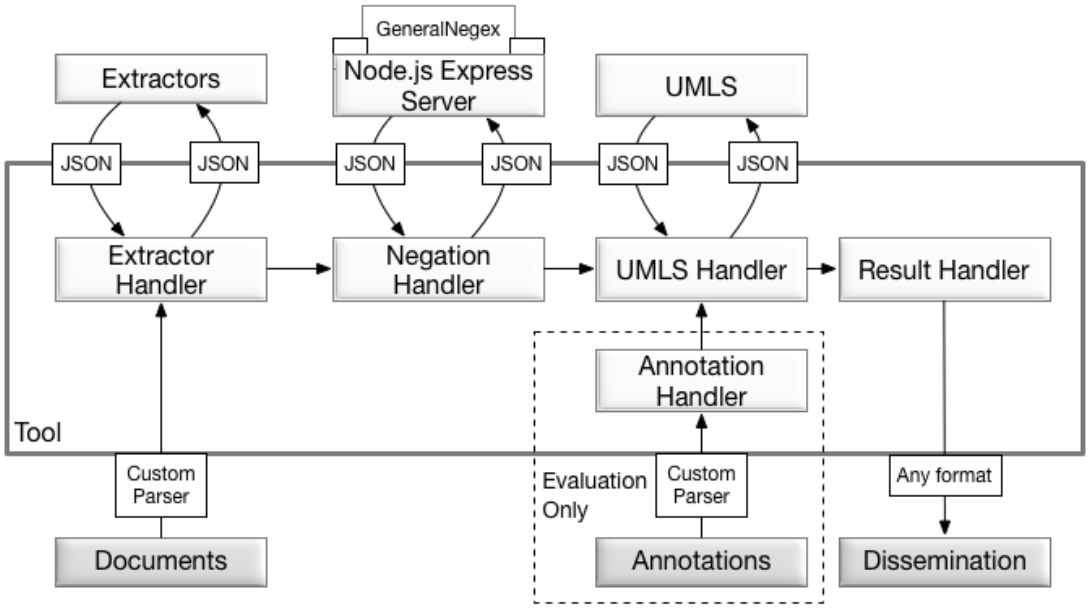

Figure 4.8: An overview of the syntactic interoperability between all the architectural elements of MAPI-NLP.

This value is larger than our default Entity Similarity Threshold of $\gamma = 0.1$, and therefore we consider "bruit" and "bruits" to be the same entity.

Finally, we will filter the entities we are not confident about. As we only consider a small snippet with a very low amount of entities, we are unable to calculate a accurate weight by using the f-score of each individual extractor. Therefore we will assign an equal value to each extractor: $1/N = 1/5 = 0.2$ (where N = number of Extractors). By default, we accept an extraction if our confidence value is over the Extractor Threshold $\theta = 0.35$. (see chapter 5 for more about the thresholds).

- Exertional Chest Pain: $\sum_{i=0}^{N} 0.2 = 0.2$ where N = Number of Extractors.
- Chest Pain: $\sum_{i=0}^{N} 0.2 = 0.4$ where N = Number of Extractors.
- Documented Coronary Artery Disease: $\sum_{i=0}^{N} 0.2 = 0.2$ where N = Number of Extractors.
- Coronary Artery Disease: $\sum_{i=0}^{N} 0.2 = 0.4$ where N = Number of Extractors.

Only chest pain and coronary artery disease pass the threshold of 0.35. We disregard exertional chest pain and documented coronary artery disease, considering them to be "not extracted".

Finally, the result handler compares the fused and non-disregarded entities with the ground truth of the data set. Using Binary Classification, we can ultimately calculate recall, precision, accuracy and the F-score of our extraction.

5. Results

This chapter will depict the results gathered with the implementation artifact. These results will be used to evaluate the proposed architecture, to see if the results are as good as alternative NLP systems.

For the evaluation we used three different, de-identified, data sets. Two of the data sets are taken from the Informatics for Integrating Biology and the Bedside (i2b2) center (*Informatics for Integrating Biology and the Bedside (i2b2)*, 2017). The third was a small subset of test data from the STRIPA system.

- **2008 Obesity Challenge - i2b2:** This dataset exists out of 611 discharge letters. All discharge letters are annotated on 16 different medical condition terms in the context of obesity. The following terms were annotated:

  - Asthma
  - Coronary Artery Disease
  - Congestive Heart Failure
  - Depression
  - Diabetes
  - Gallstones
  - Gastroesophageal reflux disease (GERD)
  - Gout
  - Hypercholesterolemia
  - Hypertension
  - Hypertriglyceridemia
  - Osteoarthritis
  - Obesity
  - Obstructive Sleep Apnea
  - Peripheral Vascular Disease
  - Venous Insufficiency

  Terms could be either annotated as being in the document, not being in the document, or undecided/unknown which was treated as a not being in the document. The strength of this data set, concerning the aim of these tests, is that there are a lot of documents, its weakness that it is only annotated for 16 rather abstract terms in the context of obesity.

- **2009 Medication Challenge - i2b2:** This dataset existed out of 10 discharge letters. All discharge letters were annotated by i2b2 on any occurrence of a pharmaceutical drug. Terms could either be annotated or not. The strength of this data set is that it is annotated with all occurrences of pharmaceutical drugs. Its weakness is that the data set is there are a limited number of documents.

- **STRIPA subset:** The STRIPA dataset consists out of five admission letters from different hospitals throughout the Netherlands. They were all annotated by medical experts from the University Medical Center (UMC) Utrecht. The annotations contain both medical conditions and pharmaceutical drugs. The annotations also contain transformations to the ICD-10 standardization

domain. The results will include the measurements of recall, precision and f-measure for this transformation. The strength of this data set is that it is very specifically annotated. Its weakness is that the data set is there are a limited number of documents.

All test data was in the English language. The tool is able to process various languages, albeit with varying results as not all extractors are able to handle all languages. Unfortunately we were unable to attain test data in any other language.

The tools is able to vary the NLP process based on three different variables:

- **Negation (κ):** Whether the results of negation processing are taken into account. If this boolean value is false, all extracted entities are considered not to be negated and thus the local NLP task of negation will be completely ignored.
- **Entity Similarity Threshold (γ):** As part of the information fusion process, entities will be compared based on their relation in the UMLS database. The result of this comparison is a strength of the relation between the two entities. If this strength passes the threshold, the entities are considered to be equal. A threshold value of 0 results in matching entities even when only one of the concepts are equal. A threshold value of 1 only results in matching entities when all concepts of both entities are equal.
- **Extractor Threshold (θ):** As part of the information fusion process, all extractors get assigned a weight based on their individual F-Score for a specific text. During the final round, only entities extracted by a combined weight that surpasses the 'extractor threshold' will be considered to be truly extracted. If the combined weight does not reach the extractor threshold, we are not confident the extraction was meaningful. A threshold value of 0 results in a confident extraction even if only one extractor was able to extract it. A threshold value of 1 means that all extractor need to have extracted a concept before it is accepted.

The above three variables greatly influence the results, both in a positive and a negative way. The aim is to strive for the best combination of these variables for each specific data set. We will start with a set of default values we feel are accurate, from which we will perform a manual grid search to find the best combination of parameters per data set. After some tweaking around with the numbers we concluded that the values of κ = false, $\gamma = 0.1$ and $\theta = 0.35$ were a decent starting point for further exploration. By increasing the γ value we need entities to be more similar before they are seen as one entity, resulting in a lower false positive and higher false negative number. By increasing the θ value we need entities to be extracted by more extractors, thus decreasing the number of false positives and increasing the number false negatives. However, increasing either value will also result in a lower amount of true positives.

5.1 Results - Overall

Apart from the results of the three data sets, it is interesting to look how the six extractors compared to each other. In total 115 documents were analyzed for entities

	Watson	OpenCalais	MeaningCloud	Dandelion	TextRazor	Haven	Total
Medical Conditions	1271	2673	1523	2486	3886	3171	15010
Pharmaceutical Drugs	988	0	0	3012	405	1737	6142
Total	2259	2673	1523	5498	4291	4908	21152

Figure 5.1: The extraction results per extractor

of type "Medical Condition" or "Pharmaceutical Drug". All six extractors and both types combined this resulted in a total of 21152 entities. Note this number contains

- Entities that are not part of the annotations, but still be considered as relevant by the extractors.
- Exactly the same entities in the same document, but from different extractors.

The individual additions of the extractors to this total can be seen in table 5.1. This table already shows the effect of the data fusion theory described in 3.4. Some extractors were capable of extracting pharmaceutical drugs, while others were unable to or returned a very low amount of results. This does not, however, state which extractor is best and which is worst. Extractors may very well extract a lot of irrelevant or incorrect data.

5.2 Results - Obesity Challenge

The de-identified clinical records used in this dataset were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY. The obesity challenge dataset is the largest data set we use for evaluation, with over 600 discharge letters. Due to limitations of the free API's, we were only able to apply our prototype to a random 100 out of the 611 documents. The annotations that come with the documents are, however, limited to a static set of 16 diseases.

Calculating the recall, precision and f-measure as described in 2.4.1 by using the default parameter values ($\gamma = 0.1$ and $\theta = 0.35$) we get the results as in table 5.1. The table presents us with extremely high false positive and true negative number. Straight away one can see the difference between F-Score and Accuracy. Even though there is an immense amount of false positives, which should be as close to 0 as possible, the accuracy is still well over 60%. Therefore, as discussed in section 2.4.1, F-Score is to be considered a superior evaluation metric.

While the true negative isn't a problem, the false positive count makes the F-score plummet to only 0.258. These results can be explained by the simple fact that the annotations are only about 16 obesity related terms. This obesity-filter is not and cannot be applied on the prototype side. However, we can tweak the tool to only compare extractions with annotations and only count those who are linked to one of the annotations. The results of this change can be seen in table 5.2. As can be seen, the FP and TN are much lower now, giving a much better representation of the actual F-score.

The amount of false negatives are relatively high compared to the amount of false positives. We decrease the value of γ and we lower the value of θ to filter the

Table 5.1: Results of the obesity challenge data set when the default parameter values are applied

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.1	0.35	351	1921	3229	96	0.785	0.154	0.258	63.9%
true	0.1	0.35	328	1871	3279	131	0.715	0.149	0.247	64.3%

Table 5.2: Results of the obesity challenge data set when the default parameter values are applied

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.1	0.35	351	22	1123	96	0.785	0.941	0.856	92.6%
true	0.1	0.35	329	17	1128	126	0.723	0.951	0.82	90.8%

false positives. We also see if keeping γ the same value and lowering θ to see if that gives better numbers. The results can be seen in table 5.3

When looking at the results it becomes immediately clear that when using negations the overall results are worse then when not using negation. Using negation also carries the risk that entities which are not negated are considered to be negated, effecting in a lower amount of true positives and a higher amount of false positives.

Secondly, we tried finding the best balance between γ and θ to find the best FP and FN numbers in relation to TP. As table 5.3 shows, the best results came when using a lower θ and a higher γ than the default. This is because when lowering the γ value, exceptionally more false positives are found which can not be filtered by raising the extraction threshold.

Overall, the parameters values that result in the highest accuracy:

- κ = false
- γ = 0.10 / 0.20
- θ = 0.20
- F-Score = 0.861
- Accuracy = 92.7%

5.3 Results - Medication Challenge Data Set

The deidentified clinical records used as this dataset were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY. The corpus consists out of 10 discharge letters, annotated on pharmaceutical drugs. We start the evaluation by using our default values for γ and θ again. The results of these analyses can be seen in table 5.4.

The first thing one might notice is that the numbers for using negation and not using negation are not equal. This is because a term was tagged as both negated and not negated in one single document. This results in it being counted as both in the text and not in the text, resulting in one higher false negative when using

Table 5.3: Results of the obesity data set when the parameter values are slightly altered

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.05	0.35	351	22	1123	96	0.785	0.941	0.856	92.6%
false	0.00	0.35	370	58	1087	77	0.828	0.864	0.846	91.5%
false	0.00	0.40	365	51	1094	82	0.817	0.877	0.846	91.6%
false	0.00	0.45	360	45	1100	87	0.805	0.888	0.845	91.7%
false	0.00	0.50	356	39	1106	91	0.796	0.901	0.846	91.8%
false	0.00	0.30	372	63	1082	75	0.832	0.855	0.843	91.3%
false	0.10	0.30	352	26	1119	95	0.832	0.855	0.843	92.4%
false	0.10	0.25	356	27	1118	91	0.796	0.930	0.858	92.6%
false	0.10	0.20	360	29	1116	87	0.805	0.925	0.861	92.7%
false	0.20	0.20	360	29	1116	87	0.805	0.925	0.861	92.7%
true	0.00	0.35	343	50	1097	117	0.746	0.873	0.804	89.6%
true	0.05	0.35	330	17	1128	126	0.724	0.951	0.822	91.1%
true	0.05	0.40	325	15	1130	130	0.714	0.956	0.818	90.9%
true	0.10	0.40	324	15	1130	130	0.714	0.956	0.818	90.9%
true	0.15	0.40	324	15	1130	130	0.714	0.956	0.818	90.9%
true	0.15	0.40	324	15	1130	130	0.714	0.956	0.818	90.9%
true	0.15	0.45	317	14	1131	137	0.698	0.958	0.808	90.6%
true	0.20	0.40	324	15	1130	130	0.714	0.956	0.818	90.9%

Table 5.4: Results of the medication challenge data set when the default parameter values are applied

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.1	0.35	140	42	48	80	0.636	0.838	0.724	59.9%
true	0.1	0.35	136	42	48	84	0.618	0.764	0.683	59.5%

Table 5.5: Results of the medication challenge data set when the parameter values are slightly altered

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.00	0.25	156	59	17	63	0.712	0.723	0.719	58.6%
false	0.00	0.30	143	38	38	76	0.653	0.790	0.715	61.4%
false	0.00	0.35	139	37	39	80	0.635	0.790	0.704	60.4%
false	0.00	0.40	130	35	41	89	0.594	0.788	0.677	58.0%
false	0.05	0.25	153	66	19	66	0.699	0.699	0.699	56.6%
false	0.05	0.30	144	43	42	75	0.658	0.770	0.709	61.2%
false	0.05	0.35	140	42	43	79	0.639	0.769	0.698	60.2%
false	0.05	0.40	127	39	46	92	0.580	0.765	0.660	56.9%
false	0.05	0.45	101	37	48	118	0.461	0.732	0.566	49.0%
false	0.10	0.40	121	39	51	100	0.582	0.848	0.690	55.3%
false	0.10	0.45	101	39	51	119	0.468	0.851	0.604	49.0%
true	0.00	0.25	152	59	17	67	0.694	0.720	0.707	57.3%
true	0.00	0.30	144	38	38	76	0.655	0.791	0.716	61.5%
true	0.00	0.35	140	37	39	80	0.636	0.791	0.705	60.5%
true	0.00	0.40	131	35	41	89	0.595	0.789	0.679	58.1%
true	0.05	0.25	149	66	19	70	0.681	0.694	0.687	55.3%
true	0.05	0.30	140	43	42	79	0.639	0.765	0.697	59.9%
true	0.05	0.35	136	42	43	83	0.621	0.764	0.685	58.8%
true	0.05	0.40	123	39	46	96	0.562	0.759	0.646	55.6%
true	0.05	0.45	99	37	48	120	0.452	0.728	0.558	48.4%
true	0.10	0.40	120	39	51	100	0.545	0.755	0.633	55.2%
true	0.10	0.45	101	39	59	119	0.459	0.721	0.561	49.0%

negation. The results yield a high amount of false negatives, which basically means that a lot of annotated terms are not found by the tool. This can be explained by the lack of two extractors (OpenCalais and MeaningCloud are unable to extract Pharmaceutical Drug Entities), resulting in a lower amount of extracted entities and a different scale of extractor weights. Another reasons might be that pharmaceutical drugs are less easily compared than medical conditions are. To test this, we will lower the γ and θ parameters. The results of these modifications can be seen in table 5.5

Looking at the results, we see there is a slight difference between using negation and not using a negation, in favor of not using a negation. However, the average difference is so small that we could almost neglect this. It is more interesting to see that the overall results are a lot lower compared to the results of the obesity tests. Even when pushing γ to 0.00, which results in two entities being the same if any one of the 10 concepts is similar, the recall barely rises. Raising the θ value also barely increases the precision, but does decrease the recall.

Based on these numbers we can conclude that extraction of information from this data set is not adequate enough, resulting in a weak information fusion process. The best results we were able to gather were with the following parameters:

- κ = false
- γ = 0.10

Table 5.6: Results of the STRIPa data set when the default parameter values are applied, using the medical condition annotations

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.1	0.35	20	79	63	12	0.625	0.202	0.305	47.7%
true	0.1	0.35	20	79	63	12	0.625	0.202	0.305	47.7%

Table 5.7: Results of the STRIPA data set when the parameter values are slightly increased to decrease the amount of false positives

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.10	0.50	19	51	91	13	0.59375	0.271	0.373	63.2%
false	0.30	0.35	18	75	71	14	0.563	0.194	0.288	50.0%
false	0.30	0.50	17	53	93	15	0.531	0.243	0.333	61.8%
false	0.30	0.60	16	39	107	16	0.500	0.291	0.368	69.1%
false	0.40	0.60	10	41	113	22	0.313	0.196	0.241	69.1%
false	0.40	0.70	9	22	132	23	0.281	0.290	0.286	75.8%
true	0.10	0.50	19	51	91	13	0.59375	0.271	0.373	63.2%
true	0.30	0.35	18	75	71	14	0.563	0.194	0.288	50.0%
true	0.30	0.50	17	53	93	15	0.531	0.243	0.333	61.8%
true	0.30	0.60	16	39	107	16	0.500	0.291	0.368	69.1%
true	0.40	0.60	10	41	113	22	0.313	0.196	0.241	69.1%
true	0.40	0.70	9	22	132	23	0.281	0.290	0.286	75.8%

- $\theta = 0.35$
- F-Score = 0.724
- Accuracy = 59.5%

5.4 Results - STRIPA

The third, smallest but most specific data set we will use for evaluation is the STRIPA data set. The data set consists out of five admission letters, annotated with very specific medical conditions and medication / pharmaceutical drug usage.

Medical Conditions test We start this evaluation by using the default values of $\gamma = 0.1$ and $\theta = 0.35$ on just the annotations of medical conditions. These results can be seen in table 5.6. An interesting thing to see is that negations in this data set do not have any influence at all. As it turns out, no matched entities were negated, resulting in equal numbers. Also, the amount of false positives in this set is extremely high compared to the true positives, true negatives and false negatives. Therefore we will adjust the values of both γ and θ upwards. The results of these adjustments can be seen in table 5.7

Again, the results show that there is no difference between using and not using negations. Increasing γ and θ did not have the expected effect of lowering the amount of false positives and increasing the F-Score. Instead the amount of TP dropped almost equally as fast, resulting in an even lower F-Score. After analysis

Table 5.8: Results of the STRIPA data set when the default parameter values are applied, using the medication annotations.

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.1	0.35	29	10	13	10	0.744	0.744	0.744	67.7%
true	0.1	0.35	29	10	13	10	0.744	0.744	0.744	67.7%

of why the amount of false positives was so high, we had to conclude that the annotations were somewhat lackluster. For example, the sentence *"Fall during the night, multiple hematomas. Orthostatic hypotension proven."* contains two medical conditions: hematoma and orthostatic hypotension. Hematoma was found by two out of six extractors (and filtered once θ was raised to 0.50), orthostatic hypotension was found by five out of six. However, neither of these two was annotated, most likely because the context of the sentence was in past tense and potentially not applicable to the current state of the patient. During the test with the Obesity Challenge data set, we argued that we could include just the extractions that match one of the annotations, as the data set only focused on diabetes related terms. However, we can not do that with this data set as the documents cover all medications and thus we must also use all extractions.

The best results were gathered with the following parameters:

- κ = true / false
- $\gamma = 0.10$
- $\theta = 0.50$
- F-Score = 0.373
- Accuracy = 63.2%

Pharmaceutical Drugs test We start this evaluation by using the default values of $\gamma = 0.1$ and $\theta = 0.35$ on just the annotations of pharmaceutical drugs. These results can be seen in table 5.8. Again, negation handling in this data set does not have any influence at all. As it turns out, no matched entities were negated, resulting in equal numbers between both tests. As with the medical condition test, both the amount false positives and the amount false negatives are neither exceptionally high nor low. Therefore we will adjust the values of both γ and θ both upwards and downwards. The results of these adjustments can be seen in table 5.9.

Again, the results show that there is no difference between using and not using negations. Lowering γ to 0.00 doesn't even nearly remove all false negatives. This means that even when the least strict entity similarity measurement is used, still eight annotated entities are not found by the system. This result can be caused by either the extractors not being able to correctly extract entities, as we only have four extractors for medications, or the annotations being too specific. As the latter can not really be true, we must conclude that the extractors can not correctly extract these eight entities.

- κ = true / false

Table 5.9: Results of the STRIPA data set when the parameter values are slightly increased to decrease the amount of false positives. This test uses the medication annotations.

κ	γ	θ	TP	FP	TN	FN	Recall	Precision	F-Score	Acc
false	0.00	0.35	31	7	11	8	0.795	0.816	0.805	73.7%
false	0.00	0.50	31	7	11	8	0.795	0.816	0.805	73.7%
false	0.10	0.25	29	17	6	10	0.744	0.707	0.725	56.5%
false	0.10	0.30	29	15	8	10	0.744	0.659	0.699	59.7%
false	0.10	0.40	29	10	13	10	0.744	0.744	0.744	67.7%
false	0.10	0.45	28	10	13	11	0.718	0.737	0.727	66.1%
false	0.10	0.50	28	10	13	11	0.718	0.737	0.727	66.1%
false	0.15	0.40	28	11	13	11	0.718	0.718	0.718	65.1%
true	0.00	0.35	31	7	11	8	0.795	0.816	0.805	73.7%
true	0.00	0.50	31	7	11	8	0.795	0.816	0.805	73.7%
true	0.10	0.25	29	17	6	10	0.744	0.707	0.725	56.9%
true	0.10	0.30	29	15	8	10	0.744	0.659	0.699	59.7%
true	0.10	0.40	29	10	13	10	0.744	0.744	0.744	67.7%
true	0.10	0.45	28	10	13	11	0.718	0.737	0.727	66.1%
true	0.10	0.50	28	10	13	11	0.718	0.737	0.727	66.1%
true	0.15	0.40	28	11	13	11	0.718	0.718	0.718	65.1%

- $\gamma = 0.00$
- $\theta = 0.35$ / 0.50
- F-Score = 0.805
- Accuracy = 73.7%

6. Conclusion

This chapter will conclude the research. We will discuss the framework, prototype and provide answers on the research questions, stated in section 1.4. The research evaluation, approach and results will be put up for discussion on what should have gone different or better. Finally, we will discuss what work might be necessary to improve this research.

6.1 Research Questions

Now that we finished evaluation of our framework with the test data sets, it is time to conclude whether the framework has any chance of success. We start this off by answering our sub research questions followed by answering the main research question.

SRQ 1: How can we reuse existing API-based NLP solutions? As discussed in chapter 3, multi-API based NLP is a natural language processing approach by using multiple external natural language processing system, accessed by and referred to as an API. We use these remote API-based NLP solutions to create a new approach for NLP. In contrast to the somewhat older approach of building an NLP framework which uses only one type of NLP task and one type of knowledge database, we outsource our tasks and knowledge, combining them internally and create a easier, cheaper, and more efficient NLP solution.

SRQ 2: What are the characteristics of the multi-API based NLP framework? In section 3.5 we explained the positive characteristics of Multi-API based NLP, relative to single or local alternatives.

The aim of a multi-API based NLP is to perform NLP processes externally to save money and time, making the NLP processes more easy to implement and the overall NLP task more efficient. In contrast to a local NLP implementation, the MAB approach makes use of multiple NLP systems allowing for more functionality and different NLP views on the to be processed text. This concretely results in different dictionaries per NLP API when processing entities, multiple supported languages and better processing times.

SRQ 3: How to implement the framework In chapter 4.3, we discussed our approach for the implementation of the framework. This implementation functioned as both prototype and evaluation for the framework itself.

We constructed an implementation which does not mimic the framework 100%, but instead constructed a tool that was capable of testing whether the concept of the MAPI-NLP framework would perform well.

SRQ 4: Does the concept of information fusion work for entity extraction? We've described our implementation of information fusion in section

Dataset	Highest F-Score	Highest Accuracy
Obesity Challenge Dataset	0.861	92.7%
Medication Challenge Dataset	0.724	62.7%
STRIPA Medical Condition Dataset	0.373	75.8%
STRIPA Medication Dataset	0.805	73.7%

Table 6.1: The accuracy of the tested data sets.

4.2.4. We found this approach to be satisfactory for the implementation of our prototype. This can be seen in the results (chapter 5).

SRQ 5: How does the implementation perform with clinical test data? We created an implementation that was capable of processing numerous clinical documents and perform NLP tasks on them. Over 100 documents were processed and the results are presented in chapter 5. When we analyze the results, we can conclude that the implementation performed satisfactory, but did not result in groundbreaking numbers. The best results for each data set can be seen in table 6.1.

As can be seen with these datasets, already, is that the performance varies largely per dataset. This is caused by either the implementation or by the dataset itself. As these numbers vary largely per dataset, we are unable to compare these numbers to other well accepted NLP approaches.

To conclude, we answer our main research question:

> **MRQ: How can unstructured clinical data be evolved to structured and standardized information utilizing existing NLP API's?**

As an answer to this question we propose a framework that implements a multi-API based NLP approach. It contains elements of multi-API NLP processing, data fusion, evaluation and dissemination to cover all the needs in a NLP system. We explored the benefits of using an API based approach over a local distribution, constructed a prototype to evaluate the framework with and performed actual tests to see if the proposed framework has any chance of success.

We may conclude that evolving unstructured data to structured information by using a multi-API based NLP approach does have potential, and can grow into a full fletched NLP system when expanded with new remote NLP system and additional post processing. We did prove the utility of the framework by developing the prototype and testing it with authentic clinical data.

6.2 Discussion and limitations

With this project we showed that it is possible to create a natural language processing system based on external natural language processing providers. A lot of businesses try to master computational linguistics and put years of work with a lot of people into it to achieve as high a result as possible. However, in a matter of weeks with very limited resources, we were able to perform entity extraction with an adequate accuracy and F-Score on clinical documents.

At the start of this project we had very high hopes of all the extractors and their functions. We expected them to not only contain functions like term or entity extraction, but also functions like relation extraction or the more specific negation extraction. This did not turn out to be the case, setting the completion of the tool quite a step back as we had to implement our own negation handler. Arguments go that this makes the extractor part of not enough value to have implemented remotely, as next to the entity extraction a system would still need a number of post processing NLP techniques to improve the results. We agree that it can be viewed like that, however it does also contain the necessary benefits (as discussed in section 3.5. Apart from not having to implement the extraction part yourself, which can be a complex part by itself, it also adds the capability of multilingualism, scalability and information fusion.

The information fusion of the various extractors was quite a challenge. Again for this we needed an external knowledge base, UMLS, for this to become possible. Simple matching on strings did function adequate enough and therefore knowledge behind an entity has to be gathered before matching can be done. This knowledge is very context specific, rendering the instantiation of this framework unable to function for other domains than the clinical domain. One would need another knowledge base for this framework to function with another context, which makes the framework less flexible as a whole. Our initial hopes were that the framework would be context independent.

We were however able to produce an adequate function for information fusion. We feel that the function as described in section 3.4 did a good job at combining similar entities, both for extractions and annotations. We did see that shifting the parameters of the fusion did have a major impact on the results, making the difference between a decent and good accuracy/F-Score.

One might argue that the data sets used for the evaluation are not adequate enough. The data sets used did deliver some variance and challenge for the evaluation. It did not, however, include a data set that contained challenges like volume, variety, completeness or being very specific. This made the evaluation look at only specific sides of the spectrum, whereas for the research we were aiming to provide a view on the full spectrum. This limitation was ultimately visible in the lack of data sets in other languages than English, practically making us unable to evaluate an important part of the framework: multilingualism.

As suggested during the research and development of the framework, this project did indeed show that using concepts like outsourcing and information fusion in NLP dramatically lowers development time and thus lowers development costs, while maintaining at least an equal level of accuracy. Other advantages of our approach include scalability, programming language independence and multi language support. Especially multi language support is something of extremely large value in contrast to other NLP approaches. Finally, expanding the implementation of this framework to increase accuracy is done in a breeze, as it is only necessary to include more remote NLP services.

6.3 Future Work

While this project did deliver the fundamentals for an NLP approach that is among other benefits easier and cheaper to implement, it can do much better when im-

provements are made.

One major improvement that will deliver much better results is the addition of many more extractors. Due to time and resource constraints this project only implement six extractors, which turned out to be barely enough. With more extractors come more functions, more languages and a larger dictionary of entities, improving the overall rate of extraction. More extractors also makes the weighting in the information fusion process more equal while at the same time more important. Functions like recognizing temporality, relations and negations could greatly increase the results and make the entire framework even easier to implement, as the amount of local NLP processes can be reduced.

An improved information fusion process might improve the results as well. This project delivered a relatively simple and slow weighting system, which requires an even slower UMLS on the background. Additional knowledge bases besides UMLS could be used, allowing for multiple views on the knowledge of clinical entities. This will improve the entity similarity calculations, as more knowledge about entities could be gathered. A start for this would be to accept more than the first 10 concepts from UMLS and improve the weighting system based on that. A more complex information fusion process might improve the results greatly, as could already be seen when adjusting the parameters of the current information fusion implementation.

The current framework could already benefit from more varying data sets, including sets in different languages and ones with annotations of every entity and not only those in a specific context. Using more varying data sets will show the promise of the framework being multi-language as well as showing the true potential in accuracy. It could be a study on itself to see what the effect is when both the amount of remote NLP services is increased and the amount of test data.

During discussion of the framework, data ethics was discussed and argued to be of importance. As this framework handles data and shares it with external sources, people might worry that their personal medical information ends up in commercial, wrong hands. By putting more research in the field of data ethics, a function to anonymize and de-identify texts should be implemented in the framework before sending a text to an external source. The anonymization and de-identification is already found to be of major importance, and several approaches have already been developed to tackle this issue. (Kleinberg, Mozes, van der Toolen, et al., 2017; Kushida et al., 2012; Menger, Scheepers, van Wijk, & Spruit, 2017)

References

Abbott, D. (2013). *Introduction to text mining.* Retrieved from `http://www.vscse.org/summerschool/2013/Abbott.pdf`

Aggarwal, C. C., & Zhai, C. (2012). *Mining text data*. Springer Science & Business Media.

Ambinder, E. P. (2005). Electronic health records. *Journal of oncology practice*, *1*(2), 57–63.

Apache, U. (2011). Apache software foundation. *URL http://java. apache. org*.

Aronis, J. M., Cooper, G. F., Kayaalp, M., & Buchanan, B. G. (1999). Identifying patient subgroups with simple bayes'. In *Proceedings of the amia symposium* (p. 658).

Barbantan, I., & Potolea, R. (2014). Exploiting word meaning for negation identification in electronic health records. In *Automation, quality and testing, robotics, 2014 ieee international conference on* (pp. 1–7).

Bates, M., & Weischedel, R. M. (2006). *Challenges in natural language processing*. Cambridge University Press.

Benson, T. (2012a). *Principles of health interoperability hl7 and snomed*. Springer Science & Business Media.

Benson, T. (2012b). Standards development organizations. In *Principles of health interoperability hl7 and snomed* (pp. 83–98). Springer.

Bernstam, E. V., Smith, J. W., & Johnson, T. R. (2010). What is biomedical informatics? *Journal of biomedical informatics*, *43*(1), 104–110.

Björkman, I. K., Fastbom, J., Schmidt, I. K., Bernsten, C. B., Caramona, M., Crealey, G., ... others (2002). Drug—drug interactions in the elderly. *Annals of Pharmacotherapy*, *36*(11), 1675–1681.

Bodenreider, O. (2007). The unified medical language system what is it and how to use it? *Tutorial at Medinfo*.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the workshop on speech and natural language* (pp. 112–116).

Cardie, C. (1997). Empirical methods in information extraction. *AI magazine*, *18*(4), 65.

Chang, Y. B., & Gurbaxani, V. (2012). Information technology outsourcing, knowledge transfer, and firm productivity: An empirical analysis. *MIS quarterly*, *36*(4).

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., & Buchanan, B. G. (2001a). Evaluation of negation phrases in narrative clinical reports. In *Proceedings of the amia symposium* (p. 105).

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., & Buchanan, B. G. (2001b). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, *34*(5), 301–310.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, *19*(2), 171.

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51–89.

Cisneros, B. (2013). *Accuracy versus f score: Machine learning for the rna polymerases.* Retrieved from `https://www.r-bloggers.com/accuracy-versus-f-score-machine-learning-for-the-rna-polymerases/`

Claxton, A. J., Cramer, J., & Pierce, C. (2001). A systematic review of the associations between dose regimens and medication compliance. *Clinical therapeutics*, *23*(8), 1296–1310.

Collins, G., & Sisk, D. (2017). *Api economy, from systems to business services.* Retrieved from `https://dzone.com/articles/api-trends-for-2017`

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Croft, B., & Lafferty, J. (2013). *Language modeling for information retrieval* (Vol. 13). Springer Science & Business Media.

Cunningham, H., Bontcheva, K., Peters, W., & Wilks, Y. (2000). Uniform language resource access and distribution in the context of a general architecture for text engineering (gate). In *Proceedings of the workshop on ontologies and language resources (ontolex'2000), sozopol, bulgaria.*

Cunningham, H., Tablan, V., Roberts, A., & Bontcheva, K. (2013). Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLoS computational biology*, *9*(2), e1002854.

Cunningham, H., Wilks, Y., & Gaizauskas, R. J. (1996). Gate: a general architecture for text engineering. In *Proceedings of the 16th conference on computational linguistics-volume 2* (pp. 1057–1060).

Das, T., & Kumar, P. M. (2013). Big data analytics: A framework for unstructured data analysis. *International Journal of Engineering Science & Technology*, *5*(1), 153.

Dbpedia. (2017). Retrieved from `http://wiki.dbpedia.org/`

Demirkan, H., & Delen, D. (2013). Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decision Support Systems*, *55*(1), 412–421.

Doan, S., Conway, M., Phuong, T. M., & Ohno-Machado, L. (2014). Natural language processing in biomedicine: a unified system architecture overview. *Clinical Bioinformatics*, 275–294.

Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data.* Cambridge university press.

Ferrucci, D., & Lally, A. (2004). Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, *10*(3-4), 327–348.

Fine, S., Singer, Y., & Tishby, N. (1998). The hierarchical hidden markov model: Analysis and applications. *Machine learning*, *32*(1), 41–62.

Flint, D. (2017). Storms ahead for cloud service providers. *Business Law Review*, *38*(3), 125–126.

Frakes, W. B., & Baeza-Yates, R. (1992). Information retrieval: data structures

and algorithms.

Friedl, J. E. (2002). *Mastering regular expressions*. " O'Reilly Media, Inc.".

Gennegex – a java class to implement wendy chapman's negex algorithm. (2008). Retrieved from `https://github.com/mongoose54/negex/tree/master/GeneralNegEx.Java.v.1.2.05092009`

Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., ... Springsteel, F. (1991). *Ieee standard computer dictionary: Compilation of ieee standard computer glossaries*. IEEE Press.

Gharehchopogh, F. S., & Khalifelu, Z. A. (2011). Analysis and evaluation of unstructured data: text mining versus natural language processing. In *Application of information and communication technologies (aict), 2011 5th international conference on* (pp. 1–4).

Greenes, R. A. (2011). *Clinical decision support: the road ahead*. Academic Press.

Grimes, S. (2008). Unstructured data and the 80 percent rule. *Carabridge Bridgepoints*.

Gupta, V., Lehal, G. S., et al. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, *1*(1), 60–76.

Habash, N. Y. (2010). Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, *3*(1), 1–187.

Hajjar, E. R., Cafiero, A. C., & Hanlon, J. T. (2007). Polypharmacy in elderly patients. *The American journal of geriatric pharmacotherapy*, *5*(4), 345–351.

Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, *85*(1), 6–23.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on nosql database. In *Pervasive computing and applications (icpca), 2011 6th international conference on* (pp. 363–366).

Harris, D. (2017). *The easiest-to-use free/open source text analysis software.* Retrieved from `http://www.softwareadvice.com/resources/easiest-to-use-free-and-open-source-text-analysis-software/`

Haven ondemand. (2017). Retrieved from `https://www.havenondemand.com/`

Haynes, R. B., Hayward, R. S., & Lomas, J. (1995). Bridges between health care research evidence and clinical practice. *Journal of the American Medical Informatics Association*, *2*(6), 342–350.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems*, *19*(2), 4.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, *28*(1), 75–105.

HIMSS. (2017). *Himss dictionary of health information technology terms, acronyms, and organizations* (4th ed.). CRC Press.

Hoerbst, A., & Ammenwerth, E. (2010). Electronic health records. *Methods Inf Med*, *49*(4), 320–336.

Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. In

Ldv forum (Vol. 20, pp. 19–62).

Hripcsak, G., & Albers, D. J. (2013). Next-generation phenotyping of electronic health records. *Journal of the American Medical Informatics Association*, *20*(1), 117–121.

Ibm watson - natural language understanding. (2017). Retrieved from `https://www.ibm.com/watson/developercloud/doc/natural-language-understanding/`

Icd-10 basics. (2015). Retrieved from `http://www.roadto10.org/icd-10-basics/`

Icd purpose and uses. (2016). Retrieved from `http://www.who.int/classifications/icd/en/`

Informatics for integrating biology and the bedside (i2b2). (2017). Retrieved from `https://www.i2b2.org/`

It outsourcing: The reasons, risks and rewards. (2017). Corporate Computer Services, Inc. Retrieved from `http://www.corpcomputerservices.com/articles/outsourcing-reasons`

Jensen, P. B., Jensen, L. J., & Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, *13*(6), 395–405.

Kambatla, K., Kollias, G., Kumar, V., & Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing*, *74*(7), 2561–2573.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28–44.

Kleinberg, B., Mozes, M., van der Toolen, Y., et al. (2017). Netanos-named entity-based text anonymization for open science.

Koh, H. C., & Tan, G. (2011). Data mining applications in healthcare. *Journal of healthcare information management*, *19*(2), 65.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). *Supervised machine learning: A review of classification techniques.*

Kumar, E. (2011). *Natural language processing.* IK International Pvt Ltd.

Kushida, C. A., Nichols, D. A., Jadrnicek, R., Miller, R., Walsh, J. K., & Griffin, K. (2012). Strategies for de-identification and anonymization of electronic health record data for use in multicenter research studies. *Medical care*, *50*, S82–S101.

Labrinidis, A., & Jagadish, H. V. (2012). Challenges and opportunities with big data. *Proceedings of the VLDB Endowment*, *5*(12), 2032–2033.

Lacity, M., Yan, A., & Khan, S. (2017). Review of 23 years of empirical research on information technology outsourcing decisions and outcomes. In *Proceedings of the 50th hawaii international conference on system sciences.*

Lacity, M. C., Willcocks, L. P., & Feeny, D. F. (1996). The value of selective it sourcing. *Sloan management review*, *37*(3), 13.

Leendertse, A. J., Egberts, A. C., Stoker, L. J., & van den Bemt, P. M. (2008). Frequency of and risk factors for preventable medication-related hospital admissions in the netherlands. *Archives of internal medicine*, *168*(17), 1890–1896.

Liddy, E. D. (2001). Natural language processing.

Liggins II, M., Hall, D., & Llinas, J. (2017). *Handbook of multisensor data fusion: theory and practice.* CRC press.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*(1), 1–167.

Liu, H., Christiansen, T., Baumgartner, W. A., & Verspoor, K. (2012). Biolemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of biomedical semantics*, *3*(1), 3.

Manning, C. D., Schütze, H., et al. (1999). *Foundations of statistical natural language processing* (Vol. 999). MIT Press.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Acl (system demonstrations)* (pp. 55–60).

Marr, B. (2016). *How big data is transforming medicine.* Retrieved from `http://www.forbes.com/sites/bernardmarr/2016/02/16/how-big-data-is-transforming-medicine`

Martin, J. H., & Jurafsky, D. (2000). Speech and language processing. *International Edition*, *710*, 25.

Mead, C. (2006). Data interchange standards in healthcare it-computable semantic interoperability: Now possible but still difficult. do we really need a better mousetrap? *Journal of Healthcare Information Management*, *20*(1), 71.

Mehrabi, S., Krishnan, A., Sohn, S., Roch, A. M., Schmidt, H., Kesterson, J., ... others (2015). Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, *54*, 213–219.

Menger, V., Scheepers, F., van Wijk, L. M., & Spruit, M. (2017). Deduce: A pattern matching method for automatic de-identification of dutch medical text. *Telematics and Informatics*.

Meulendijk, M., Spruit, M., Jansen, P., Numans, M., & Brinkkemper, S. (2015). Stripa: A rule-based decision support system for medication reviews in primary care. In *Ecis.*

Meulendijk, M., Spruit, M., Lefebvre, A., & Brinkkemper, S. (2017). To what extent can prescriptions be meaningfully exchanged between primary care terminologies? a case study of four western european classification systems. *IET Software*.

Munger, M. A. (2010). Polypharmacy and combination therapy in the management of hypertension in elderly patients with co-morbid diabetes mellitus. *Drugs & aging*, *27*(11), 871–883.

Musen, M. A., Middleton, B., & Greenes, R. A. (2014). Clinical decision-support systems. In *Biomedical informatics* (pp. 643–674). Springer.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544–551.

Nag, R., Wong, K., & Fallside, F. (1986). Script recognition using hidden markov models. In *Acoustics, speech, and signal processing, ieee international conference on icassp'86.* (Vol. 11, pp. 2071–2074).

Ng, V. (2008). Statistical natural language processing.

Ogren, P. V., Wetzler, P. G., & Bethard, S. (2008). Cleartk: A uima toolkit for statistical natural language processing. *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP*, *32*.

Overview of snomed ct. (2016). Retrieved from

https://www.nlm.nih.gov/healthit/snomedct/snomed_overview.html

Rabiner, L., & Juang, B. (1986). An introduction to hidden markov models. *ieee assp magazine*, *3*(1), 4–16.

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, *2*(1), 3.

Richards, N. M., & King, J. H. (2014). Big data ethics.

Rizzo, G., & Troncy, R. (2012). Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the demonstrations at the 13th conference of the european chapter of the association for computational linguistics* (pp. 73–76).

Rxnorm overview: What is rxnorm? (2017). Retrieved from https://www.nlm.nih.gov/research/umls/rxnorm/overview.html

Saunders, C., Gebelt, M., & Hu, Q. (1997). Achieving success in information systems outsourcing. *California Management Review*, *39*(2), 63–79.

Saverno, K. R., Hines, L. E., Warholak, T. L., Grizzle, A. J., Babits, L., Clark, C., ... Malone, D. C. (2010). Ability of pharmacy clinical decision-support software to alert users about clinically important drug—drug interactions. *Journal of the American Medical Informatics Association*, *18*(1), 32–37.

Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., & Chute, C. G. (2010). Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, *17*(5), 507–513.

Smith, T. (2017). *Api trends for 2017.* Retrieved from https://dzone.com/articles/api-trends-for-2017

Snidaro, L., García, J., & Llinas, J. (2015). Context-based information fusion: a survey and discussion. *Information Fusion*, *25*, 16–31.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australian conference on artificial intelligence* (Vol. 4304, pp. 1015–1021).

Steinman, M. A., Seth Landefeld, C., Rosenthal, G. E., Berthenthal, D., Sen, S., & Kaboli, P. J. (2006). Polypharmacy and prescribing quality in older people. *Journal of the American Geriatrics Society*, *54*(10), 1516–1523.

Stewart, W. F., Shah, N. R., Selna, M. J., Paulus, R. A., & Walker, J. M. (2007). Bridging the inferential gap: the electronic health record and clinical evidence. *Health Affairs*, *26*(2), w181–w191.

Stichting Farmaceutische Kengetallen, S. (2005). Polyfarmacie. *Pharm Weekbl*, *32*, 968.

Stolcke, A., & Omohundro, S. (1993). Hidden markov model induction by bayesian model merging. In *Advances in neural information processing systems* (pp. 11–18).

Tan, A.-H., et al. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases* (Vol. 8, pp. 65–70).

Textrazor. (2017). Retrieved from https://www.textrazor.com/

Vaishnavi, V. K., & Kuechler, W. (2015). *Design science research methods and patterns: innovating information and communication technology.* Crc Press.

Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A comprehensive framework for evaluation in design science research. In *International conference on design*

science research in information systems (pp. 423–438).

Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, *28*(1), 75–105.

Waltz, E., & Llinas, J. (1990). *Multisensor data fusion* (Vol. 685). Artech house Boston.

Weiss, S. M., Indurkhya, N., Zhang, T., & Damerau, F. (2010). *Text mining: predictive methods for analyzing unstructured information.* Springer Science & Business Media.

Wixom, B., Ariyachandra, T., Douglas, D., Goul, M., Gupta, B., Iyer, L., ... Turetken, O. (2014). The current state of business intelligence in academia: The arrival of big data. *Communications of the Association for Information Systems*, *34*(1), 1.

Wright, R. M., Sloane, R., Pieper, C. F., Ruby-Scelsi, C., Twersky, J., Schmader, K. E., & Hanlon, J. T. (2009). Underuse of indicated medications among physically frail older us veterans at the time of hospital discharge: results of a cross-sectional analysis of data from the geriatric evaluation and management drug study. *The American journal of geriatric pharmacotherapy*, *7*(5), 271–280.

Zhang, Y., Qiu, M., Tsai, C.-W., Hassan, M. M., & Alamri, A. (2015). Healthcps: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*.

Zwitter, A. (2014). Big data ethics. *Big Data & Society*, *1*(2), 2053951714559253.