LUT University
School of Engineering Science
Degree Program in Computer Science
Software Engineering Major

Master's Thesis

Simo Partinen

MACHINE LEARNING AS DOCUMENT METADATA TOOL IN E-BUSINESS SERVICES

Examiners: Prof., D.Sc. (Tech.) Kari Smolander
M.Sc. (Tech.) Matti Kosunen

Supervisors: Prof., D.Sc. (Tech.) Kari Smolander
M.Sc. (Tech.) Matti Kosunen

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science
Software Development Major

Simo Partinen

Machine learning as document metadata tool in e-business services

Master's Thesis

2019

70 pages, 13 figures, 5 tables

Examiners: Prof., D.Sc. (Tech.) Kari Smolander
M.Sc. (Tech.) Matti Kosunen

Keywords: machine learning, classification, document metadata, design science

Metadata is a vital part of a digital document. It represents data about the document, which can be used in classifying, indexing and digital document management in general. Despite no lack of tools, some of document metadata is missing or blatantly wrong. Machine learning, a subset of artificial intelligence, utilizes the combination of increase in both computational processing power and amount of available material to learn models depicting the material's characteristic features. A Design Science Research Methodology process was used in this thesis to create a machine learning system, that is capable of deducing the type metadata for a document based on its contents. The system's performance was remarkably good when tested against the material used for training, but classifying the evaluation batch left room for improvement, which most likely wasn't due to the system itself. The system was successfully integrated into an existing digital service platform, but using it in production requires further development iterations.

TIIVISTELMÄ

Lappeenrannan Teknillinen Yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma
Ohjelmistokehityksen pääaine

Simo Partinen

Koneoppiminen asiakirjojen metadatatyökaluna sähköisen asioinnin palveluissa

Diplomityö

2019

70 sivua, 13 kuvaa, 5 taulukkoa

Työn tarkastajat: Prof., TkT Kari Smolander
DI Matti Kosunen

Keywords: koneoppiminen, luokittelu, asiakirjan metadata, design science

Metadata on tärkeä osa sähköistä asiakirjaa. Se sisältää tietoa asiakirjasta, mitä voidaan hyödyntää esimerkiksi luokittelussa, indeksoinnissa ja yleisesti sähköisen asioinnin toiminnoissa. Tarjolla olevista työkaluista huolimatta osa asiakirjojen metadatasta on puutteellista tai suorastaan väärin. Tekoälyn osakokonaisuudeksi luokiteltava koneoppiminen hyödyntää käytettävissä olevan laskentatehon ja materiaalin määrän kasvua oppimalla sille syötetystä materiaalista materiaalia kuvaavia malleja. Tässä diplomityössä kehitettiin Design Science Research Methodology -prosessin avulla koneoppiva järjestelmä, jonka avulla voidaan määrittää asiakirjan tyyppi-metadata sen sisällön perusteella. Järjestelmän suorituskyky oli erittäin hyvä koulutuksessa käytetyllä aineistolla, mutta arviointia varten varatun aineiston luokittelu jätti paljon parantamisen varaa, todennäköisesti järjestelmästä riippumattomista syistä johtuen. Järjestelmä integroitiin onnistuneesti olemassaolevaan sähköisen asioinnin palveluun, mutta sen hyödyntäminen tuotantokäytössä vaatii jatkokehitystä.

PREFACE

A sincere thank you to everyone who has supported, helped and believed in me along my studies.

And a special thank you to Fiia for all the encouragement and cinnamon buns required for me to finish.

Miles olim, miles semper.

Lappeenranta, May 2019
Simo Partinen

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DS	Design Science
DSR	Design Science Research
DSRM	Design Science Research Methodology
EDM	Electronic Document Management
EDMS	Electronic Document Management System
GUI	Graphical User Interface
ICT	Information and Communications Technology
IS	Information Science
IT	Information Technology
ML	Machine Learning
NISO	National Information Standards Organization
NER	Named Entity Recognition
NLP	Natural Language Processing
OCR	Optical Character Recognition
PV-DM	Distributed Memory Model of Paragraph Vectors
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency

1. INTRODUCTION

The following covers the background, motivation, objectives and restrictions, and a brief summary of the thesis' structure.

1.1 Background

According to Merriam-Webster's online dictionary metadata is defined as data providing information on other data. Therefore document metadata is information about the document, stored on the document for convenience as illustrated in figure 1. It can be used to locate and help understanding the document by providing an additional context for the actual content of the document. Figure 1 also illustrates some of the provided context document metadata may include, for example, document author, date of creation, file size and the type of the document.

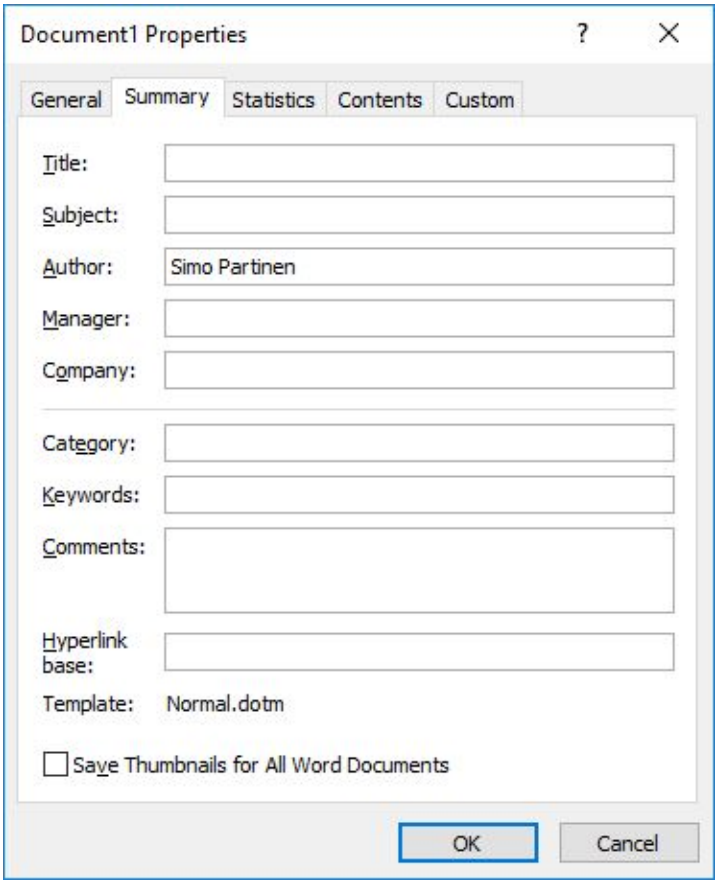

FIGURE 1. A document's metadata fields as shown by Microsoft Word

Definition of document metadata is useful for searching, browsing and filtering (Adefowoke, Sunday Adewale and Oluwole, 2009) and many of the modern text manipulation software provides the user with an option to provide metadata for the edited document. Ideally, metadata would be defined by the document's authors to be used by other systems (Adefowoke, Sunday Adewale and Oluwole, 2009). In a 2003 report on Metadata and Search workshop, the authors argue that despite no lack of tools, author created metadata is often of poor quality and includes incomplete or incorrect information (Crystal and Land, 2003). Thus automated extraction of metadata from document body can be recognized as an important research issue (Adefowoke, Sunday Adewale and Oluwole, 2009; Hui et al., 2003; Liddy et al., 2002; Yilmazel, Finneran and Liddy, 2004).

According to the Finnish Ministry of Finance (2019) digital services, also known as eservices or e-services, are used to increase the opportunities for citizens, companies and corporations to use public services regardless of time and place. In Finland public sector provided digital services are governed by legislation. Section 5 of the Act on Electronic Services and Communication in the Public Sector dictates:

> *"An authority in possession of the requisite technical, financial and other resources shall, within the bounds of these, offer to the public the option to send a message to a designated electronic address or other designated device in order to lodge a matter or to have it considered. Furthermore, the authority shall offer to the public the option to deliver statutory or ordered notifications, requested accounts and other similar documents and messages by electronic means."*

Use of digital services saves public resources (The Finnish Ministry of Finance, 2019; Nikkilä, 2017; Hynninen, Jäske and Tiili, 2015) by enhancing the public service production and thus they are to be held as the most attractive service medium to the clients (The Finnish Ministry of Finance, 2019). As the following Table 1 illustrates, according to the 2014 evaluation of the digital services provided by the city of Helsinki, it was determined that the amount of registered civilian users had multiplied by a 12-fold in the span of four years (Hynninen, Jäske and Tiili, 2015). By the same year, but in the span of just three years, the amount of business users had multiplied by a 17-fold (Hynninen, Jäske and Tiili, 2015). Both of these statistics support a presumption of a trend indicating continuous growth as more digital services are made available to the public.

TABLE 1. Usage data of asiointi.hel.fi (Hynninen, Jäske and Tiili, 2015)

Year	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**
Number of available digital services	2	12	52	50	53	57
Registered private customers		16940	58500	101796	154194	201634
Registered business customers			58	610	815	1003
Total sessions				317369	494148	559064

Despite of the exact nature of a digital service, the key component is relaying an electronic message (Pajukoski, 2004). A message can be perceived as an abstract definition that can be extended to such day to day concepts as email, document, picture or merely a string of text between the recipients. Relaying passive information and automatic replies are often seen as a defining aspect of a digital

service (Pajukoski, 2004). In the context of this thesis digital services consist of government, municipality or bureau provided, information technology aided systems for citizens, companies and corporations forming the main user base.

Machine learning (ML) is used as a blanket term for a set of tools and methods that are capable of adjusting their behavior based on the data they've processed. Rather than using a strict rule-based code to program everything for a system to do, the system learns from the data, similar to how humans process information (Morgan, 2018). At the highest level of abstraction, a machine learning system's core function is to infer patterns and extract insight from a record of the observable world (Conway and White, 2012).

As depicted in the figure 2, machine learning is often perceived as a subset of artificial intelligence (AI). In layman's terms artificial intelligence is used to describe use of system executed algorithms to solve intelligence requiring complex problems. More on machine learning can be read in Chapter 3.

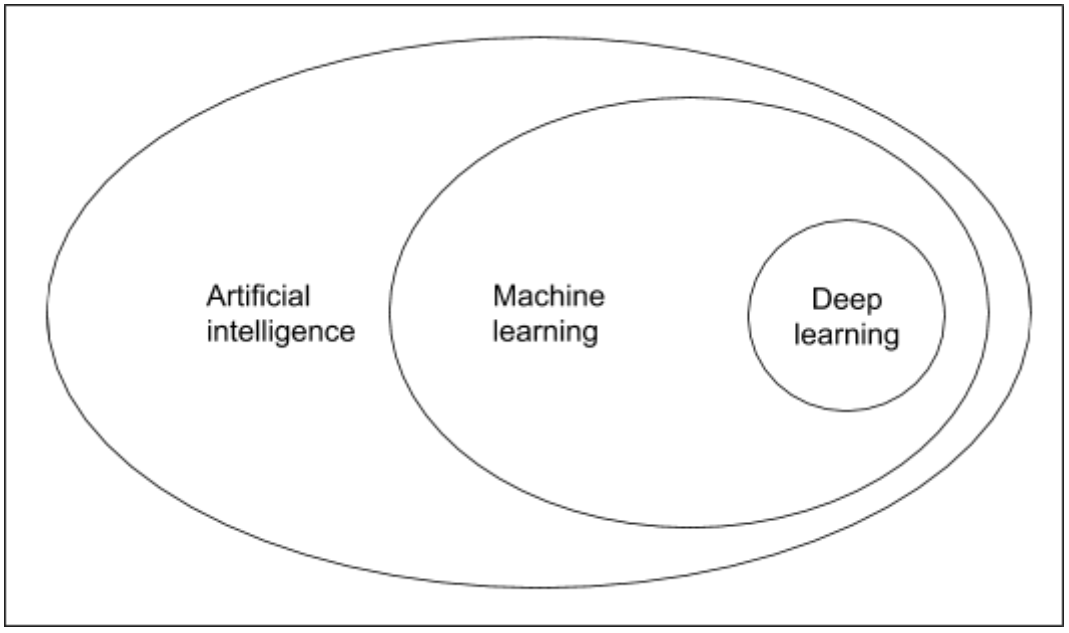

FIGURE 2. Machine learning in relation to artificial intelligence and deep learning

1.2 Visma Consulting Oy

Visma Consulting Oy is a Finnish information technology (IT) company and a part of the Nordic information and communications technology (ICT) focused Visma Corporation. As one of the eight Visma companies in Finland, Visma Consulting focuses on providing consulting services and various software solutions for Finnish corporations and public sector clients. Its core business is service design, digital services, specialised IT-solutions and knowledge management (Visma.fi, 2019). Employing nearly 350 persons and awarded with several financial health excellency proving certificates, Visma Consulting achieved a turnover of over 40 million euros in the fiscal year of 2018 (Visma.fi, 2019).

A rough subject for this thesis was suggested by Visma Consulting because digital services are one of the company's numerous core businesses and there is growing interest in integrating machine learning in both future and existing products and projects. The final artifact of this thesis is proposed to be integrated into an existing digital service platform to help fill the possible gaps (Crystal and Land, 2003) in document metadata. The company also hopes to increase the in-house knowledge on machine learning as a byproduct.

1.3 Objectives and restrictions

The main objective for the thesis, as set by the stakeholders, is to develop and implement a machine learning system that is capable of filling gaps in document metadata in the context of digital services. With Design Science (DS) forming the basic guideline for creating the final artifact, and the research question being strictly limited by the main objective of the thesis, the main research question is:

> *Is it possible to create a machine learning system, which can be used to fill gaps in document metadata?*

The main research question is backed by sub-questions:

> *How to integrate a machine learning sub-system into an existing system?*

> *How to implement extensibility in a machine learning system?*

Due to the fact that the thesis artifact is planned to be integrated into a time and budget constrained client project, the thesis artifact's functionalities to be implemented are limited by the client project constraints. Therefore the thesis artifact will focus on filling missing document type metadata only. The artifact is to be designed with future integrability and extensibility in mind.

1.4 Methodology

The development process of the thesis artifact follows the guidelines of Design Science Research Methodology (DSRM) (Peffers et al., 2007). DSRM's foundations lie in Design Science, which was first described in context of information systems research by a group of researchers lead by Alan Hevner in 2004. The original paper (Hevner et al., 2004) does not present a model or process for performing Design Science Research (DSR). However, Hevner himself (2007)

and others, most notably Peffers et al. (2007) in the context of this thesis' methods, have driven the concept further.

Chapter 4 offers a deeper insight on the used DSRM process.

1.5 Structure of the thesis

The rest of the thesis is structured as follows: Chapter 2 offers a brief introduction on document management and the role document metadata has in it. Chapter 3 aims to offer enough information on machine learning to familiarize the reader with the concepts under the hood of the system to be implemented. Before diving into the implementation process of the artifact, Chapter 4 offers more insight on the theory behind the artifact's implementation. Chapter 5 describes the generation process of the artifact and the artifact itself in relation to how it's implementation was planned earlier in the Chapter 4. Chapter 6 contains discussion based on the generation process of the artifact such as how implementable the system is in the given context, analysis on its evaluation and possible future research prospects. Chapter 7 concludes the thesis and is followed only by the list of references and the list of appendices.

2. DOCUMENT MANAGEMENT AND METADATA

Both electronic document management (EDM) and metadata are of great importance in modern society. The following chapter focuses on describing the two in the context of this thesis.

According to a global study, conducted in 2015 by the International Data Corporation (Webster, 2015), many of today's enterprise business processes require a final step that is a disconnected, discontinuous experience from the rest of the process. The report of the study calls this the "last mile". The last mile often requires exchanging information in document form as documents are "how people communicate ideas, share information and record their understandings" (Webster, 2015). For the aforementioned study, the International Data Corporation interviewed over 1500 business leaders, IT leaders and information workers in Western industrial countries. The study revealed 51 % of the interviewees having observed documents getting misfiled or lost, 36 % admitting to having cited agreements that have lacked information and 46 % being unsure whether or not copies exist of all of the signed agreements (Webster, 2015). Most importantly from a business point of view, the study states that 46 % of the business leaders claim ineffective document processes impairing their ability to plan, forecast and budget. As for the information workers the study revealed that approximately 80 % of their time is spent working with documents (Webster, 2015).

In construction industry only, a single project may accumulate a very large amount of documents, with the total number of the industry's documents and online available documentable data resources continuously increasing over time (Hjelt and Björk, 2006; Pathirage, Amaratunga and Haigh, 2007; Shin, 2015). Craig and Sommerville (2006) approximate that some 7200 documents may be

generated for a single construction project consisting of useful information and insights (Ma, La and Wu, 2011; Soibelman et al., 2008). Besides emerging online resources (Pathirage, Amaratunga and Haigh, 2007; Shin, 2015), regulatory and compliance pressures are also contributing factors to the ever accumulating amount of documents (Churchill, 2019). Despite the surge in total amount of documents, paper documents have not completely been overthrown by their digital alternatives due to tangible paper's several useful properties. Paper documents' ease of navigation and their annotativity are among those identified by Sellen and Harper in their paperless office study concluded in 2011. In his article on online document management for Infonomics magazine, Chris Churchill goes on to list improved control and compliance, quicker access to information and greater processing efficiencies that drive cost savings as some of the benefits of e-documents. Jervis and Masoodian (2017) also acknowledge the same benefits recognized by Churchill.

Electronic document management systems (EDMS) can be used to support the end users in information processes. Their aim is to provide simple, logical and quick ways of storing, finding and retrieving documents in an electronic format (Degerstedt, 2000; Löwnertz, 1998). Hjelt and Björk (2006) identify that the internet and its protocols have played a crucial role in enterprising these systems from the mere tailor-made network solutions they were before the internet. Case studies conducted by Sulankivi, Lakka and Luedke (2002) indicate improved speed, quality and cost efficiency in EDMS supported information processes. Thus it can be deducted that accompanying e-documents with an EDMS, may result in positive results by streamlining business processes via workflow and information sharing, which, according to Hammer and Hershman (2010) can also improve information management. The two also argue that electronic document management system implementation can resolve complex business problems, deliver real competitive advantage and transform organisations. An electronic document management system enables organizations to manage their documents

throughout their lifecycle from a draft into an archive (Jones, 2012). A successful implementation of an EDMS does, however, require strong management commitment, detailed document management guidelines and sufficient training (Sulankivi, Lakka and Luedke, 2002; O'Brien, 2000).

Metadata is often described only as data about data. United States based National Information Standards Organization (NISO) primer provides us with a more thorough definition:

> *"[Metadata is] structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource."* (Riley, 2004)

In the updated primer metadata is defined as:

> *"[Metadata is] the information we create, store, and share to describe things, [which] allows us to interact with these things to obtain the knowledge we need"* (Riley, 2017)

In the latest primer, NISO has also identified and provided examples of several different metadata types listed in table 2. Descriptive metadata is, as the name suggests, descriptive to the resource's properties providing data like title, author and date of publication. Technical metadata denotes the resource's technical properties such as file size and type. Preservation metadata can be used to describe the integrity monitoring checksum and the conditions regulating the resource's preservation. Rights metadata explains the resource's rights, e.g. copyright and distribution information. Structural metadata can be used to define how to fit together partitioned resources. Finally the more seldom used markup languages metadata is used to denote notable features in the content such as paragraphs and text formatting in a textual resource

TABLE 2. Metadata types, example properties and use cases (Riley, 2017)

Metadata Type	Example Properties	Primary Uses
Descriptive metadata	Title Author Subject Genre Publication date	Discovery Display Interoperability
Technical metadata	File type File size Creation date/time Compression scheme	Interoperability Digital object management Preservation
Preservation metadata	Checksum Preservation event	Interoperability Digital object management Preservation
Rights metadata	Copyright status License terms Rights holder	Interoperability Digital object management
Structural metadata	Sequence Place in hierarchy	Navigation
Markup languages	Paragraph Heading List Name Date	Navigation Interoperability

Kip Wolf (2008) of Tunnel Consulting argues that document metadata is as valuable as the content, when it comes to managing and creating electronic documents. According to him, every electronic document management system's fundamental principle is based on the document objects consisting of the content and the metadata of the documents as illustrated in Figure 3. In a broader, EDMS linked sense, metadata can be described as the core set of elements needed for the effective retrieval and management of information (Sprehe, 2004).

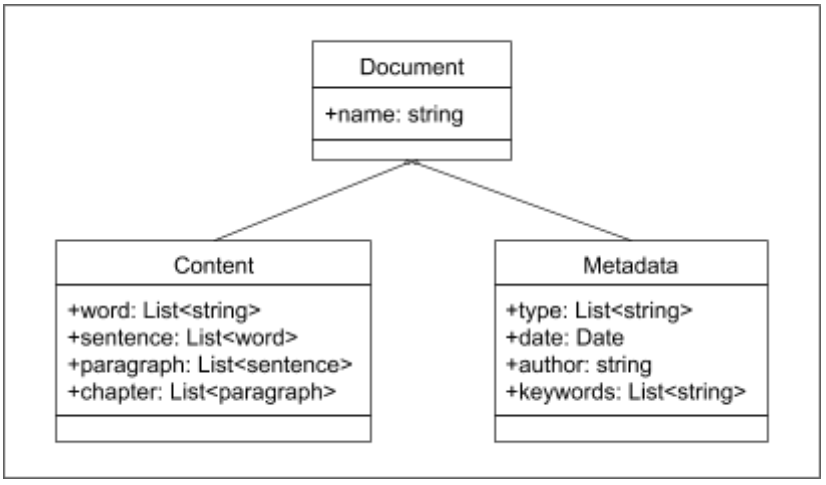

Figure 3. A simplified class diagram of a document object consisting of content and metadata (adapted from Wolf, 2008)

Losing (for prevalence see Webster, 2015) any of the components illustrated in figure 3, may lead to losing value of the object altogether (Wolf, 2008). Due to the importance of metadata and its usefulness in searching, browsing and filtering e-documents (Adefowoke, Sunday Adewale and Oluwole, 2009), storing, monitoring and altering metadata can be considered as a standard EDMS feature. Although it's setup can be partially automated and research (Adefowoke, Sunday Adewale and Oluwole, 2009; Hui et al., 2003; Liddy et al., 2002; Yilmazel, Finneran and Liddy, 2004) suggests a need for further input automation, some of the metadata fields still need to be input manually (Adefowoke, Sunday Adewale and Oluwole, 2009) possibly causing human errors (Crystal and Land, 2003).

3. ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

The following offers a brief glimpse into the basics behind machine learning; what is machine learning, why is it seemingly such a big thing all of a sudden and how is machine learning being implemented nowadays. The chapter also contains information on machine learning in the context of this thesis by describing the used tools and methods.

3.1 Overview

The term artificial intelligence originates from the 1950s. AI was first discussed in its current form at three occasions. First at the 1955 Session on Learning Machines held in conjunction with Western Joint Computer Conference in Los Angeles, then at the 1956 Dartmouth Summer Research Project on Artificial Intelligence and finally at the 1958 Mechanization of Thought Processes symposium held in London (Solomonoff, 1985; Moor, 2006; Nilsson, 2009, Russell and Norvig, 2010). Since its birth, effective unsupervised learning, sometimes hailed as artificial general intelligence (AGI) to describe it's lack of field limitations, or even as artificial superintelligence (ASI), the hypothetical ability of a machine to far surpass the human brain, has been regarded as the holy grail of AI. With the current technology, however, AI has yet to reach the sophistication level of it's biological counterpart, and it is uncertain whether or not the current AI implementations, e.g. deep learning algorithms, are the correct path towards AGI (Russell and Norvig, 2010; Sabour, Frosst and Hinton, 2017; Morgan, 2018). Patterson and Gibson (2017) even suggest refraining from talking about machine learning as AI to avoid brain analogies.

Although the artificial general intelligence and it's derivatives still remain as a frontier yet to be reached, the current AI boom has seen the emergence of the narrow AI: systems capable of performing individual tasks, around which also this thesis revolves. According to World Intellectual Property Organization (WIPO) the current popularity extends a series of ups and downs often referred as AI summers and winters with over half of the identified inventions dating after the year 2013. WIPO also states that the ratio of scientific papers to inventions has decreased from 8:1 in 2010 to 3:1 in 2016, indicating the AI's shift from theoretical to practical implementations (WIPO, 2019). In WIPO Technology Trends 2019 - Artificial Intelligence -report Andrew Ng, the CEO of Landing AI and deeplearning.ai, already recognizes AI as a creator of economic wealth (also Grech et al., 2018) and AI is predicted to add $15,7 trillion to the global economy by 2030 (Morgan, 2018).

The new AI opportunities have opened up as large amounts of data to be compiled and shared have become accessible. Another contributing factor is the spread of low-cost graphics processing units (GPU), capable of huge computational loads (Chio and Freeman, 2018; Grech et al., 2018; Osinga, 2018; WIPO, 2019). AI's entrance to the global marketplace from the theoretical realm is thus fueled by a combination of digitized data and rapidly advancing computational processing power (Russell and Norvig, 2010; Patterson and Gibson, 2017; Grech et al., 2018; Osinga, 2018; WIPO, 2019). This growth, combined with advances in global connectivity (WIPO, 2019), has acted as the seed for the current AI summer. It can be speculated that the increased number of AI based solutions will generate more economic wealth, thus increasing the amount of research and development funds directed towards new AI implementations. This in turn has the possibility of creating a positive feedback loop.

To better grasp machine learning, it is important to define what learning means in its context. Patterson and Gibson (2017) describe learning as, "gaining knowledge

by studying, experience or being taught". The Nobel Prize-winning economist Herbert Simon has stated that, "Learning is any process by which a system improves its performance from experience" (Pustejovsky and Stubbs, 2012). According to Russell and Norvig (2010), the modern grandfathers of AI (Patterson and Gibson, 2017), a learning agent is capable of improving its performance based on previous experiences. The two also list three motivators for learning agents: the impossibility to anticipate all of the agent's possible future scenarios, the impossibility to predict all changes over time and the occasional human designers' incapability to program a solution themselves.

Russell and Norvig (2010) define the three main types of machine learning as follows. In unsupervised learning the training data is used to teach the agent existing underlying patterns without explicit feedback. It is a technique implemented by the likes of support vector machines, clustering and feature selection and transformation (Nilsson, 2009; Kirk, 2017). The most common type of machine learning (Kirk, 2017), supervised learning, has the agent observing input-output pairs and learns a function that maps from input to output. Methods such as decision trees and neural networks are prime examples of supervised learning (Nilsson, 2009; Kirk, 2017). Lastly reinforcement learning is about teaching the agent with the help of rewards or punishments with important applications ranging in hundreds (Nilsson, 2009; Kirk, 2017). They can be thought as algorithms that optimize the life of something (Kirk, 2017).

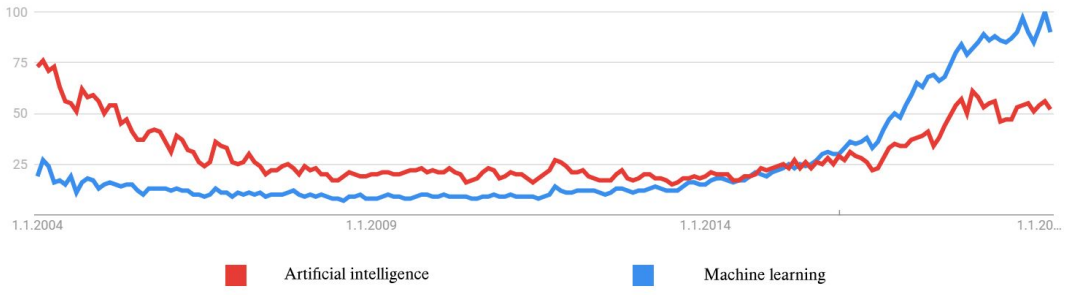

FIGURE 4. Search frequencies for artificial intelligence and machine learning

WIPO's report acknowledges machine learning as the most prominent of the AI related innovations with ML being involved in over a third of all identified inventions (2019; also Grech et al., 2018). The popularity of machine learning can also be observed in figure 4, depicting Google searches of both artificial intelligence and machine learning, where ML has far surpassed AI as a search trend. Haifen Wang, the Senior Vice President of Baidu, recognizes that for nearly the past ten years deep learning in particular has been well studied and significant progress has been made (WIPO, 2019). Kazuyuki Motohashi, a professor in the University of Tokyo, states that the idea for the deep neural networks comes from the mechanism of how the human brain works and that some of the progress in AI field has actually been driven by the interaction between computer science and cognitive science (WIPO, 2019).

Fundamentally machine learning comes down to representing data in some type of model with the help of algorithms (Patterson and Gibson, 2017) and constructing hypotheses from data (Nilsson, 2009). Malcolm Johnson, the Deputy Secretary-General of International Telecommunication Union acknowledges that AI's key advantage is its ability to analyze big data and identify patterns and correlations that might otherwise pass unnoticed. In the same report Andreessen Horowitz partner Frank Chen also supports Johnson's notion by stating: "The technique is basically: give me examples and I will figure out which ones are relevant. The bigger the set of data you have, the better predictions you make." (WIPO, 2019). Conway and White (2012) add to this by stating that machine learning is learning from the subject's records and then creating a model of that will inform our understanding of this context going forward.

3.2 Classifying

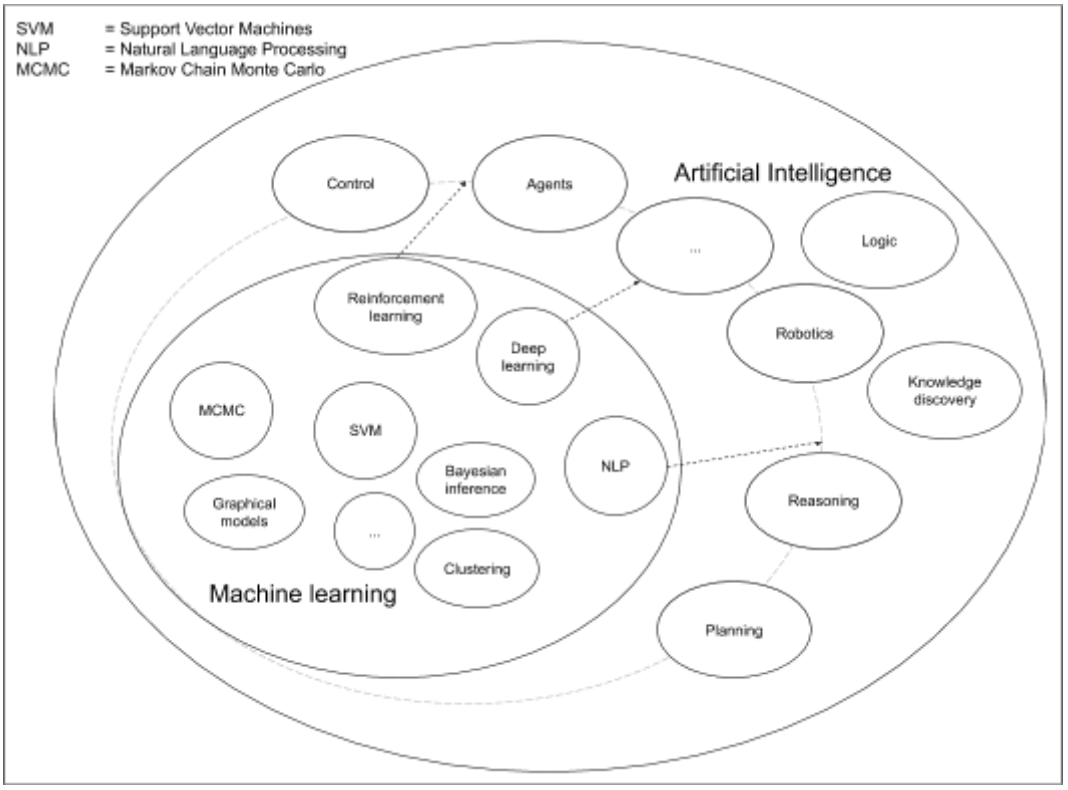

FIGURE 5. Illustration of machine learning in the field of AI (adapted from Grech et al., 2018)

Artificial neural network (ANN) assisted deep learning, visible in figure 5, is an important subset of machine learning. According to WIPO's patent filings, deep learning and neural networks are the fastest growing AI techniques with 175 % and 46 % respective growth rates from 2013 to 2016 (WIPO, 2019). The growth in interest is also visible in GitHub, a collaborative open source software development platform. The repository data with mentions of deep learning and neural networks have skyrocketed from 238 mentions of deep learning and 43 mentions of neural networks in 2014 (WIPO, 2019) to 116 995 and 129 154 mentions in 2019 respectively. Inspired by the human brain and its neurons, deep learning is especially predicated on an idea of learning from example. It has multiple definitions but they all revolve around this same idea. Patterson and

Gibson (2017) describe deep learning as a “neural network with more than two layers” and “neural networks with a large number of parameters and layers”. So to understand deep learning, one has to grasp the concept of neural networks. Neural networks, such as the feed-forward multilayer neural network in figure 6, consist of a series of layers which in term consist of artificial neurons called perceptrons or nodes. Much like their biologic counterparts, each perceptron accepts an input or multiple inputs, proceeds to process the given data and then relays the result for the next layer of perceptrons to process.

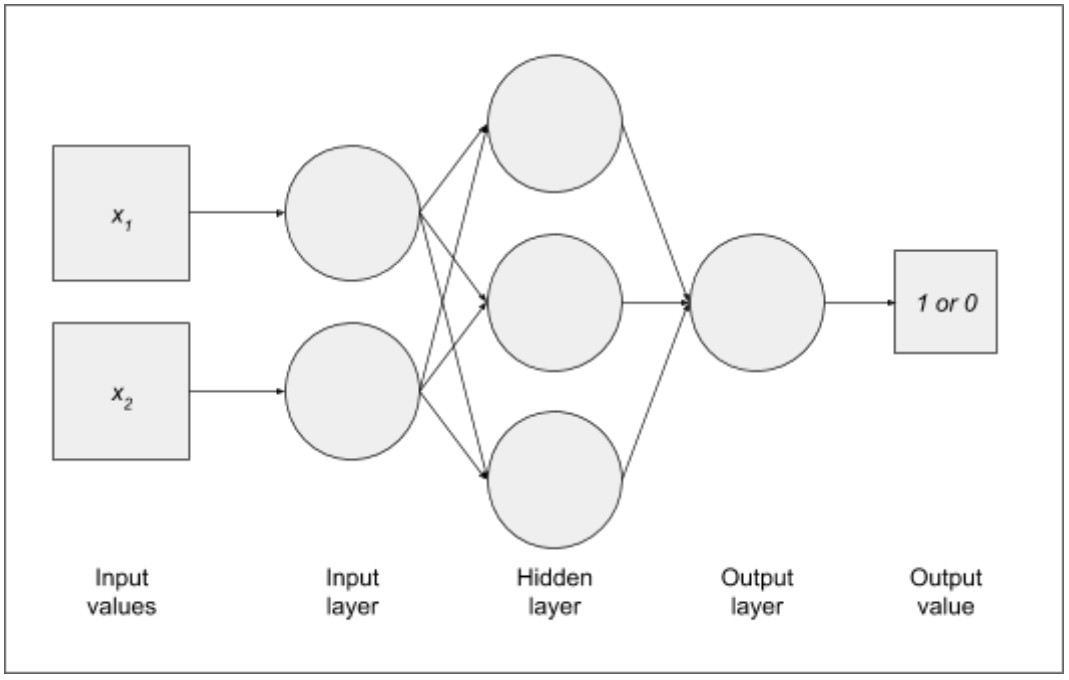

FIGURE 6. A multilayer neural network

The connections between each perceptron are called weights. Adjusting the weights with the help of a back propagation (Nilsson, 2009) adjusts the importance of that input in terms of that perceptron’s output by minimizing error. Because the data given to a neural network varies, its weights are the closest thing to long-term information storage available. Besides the weights and the data given, a perceptron also contains a bias value aimed to ensure activation of certain

nodes. An activation function combines the inputs, weights and biases, transforming them into an input for the next perceptrons in line.

The Figure 5 is also depicting machine learning's growing importance in the AI field (Grech et al., 2018). Besides the expansion driving artificial neural networks, it also illustrates several other important machine learning methods. Especially during the declined interest of ANNs in the 90's and early 2000s, the more easily approachable models like support vector machines (SVMs) and decision trees became popular (Osinga, 2018).

Support vector machines are used to boost supervised machine learning methods (Nilsson, 2009; Kirk, 2017). SVMs are used to maximize the generalization of training samples by minimizing the expected generalization loss (Russell and Norvig, 2010; Kirk, 2017). Figure 7 shows two groups of training samples with one group consisting of squares and the other consisting of dots.

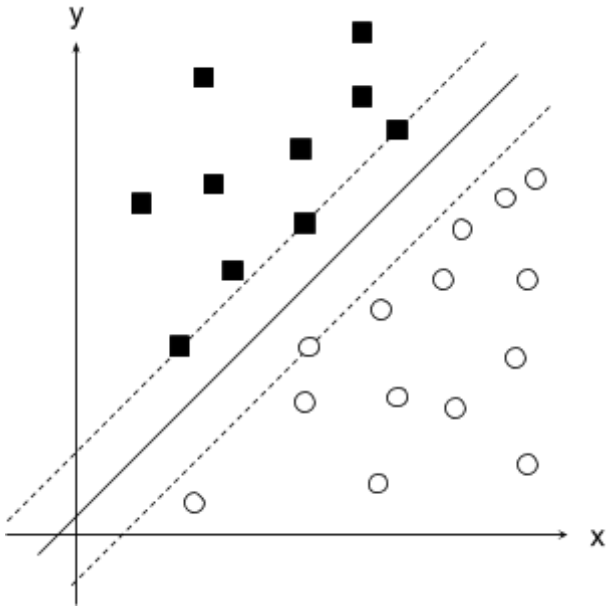

FIGURE 7. Separating points with a linear boundary (adapted from Nilsson, 2009)

The figure also has three linear boundaries. The two dotted lines that intersect a training sample or multiple samples represent the minimum margin separator and the third, solid line represents a maximum margin separator (Russell and Norvig, 2010). Without going too deep into the mathematics behind SVMs, their key insight is that some samples are more important than others (Russell and Norvig, 2010). By minimizing the expected generalization loss the samples yet to be plotted, are more likely to end up in the correct category. With the help of kernel tricks, which is essentially changing the projection of the data, SVM's can be applied even in data that is not linear (Kirk, 2017).

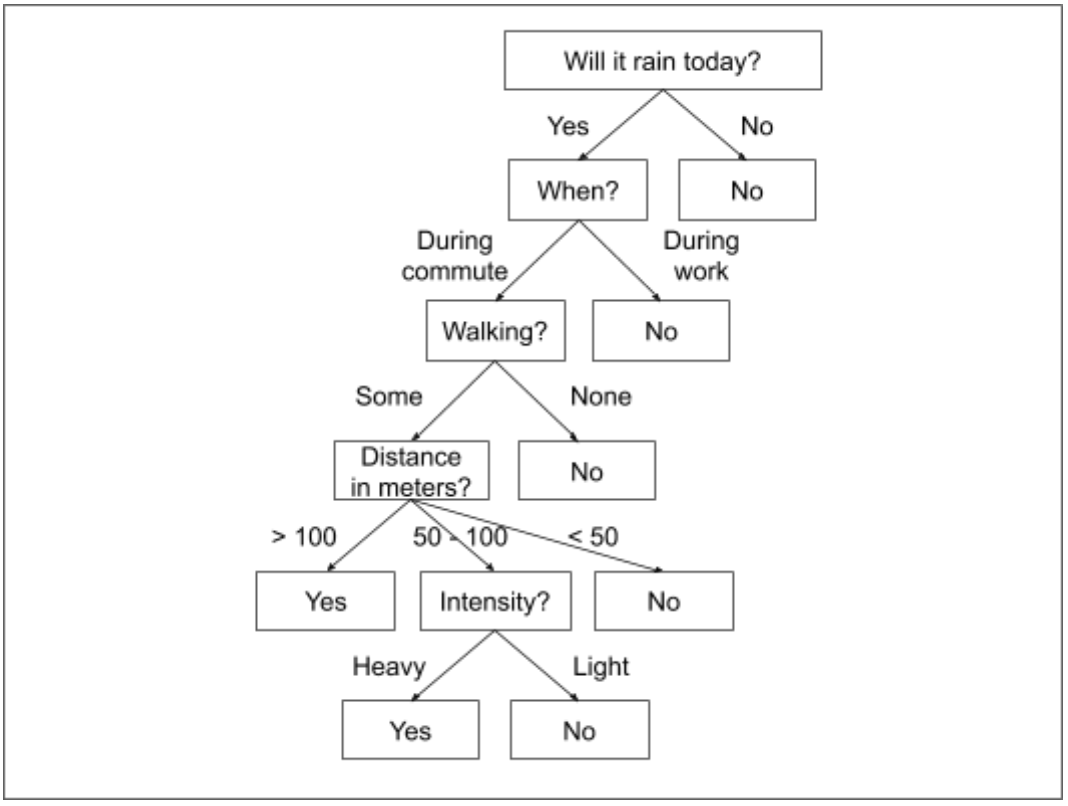

FIGURE 8. A simple decision tree for whether to bring an umbrella to the work or not

Decision tree learning is simple, yet successful implementation of machine learning (Russell and Norvig, 2010) and the trees themselves can be constructed automatically from large databases (Nilsson, 2009). Consisting of multiple binary

(Chio and Freeman, 2018) test sequences as shown in Figure 8, the goal is to create a model predicting the decision based on an input vector. In some cases decision tree learning may suffer from a problem called overfitting. Basically an overfitted model has been brute-forced out of the noise in training data although there should be none. Such situation may occur if the hypothesis space and the number of input attributes grows (Russell and Norvig, 2010). To combat overfitting, a technique called pruning may be used to eliminate clearly irrelevant nodes (Russell and Norvig, 2010; Kirk, 2017) resulting in a subtree inside the full decision tree. Understandably in the more complex decision trees the amount of possible subtrees can become overwhelming, making it more demanding to find the most optimal subtree.

Ensemble methods can be used to overcome the problems spawning from complexity. They can be thought as meta-programming for machine learning: a model may consist of submodels (Kirk, 2017). In the context of decision trees, an ensemble method called bootstrap aggregated, often shortened as bagging, random forest has been proven fit (Nilsson, 2009; Kirk, 2017). The idea of bagging is to improve the model by changing only the training set by aggregating multiple random versions of the training set or feature space and using these aggregates to train additional classifiers (Nilsson, 2009; Kirk, 2017). Once the training data has been bagged, and all of the classifiers have been trained, a final classification can be made by a majority vote. When bagging is applied to decision tree learning, one ends up with a forest of randomized decision trees - a random forest.

Logistic regression is a classifying method used to locate boundaries in discrete classes in given data with the help of a sigmoid function (Burger, 2018). Using numerical input vectors, it approximates the log odds for each data point. The log odds can be calculated as $log\ p/(1-p)$, where p is the event's occurrence probability (Chio and Freeman, 2018). The calculated value can be used to determine a given data point's class (Buduma and Locascio, 2017; Burger, 2018).

3.3 Natural Language Processing (NLP)

In the context of this thesis it is also important to discuss machine learning's application against textual vectors. Natural languages entail the languages used in interhuman communication (Pustejovsky and Stubbs, 2012). Computerized understanding, generating, translating and conversing (Nilsson, 2009) in natural languages in AI context can all be engulfed under the term natural language processing. Because of the natural languages' size and ever changing nature, even the best NLP models are merely approximations (Russell and Norvig, 2010). According to Russell and Norvig (2010) there are two reasons for the interest in computer agents capable of natural language processing: first, to communicate with humans and second, to acquire information from textual data, with the latter of the two being more of interest in the context of this thesis. Pustejovsky and Stubbs (2012) mention document classification as one of the most successful NLP areas due to the relative simplicity of the learning models needed for classification algorithms. The role of categorization in modern society can be speculated to be aggravated due to the sheer amount of spam in e-mail communication.

Before actually modeling the source material with the help of machine learning, a series of preprocessing steps may be taken. A common first measure is to remove stop words such as various articles and short function words with no contextual information value or semantic (Rajaraman, Ullman and Leskovec, 2014). Tokenization is the task of chopping up a document into character sequences called tokens. A token refers to a character sequence in a particular context that are grouped together as a semantic unit for processing (Manning, Raghavan, and Schütze, 2008). All identical character sequences in a document form a type. An example phrase "*to exercise harder to live"* contains five tokens, but only four types. If stop words are removed one is left with only three tokens: *exercise, harder* and *live*. Next up the tokens can be stemmed and lemmatized to locate the root words by reducing the inflectional or related form (Manning, Raghavan, and

Schütze, 2008). Applying either of these to the example tokens would reduce *harder* into its root word *hard*. However, there is a difference between the two methods. Stemming usually relies on a crude heuristic process that chops off the ends of words, sometimes ending up completely butchering the originals, whereas vocabulary and morphological analysis relying lemmatization aims to remove inflectional endings only, leaving the lemma, the dictionary form of a word, intact (Manning, Raghavan, and Schütze, 2008). Lastly Named Entity Recognition (NER) may prove useful in cases when the data contains a lot of tokens that by themselves hold little value for the context. NER is used to label such tokens in a way that it enriches the context (Pustejovsky and Stubbs, 2012) by describing the type of the token, e.g. name, location, date or sum.

N-gram models are based on the characteristics of natural language. In natural languages a body of text has multiple sentences which in term are formed from sequences of words that consist of characters. N-gram models refer to sequences of written symbols of *n* length. Thus a sequence of two symbols or words would be referred as bigram and a sequence of three would be referred as trigram and so forth. A model of the probability distribution of n-letter sequences is thus called an n-gram model (Russell and Norvig, 2010). N-gram character models can be used, for example, for language identification whereas n-gram word models are more suitable for text classification purposes (Russell and Norvig, 2010) with the help of classification methods such as those discussed in the previous chapter. Bag-Of-Words (BOW) model is an often used implementation of a unigram. In BOW all of the words in training corpus, the annotated dataset, are indexed and used as a vector to train a machine learning model. This however comes with the cost of losing the notion of the words (Russell and Norvig, 2010; Le and Mikolov, 2014).

Doc2vec and word2vec are both NLP tools proposed and developed by Google (Mikolov et al, 2013; Le and Mikolov, 2014). They both employ the same

principle of representing textual structures, e.g. a word, a paragraph or a document, as a vector concatenated or averaged with other vectors in a context. The resulting vector can then be used to predict other structures in context. Especially Doc2vec that is based on Paragraph vector by Le and Mikolov (2014) performs significantly well with a 32% relative improvement in terms of paragraph feature calculation error rate in comparison to other state of the art methods such as Bag-Of-Bigrams or Bag-Of-Words (Le and Mikolov, 2014).

4. METHODOLOGY

This chapter describes the methodology process and how it was implemented in practice during the research process.

4.1 Design Science Research Methodology

Design Science Research Methodology was chosen as the main resource to help design and implement the thesis artifact. When Alan Hevner first described Design Science in the context of information systems research, he defined that its most important guideline is to produce an artifact created to address the problem (Hevner et al., 2004). Hevner, somewhat of a Design Science pioneer, along with his colleague Samir Chatterjee further described Design Science Research as a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. They state that the designed artifacts are both useful and fundamental in understanding that problem (Hevner and Chatterjee, 2010). Its applicability in solving human problems via software makes it a suitable option for situations where humans interact with software systems.

To further illustrate the nature of the artifact creation, Antti Knutas mentions in his doctoral thesis that a design science research process describes a pipeline where the desired artifact is created through an iterative design and evaluation process (2016). A prototype, created at the end of each iteration, is used to evaluate the current design until the predefined requirements of the artifact have been met. The thesis artifact's performance and its implementation will be reviewed by the author in collaboration with the head software architect of Visma Consulting Oy.

Peffers et al. acknowledge that design science is of importance in a discipline oriented to the creation of successful artifacts with several researchers pioneering DS research in information science. However, by 2007 only some DS research had been done within the discipline itself. In their paper Peffers et al. describe a methodology to serve as a framework for DS research and a template for its presentation (2007). The Design Science Research Methodology incorporates principles, practices, and procedures required to carry out such research and meets three objectives: it is consistent with prior literature, it provides a nominal process model for doing DS research, and it provides a mental model for presenting and evaluating DS research in IS. (Peffers et al., 2007) The defined process includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. (Peffers et al., 2007) Figure 9 illustrates the process, chosen as a guideline for this thesis, depicting the six aforementioned parts:

Activity 1. Problem identification and motivation

As a problem with a need to solve it arises, the problem should be defined and atomized to help developing and evaluating the solution providing artifact. By atomizing the problem, its complexity becomes clearer which in term helps justify the value of a solution. This activity requires knowledge of the state of the problem and the importance of its solution. (Peffers et al., 2007)

Activity 2. Define the objectives for a solution

From the basis of the identified problem, and knowing the possibilities and constraints, a set of objectives that the solution should meet is defined. The objectives should be inferred from the problem specification. This activity requires knowledge of the current state of problems and solutions. (Peffers et al., 2007)

Activity 3. Design and development

Implementation of the solution in the form of a design research artifact. All of the designed objects that are embedded in the design thanks to prior research on the subject, can be treated as design research artifacts. This activity requires theoretical knowledge that is used to form the artifact or artifacts. (Peffers et al., 2007)

Activity 4. Demonstration

The artifacts capabilities are showcased against the defined problem. This activity requires knowledge on how to use the artifact in the problem context. (Peffers et al., 2007)

Activity 5. Evaluation

Observing and measuring the artifacts capability in solving the problem. Its performance should be compared to the objectives of a solution. The possible metrics to be evaluated are numerous. They include, for example, objective quantitative measures such as improvements in task execution time, or more subjective, qualitative analysis in the form of client feedback or user satisfaction surveys. This activity determines whether or not to iterate back to activity 3 to improve the artifact further or to continue on. This activity requires knowledge of relevant metrics and analysis techniques. (Peffers et al., 2007)

Activity 6. Communication

Diffusing the resulting knowledge to project stakeholders. The problem and its importance and the artifact, its capabilities and implementation are communicated to the relevant audiences. This activity requires knowledge of the disciplinary culture. (Peffers et al., 2007)

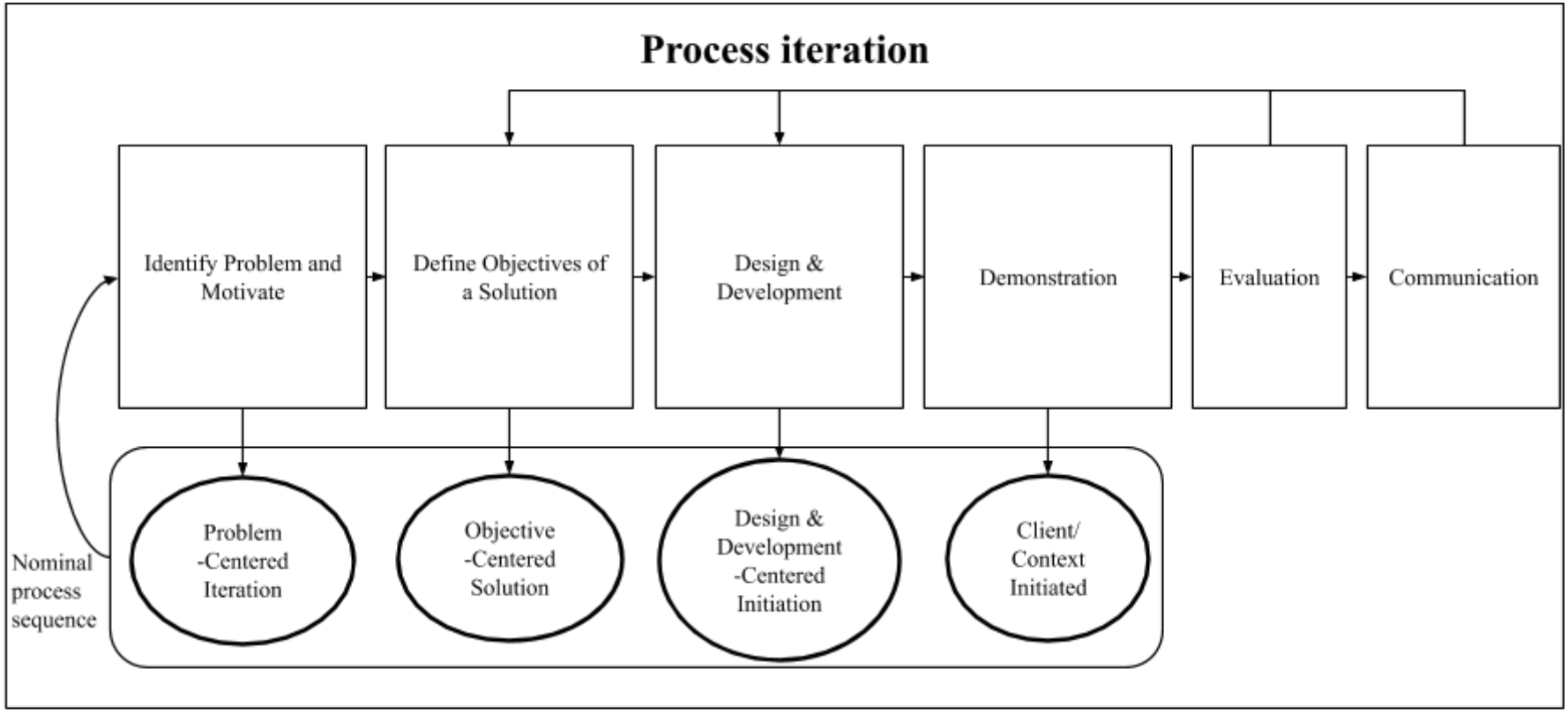

Figure 9. DSRM Process Model (adapted from Peffers et al., 2007)

Peffers et al. note that there is no expectation to proceed in sequential order through the activities. As depicted in the Figure 9, the process may start at almost any step and span outward (Peffers et al., 2007). A solution based on DSRM process beginning at activity 1 is called a problem-centered iteration. It often spans from a recognized problem. An objective-centered solution begins from activity 2. It may be initiated by recognizing that an artifact is needed to resolve an encountered situation. Third solution is a design- and development-centered approach starting respectively at activity 3. Design- and development-centered solution may be of option in situations where an already existing solution or artifact is deemed fit to resolve another, differentiating problem. The fourth and final solution is called a client-/context-initiated solution. It starts from activity 4 and requires applying the DSRM process retroactively to end up with a DS solution.

4.2 Applying DSRM

Because this research was spawned from a recognized problem, the process was to be iterated as problem-centered. Thus the process would progress from the leftmost activity in figure 9 to the rightmost with the possibly required iterative steps included.

A client chose an existing digital service platform as a solution for managing and archiving documents created for various projects. The selection was followed by a series of meetings between the stakeholders regarding the more specific requirements and planned use cases for the solution. Mapping the requirements and use cases would act as the problem identification and motivation step of the DSRM process.

A machine learning system was proposed to overcome the problem of unknown metadata. ML was chosen instead of a purely algorithmic solution because of the varying nature of different document types, the availability of labeled training

data and the fact that the client was also interested in machine learning capabilities. In an agreement with the client, a proof of concept -project, acting also as a basis for this thesis, was spawned to explore the feasibility of machine learning as a tool to fill gaps in document metadata.

The design and development step of the process would consist of several parts. A sequence diagram was generated to illustrate the general flow of the system. The identified problems and their solutions were used as the main guideline for the illustration. Following the creation and approval of the sequence diagram, the development process was started by gathering the documents required to train the machine learning system. However, before using the documents in training the machine learning models they would need to be converted into a plain text format with `Tesseract` optical character recognition (OCR) software. This step was required because the models would be trained to classify documents based on their textual content.

Due to its popularity in developing machine learning solutions and the number of available supporting libraries, the machine learning system was chosen to be developed with Python with the help of its multiple auxiliary libraries. The development would begin with descriptive analysis of the training material to gain a better understanding of the material at hand. It was also required to develop the methods to train the machine learning models. These methods would not only be used to create the initial models, but their implementation was also crucial to updating the models used by classifiers later on. Simplicity and previous success (Russell & Norvig, 2010) were the reasons for using bootstrap aggravated decision tree (See Chapter 3.2 Classifying), also known as Random forest, as one of the classifiers. It was also known that its operation could be boosted with the use of different vectorizations. Artificial neural network was chosen as the second model due to their recent popularity and versatility (WIPO, 2019).

Once trained, the capabilities of the three different models would be tested and evaluated with a material batch consisting of 185 varying documents supplied by the client. The material was not created by the client themselves, but by their contractors. The results generated by the demonstration and evaluation would represent a portion of the presentation for the stakeholders with the other portion being demonstrating the machine learning system in use.

In order to demonstrate the machine learning system in action, it was to be integrated with a digital service platform. To do this, the trained models were saved and the Python's `Tornado` library (Tornadoweb.org, 2019) was used to create a simple local web server. `Tornado` was chosen due to its simplicity, scalability (Tornadoweb.org, 2019) and the fact that there was no heavy computational load expected. The web server exposes an API endpoint returning the predicted probabilities for POST request containing a single .txt-document. The predicted document type probabilities would then be rendered by the frontend user-interface.

5. SYSTEM IMPLEMENTATION

The following describes the results of the problem-centered iteration of the DSRM process ultimately resulting in an artifact creation and implementation.

5.1 Problem identification and motivation

During the pre-project meetings with the client, it was defined that their use case would entail archiving large amounts of digital or digitized data. A single archive would consist of a single project's material. The use case would also require grouping varying document types in a single folder. However, the existing digital service platform, developed initially with another client's needs in mind, assigns document metadata, such as the document type (see 1.3 Objectives and restrictions), based on the folder they're placed in. Thus the already implemented solution, where the folder would govern some metadata of its documents, could not be reused. Yet another problem was, that although the client would be in charge of uploading the documents to the system, the documents themselves would be supplied not only by the client, but also by third parties. Without an existing unified process for setting document metadata, documents dating back several years and the users' general ignorance towards creating metadata (Crystal and Land, 2003), it was deduced that most if not all documents would contain gaps and errors in their respective metadata. Based on the requirement and use case discussions with the client the following problems were identified:

1. The current iteration of the digital service platform assigns document metadata based on its location in the hierarchy
2. A new use case requires the digital service platform to store varying types of documents in same location
3. The documents to be uploaded will most likely have gaps in metadata due to their age, lack of a unified process to create metadata or general ignorance.

5.2 Objectives for the solution

The objectives for a proof-of-concept project, acting as the solution, were derived from the identified problems in an agreement with the client. Document type was prioritized as the only metadata property to be deduced with the help of machine learning due to its importance and the multi-constrained nature of the project. It was agreed that some 18 000 documents, already created by the client with a document type assigned, could be used to train the machine learning system a selection of ten most common document types. A taught model would then be tested against a set of documents which desired type would be defined by the client. Based on the recognized requirements the following objectives were defined:

1. A machine learning system is to be developed as a proof of concept
2. The system should be taught the ten most common document types by using approximately 18 000 pre-classified documents created by the client
3. A taught system should be able to identify documents based on their textual content
4. The taught system would be measured against a real world project's documents created by third parties. These documents would be pre-classified by the client
5. The ML system should be integratable into the existing digital service platform
6. With the system integrated, the digital service platform should be able to suggest the most probable document types for the user

5.3 Design and development

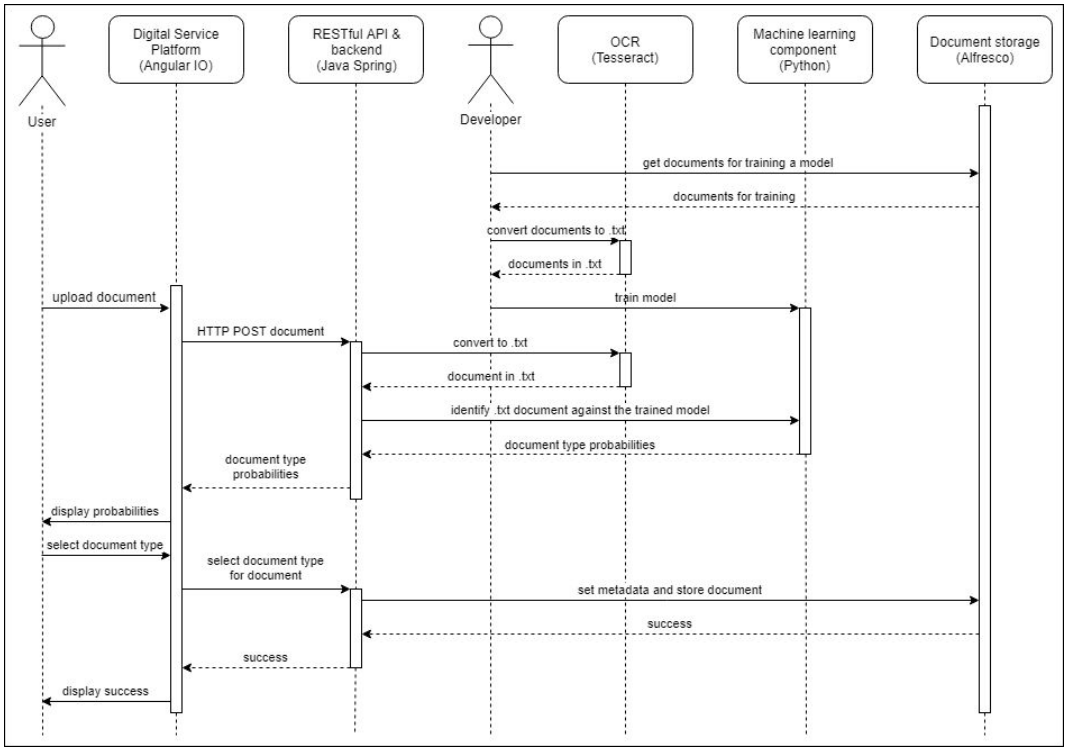

FIGURE 10. Sequence Diagram of the implementation

During the initial design phase a sequence diagram shown in Figure 10 was created. The diagram aims to give a better overview of what was to be done by modeling the high-level interactions between the user and sub-systems. The diagram features two actors; a user and a developer. The user depicts an end user willing to use the machine learning system to propose a document type for a document. The developer is a person capable of, and responsible for, retraining the machine learning models. Besides the actors, the diagram also captures five objects. Three of these, the digital service platform UI, a RESTful API and its backend and the Alfresco document management software already exist and are interconnected to form a digital service platform. The first of the two new objects was optical character recognition software `Tesseract`. It was to be used as a third party software to convert documents into plain text files. The other new object was the machine learning system itself.

Documents used for training the models would represent the ten most popular categories in Alfresco document management software. The popularity was measured in total number of documents of a given type. The extracted documents were known to be created by the client and any documents labeled as classified or containing classified information were discarded. `Tesseract` was then used to convert the extracted documents into plain text documents that are more maneuverable with text preprocessing tools. This resulted in a total of 18 313 plain text documents available for training the machine learning models. Later on the documents shorter than 64 characters would get eliminated because they were observed to contain little to none actual textual data. Because the classifying was to be done based on the documents textual content, such documents wouldn't be of any help and could be discarded. The final number of plain text documents used for training would total 13 709. At this point it was observed that some of the content in documents had been incorrectly converted by `Tesseract` resulting in malformed words or paragraphs or inconsistent line breaks. The grand majority, however, was converted successfully to plain text format.

`NLTK` (Bird, Loper and Klein, 2009) and `scikit-learn` (Pedregosa et al, 2011) Python libraries were chosen for text preprocessing due to their established reputation (nltk.org, 2019; Scikit-learn.org, 2019) in the field. A high level framework for `TensorFlow` backend called `Keras` was used in creating artificial neural networks with the help of `NumPy` which is a library for scientific computing in Python. `Keras` was chosen due to its ease of use compared to `TensorFlow` itself (Keras.io, 2019). Data vectorization was done with the robust, efficient and hassle-free (Řehůřek and Sojka, 2010; Řehůřek, 2019) `gensim` and `scikit-learn`. A boosted decision tree model, employing the vectorized data, was created with the renowned and award winning `XGBoost` (American Statistical Association, 2016; Linear Accelerator Laboratory, 2015).

Besides the listed machine learning libraries, libraries such as `pandas` and `Matplotlib` were used to help visualize the data.

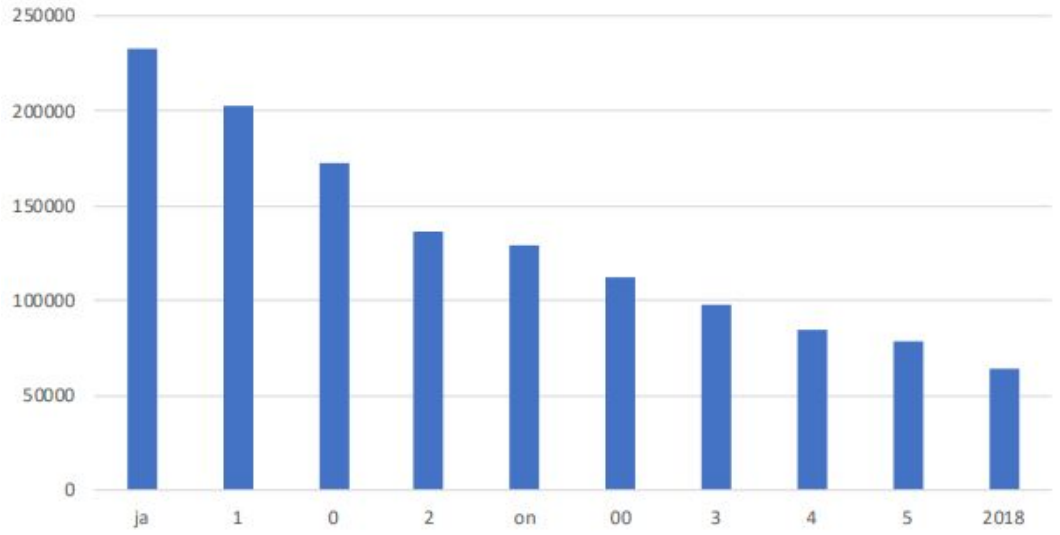

FIGURE 11. The most common words in the documents used for training

The descriptive analysis of the tokenized material revealed that the documents themselves contain a lot of varying series of numbers (see figure 11), different dates and e-mail addresses. In total there were 10 626 001 words of which 427 397 were unique.

The text was preprocessed by first conducting series of named entity recognition operations (see Chapter 3.3 Natural Language Processing) with the help of regular expressions. This was done to convert numbers, dates and e-mail addresses into named entities that could be indexed under a single type in the corpus. This would help different models associate a specific document type with a specific range of named entities. As a result of NER there were 10 169 576 words left of which 362 410 unique.

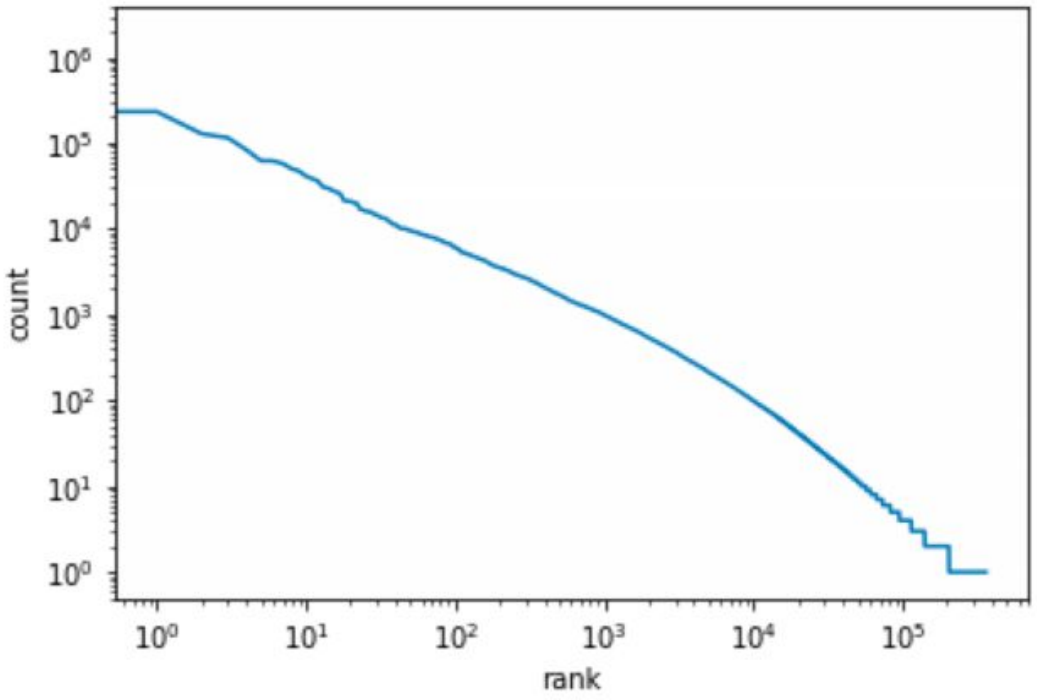

Figure 12. Word frequency plot

Further text examination revealed that the words frequency plot of the corpus, shown in figure 12, can be perceived as consistent with Zipf distribution derived from Zipf's law. It asserts that the frequency *f* of certain event, for example appearance of a word in text, is inversely proportional to their rank *r* (Encyclopedia Britannica, 2019). This meant that some words in the corpus were so frequent or so rare, that using them in training the models could be unnecessary, as they would do very little to help distinguish one type from another. Two different machine learning models were picked to analyze the corpus: boosted decision trees with two different vectorization methods and an artificial neural network.

To train a machine learning classifier, the textual tokens would need to be vectorized. The first vectorizer used for training a boosted decision tree classifier was a term frequency–inverse document frequency (TF-IDF) vectorizer. This was done by configuring `scikit-learn`'s TfidfVectorizer to vectorize tokenized

documents so that the tokens appearing in over 90 % of the documents as well as tokens appearing in less than 10 documents would be ignored leaving a total of 49 658 tokens to form the document vectors. The elimination of the most common and the rarest tokens would also act as a counter against the Zipf distributed corpus. The generated vectors could then be used to train an `XGBoost` library's XGBClassifier, a gradient boosted decision tree classifier, by using it's default of 100 decision trees. A trained classifier would also be evaluated with K-fold cross-validation where *k=5*. The K-fold validation meant that the training data would be split into 5 parts for 5 training iterations. Each iteration would use 4 parts for training and 1 part for testing. Each part would be used once to validate the training.

```
array([ 0.41794   ,  2.305189  , -1.8238662 ,  5.302312  , -3.5667315 ,
        2.9915895 ,  1.4356898 , -3.8480918 , -2.2624848 ,  1.4859542 ,
       -0.21133702,  0.2519051 , -2.1845684 ,  1.6107334 ,  2.299121  ,
       -0.28441414,  2.5029914 ,  1.2702643 ,  1.0949966 , -1.2892035 ,
        0.35185272,  2.799898  , -1.4071351 , -2.1187103 ,  3.4571905 ,
       -2.5637288 , -1.3878461 , -0.6832132 ,  0.20392504,  0.77167684,
       -0.5116474 , -7.2716007 , -1.1985132 ,  1.0320088 ,  0.68873554,
        0.8434052 , -0.9482002 ,  2.4912655 , -0.27645916, -1.471682  ,
       -0.6435345 , -2.57692   , -0.3136316 ,  0.52783006,  0.7214985 ,
        1.0310593 , -2.4275494 ,  1.5189041 ,  3.471925  ,  0.67427474,
        0.8827086 , -0.50609756,  0.65401864,  1.226068  , -0.72734535,
        0.28147936,  2.7954972 , -0.8549331 , -3.596929  , -1.791685  ,
        2.3363926 ,  1.1367042 , -2.5959752 ,  4.442723  ,  3.9380724 ,
       -2.045173  , -0.6517569 ,  1.311307  ,  4.70071   ,  4.887617  ,
       -0.11940282,  4.770954  , -2.9572108 ,  6.5445604 ,  0.42560363,
       -3.0336235 ,  3.9331143 ,  1.4560565 , -3.9139352 , -2.5177846 ,
       -0.25612873,  0.40811476,  0.55310476,  1.2956051 ,  0.12733956,
        1.374589  ,  0.5906013 , -6.4766393 , -0.12530224, -1.6744878 ,
       -1.8159788 , -0.8722993 , -0.93822855,  0.96714646,  0.04189672,
       -2.885344  , -0.3054586 ,  2.3356605 ,  4.163703  , -1.732836  ],
      dtype=float32)
```

FIGURE 13. A doc2vec vector representing a document

The other vectorizer that was chosen was the `gensim`'s implementation of doc2vec (see Chapter 3.3 Natural Language Processing). The vectorizer was configured to generate a Distributed Memory Model of Paragraph Vectors (PV-DM) as proposed by Le and Mikolov (2014). An unsupervised neural network would calculate a vector representation for each type appearing more

than twice in the corpus as well as a vector representation of a paragraph token for each document. The rate of appearance would act as a counter against the Zipf distribution. The paragraph token describes the document's type. After vectorization a single document's type vectors and it's paragraph vector would be concatenated to represent the document. This generated a single vector with a length of 100 such as the one illustrated in Figure 13. A total of 13 709 vectors similar to it were used to train an XGBClassifier that was similar to the one trained with vectors generated by TF-IDF vectorization.

An artificial neural network's applicability for the task was also tested due to their reported popularity and versatility (WIPO, 2019). To do this, the `Keras`' Sequential model was used to create a linear stack of layers (Keras.io, 2019) visualized in tables 3 and 4. The Sequential model allows for creating a simple feed forward neural network layer-by-layer. The tables' layer column indicates the type of the layer used. The output shape column describes the shape of the tensor, a multi-dimensional array of elements, that the layer outputs. The number of parameters column indicates the number of parameters handled by that specific layer that is calcuteable from its inputs.

TABLE 3. Word based artificial neural network layers. The "None" on every row in the table's Output shape column indicates that the batch size or total amount of documents is irrelevant.

Layer type	Output shape	Number of parameters
Embedding layer	None, 1500, 160	4 800 000
Pooling layer	None, 160	0
Dense layer	None, 200	32 200
Dense layer	None, 10	2010

An initial Embedding layer is given the 30 000 integer encoded tokenized words and the layer outputs 1500 dense vectors of 160 dimensions. 1500 is equal to a chosen length of a document's integer encoded tokens that should be taken into account. `Keras` recommends vectors of equal length to be used for more efficient matrix operations. This meant that the documents with less than 1500 tokens were padded with neutral data and the documents exceeding 1500 documents were truncated. 160 represents the embedding dimension, the length of the vector each integer encoded token would be mapped to. The number of parameters handled by the embedding layer is equal to the amount of input data multiplied by the given embedding dimension. A pooling layer is used to simplify the embedding layer's output matrix by taking only the maximum vector into account to prevent overfitting and to enhance the contrast between features. The last two layers are a pair of fully, densely connected neural network layers. The final dense layer outputs the probability distribution for the ten document types by using a softmax activation function.

The softmax activation is useful in multi-class learning where a sample belongs to one of many available classes as it's output range spans from 0 to 1, and the sum of all the probabilities will be equal to 1. When applied in multi-class learning, it's output vector contains probabilities for each class with the most likely class or classes having the highest probabilities. The probability vector is formed by computing a normalized exponential function of all input values of the layer.

After observing the preliminary results of the first three classifiers it was decided to partially iterate back to Design & Development phase of the DSRM process to see if a character token based artificial neural network would be more accurate in classifying the documents. This was done by creating an almost identical neural network to the one created for word tokens as Table 4 illustrates. A convolutional layer was appended and the documents' first 10 000 integer encoded characters were used as as the embedding layers input for this ANN. The convolutional

neural network (CNN) layer is used to derive the basic features from segments of the group of vectors it receives as input. CNN's output is a matrix where each column represents the weight of a feature detector. The trained character based ANN model was evaluated against the same batch of documents as the rest of the models and the results were documented (see table 4 in 5.4 Demonstration and Evaluation).

TABLE 4. Character based artificial neural network layers. The "None" on every row in the table's Output shape column indicates that the batch size or total amount of documents is irrelevant.

Layer type	Output shape	Number of parameters
Embedding layer	None, 10000, 160	108 800
Convolutional layer	None, 9996, 128	102 528
Pooling layer	None, 128	0
Dense layer	None, 200	25 800
Dense layer	None, 10	2010

After training the models it was necessary to be able to showcase the machine learning in practice. Because the client was already familiar with the digital service platform graphical user interface (GUI), the showcase was decided to be integrated into the existing GUI. To accomplish this, the trained classifiers would need to be in reach of the user. As illustrated in Figure 10, the user only interacts with the digital service platform's user interface. However, because the GUI is merely a representation of the system's state as reported by the backend, it was decided that the backend would be responsible for requesting the machine learning system to identify a document. The user would single out the document to be identified via a RESTful API request containing the document's ID.

A small web server was set up with the help of Python's `Tornado` library. As the web server is initialized, a chosen model is also loaded to be able to immediately handle any incoming requests. It was observed that the load caused by the running web server with a single model loaded, was not significantly taxing for the system. The running web server exposes an API accepting POST requests containing a plain text document, which then undergoes the same text preprocessing operations as the material used for training. After classifying the document by its contents, the API returns a JSON object containing probabilities for all of the trained document types ready to be presented in the GUI.

5.4 Demonstration and Evaluation

All of the classifiers proved their worth in training and K-fold validation as illustrated in Table 5. The table shows the accuracy of each classifier in three different phases of evaluation.

TABLE 5. Validation results

Classifier	Training accuracy	K-fold accuracy	Evaluation accuracy
TF-IDF boosted decision tree	90 %	84 %	42 %
Doc2vec boosted decision tree	99 %	87 %	20 %
Artificial Neural Network (word tokens)	99 %	91 %	44 %
Artificial Neural Network (character tokens)	91 %	82 %	44 %

All of the accuracies depicted in Table 5 are generated by the tools provided by the machine learning libraries themselves, but any of the accuracies could also be verified manually by inputting documents and comparing the predicted document types against the documents' reported type. The Python libraries used are

equipped with tools to evaluate a trained classifier's capability as a part its training as well as after it. The `XGBoost`'s XGBClassifier enables examination of the classifier's accuracy with the help of its score-function. The function outputs the ratio of correctly labeled inputs and can be run at every training epoch allowing a developer to examine when the classifier training reaches the point where further training creates negligible advantage over a previous iteration. The artificial neural network generated by `Keras` also has its own evaluate-function that outputs the classifier's ratio of correctly labeled inputs and can also be run at every training epoch.

The training accuracy in Table 5 represents the ratio either of the aforementioned evaluation functions output when they were given the set of material that was used to train them as an input. The table's K-fold accuracy column represents the classifier's ratio of correctly labeled documents when given a quintile of training material that was not used for training as input parameter. The evaluation accuracy column represents the ratio of the 185 documents reserved for evaluation, for which a classifier has predicted the correct, client reported label.

The client supplied documents, that were used for validation, were converted with `Tesseract` from .pdf-documents into .txt-files, that could then be preprocessed in an identical manner to the documents used in training the models. Each document was analyzed by all four classifiers and the result was compared against the respective document's client reported type. The complete list of predicted document types can be found in Appendix A.

5.5 Communication

Communicating the results of the DSRM process was first done between the internal stakeholders because the classifier's poor performance against the 185 documents used for evaluation as presented in Table 5 (see Chapter 5.4 Demonstration and Evaluation). Because the classifiers had performed well

against the training material, the client was inquired of their process for labeling the data used for evaluation. This revealed that the individual documents had not been inspected, but the document type was assigned based only on their parent folder's name. After closer examination of the evaluation material with the client, it was deduced, that the reported evaluation accuracy in Table 5 was not an accurate representation of the classifier's capabilities. The actual accuracy was estimated to be somewhere between the K-fold accuracy and evaluation accuracy.

The client was demonstrated the graphical user interface solution for using the classifier. The demonstration was done by randomly selecting uploaded documents for the classifier to identify. During the demonstration a future development suggestion was raised, that the machine learning system could operate automatically in the background, assigning document types for documents that have their type predicted within a certain threshold. Ultimately the client was satisfied with the performance of the machine learning system, but it was noted that the system should still be treated as work in progress due to the proof of concept nature of the project and that more development iterations should be made before applying the system in a production environment.

6. DISCUSSION

The practical use case, presented by this thesis in Chapter 5, concurs with a number of previous studies advocating automated metadata extraction (Adefowoke, Sunday Adewale and Oluwole, 2009; Hui et al., 2003; Liddy et al., 2002; Yilmazel, Finneran and Liddy, 2004). The described implementation of the machine learning system in Chapter 5 answers the main research question (see Chapter 1.3 Objectives and Restrictions), confirming that machine learning is an applicable method for filling some document metadata. This thesis' use case has the machine learning system acting as guidance for user to fill the metadata. The system could be rigged to assign the most probable document type automatically if a certain threshold is exceeded. Thus machine learning could prove itself as a valuable metadata tool in cases where user authored metadata is incorrect or lacking (Crystal and Land, 2003). However, machine learning's applicability in deducing various document metadata types may be limited.

Employing machine learning does have limitations as Table 5 in Chapter 5.4 demonstrates. Any material used for any kind of evaluation has to be correctly labeled. This also applies for the labeled training data where the data has to be valid and if possible fully validated to ensure, that the model is trained only the desired features. It is possible that the training material, created by the client, differs so greatly from the evaluation material created by the client's contractors, that a classifier is rendered useless. This may indicate, that a uniform human conception of a certain document type can't always be taken for granted. However, a machine learning model can always be retrained with controversial documents as a part of the training material to improve the classifier. The interval for retraining a model should be optimized to prevent tipping of the scales towards a single author's perception of a type.

In a hindsight the asiakirja-document type, which translates as document, included in training material used, could be regarded as too general, thus blurring the boundaries between different types. A human would arguably classify a document merely as a document in cases where classification is difficult or impossible. Such documents may contain very little features of any known document type, or the document may share features of two or more types. The classifiers proposed in this thesis return the probability for each of the trained labels, and some documents were observed to return very mixed probabilities (see documents in Appendix A where ANN's certainty is less than 50 %). In a sense this mimics the uncertainty a human might have in classifying a document. In a more automated system a general type such as "document" could be reserved for documents with such mixed probabilities.

The Python libraries used for creating machine learning models, and the libraries used in a more supporting role, enable the developed machine learning system to be integrated into existing systems via a lightweight, local `Tornado` web server. The web server can be run on any host machine that supports Python. The machine learning libraries allow the trained models to be saved and thus exported and imported to be used in a desired environment or container. `Keras` allows the user to save a trained ANN model into a .h5-file that can be used by both Python and Java applications (Weber, 2018), streamlining the deployment of machine learning software. The ability to import a pretrained model in Java applications also enables integration of machine learning into Android based mobile platforms.

Due to this research focusing on the applicability of machine learning as a document metadata tool, further research and development could be done to optimize the various parameters in the system. For example, the parameters used in `Tesseract`'s pdf to text conversion could be tweaked so the document's correct contents are used when training a model or when classifying the document with a classifier. Further research on classifier parameters could be done to ensure

optimal operation of a classifier. These parameters range from maximum amount of types used to the amount and type of layers used in a neural network or the amount of trees used in a gradient boosted decision tree.

The use case presented in this thesis only required the machine learning system to be able to figure out the document type. The described implementation of the system is only applicable in problem domains where large amount of documents can be labeled according to a desired metadata property. The metadata property should be deductible from the document content, arguably ruling out the majority of the different metadata types presented in Table 2 (see Chapter 2). Having a respective model for all required classifiers isn't efficient, because the size of a single model can easily be over 50 MB with more complex models spanning to 100 MB and beyond. Deployment of multiple models may either hog system resources as all of the models are kept readily available or cause notable delays as the desired model needs to be loaded before classifying. A single model capable of deducing a multitude of properties may be very complex and equally demanding for the system due to its size.

7. CONCLUSION

This thesis examined the machine learning's applicability as a document metadata tool. A machine learning system was developed using a problem-centered iteration of Design Science Research Methodology process. The development process entailed creation of four different models for their respective classifiers to find a suitable classifier for the problem. The classifiers were first evaluated with the documents their models were trained with, then with a K-fold cross-validation of the documents used for training and finally with an evaluation set of documents provided and labeled by the client. The trained machine learning models were exported to be used later.

While the classifiers performed remarkably well against the training material, correctly labeling at least 82 % of the documents, none of them performed well against the material reserved for evaluation. Three of the four classifiers managed to label only 44 % of the evaluation material correctly and one classifier managed to label only 20 % of the material correctly. A faulty material labeling process was a major contributing factor for the performance gap. A minor contributing factor may be individual or organizational differences in different document type concepts. With these factors taken into account when examining the overall performance, machine learning appears as a viable option to deduce a document's type metadata from its contents.

The developed machine learning system was successfully deployed as a sub-system into an existing digital service platform with the help of a local web server. As machine learning tools for other programming languages become more readily available, the deployment of machine learning applications may be significantly streamlined.

The models used by a machine learning system can always be retrained as more correctly labeled material is accumulated. This effectively extends a machine learning system's capability to recognize distinguishing features between document types. To extend a machine learning system's capabilities in other metadata types, one would need to relabel the training material accordingly nad retrain the models. However, only some of the metadata types can be deduced from the document's contents. Having multiple models for various tasks or one significantly larger model capable of multiple tasks may cause performance issues.

REFERENCES

Act on Electronic Services and Communication in the Public Sector 24.1.2003/13. Finlex. [Accessed 19 Jan. 2019].
Available at: https://www.finlex.fi/en/laki/kaannokset/2003/20030013

Adefowoke Ojokoh, B., Sunday Adewale, O. and Oluwole Falaki, S. (2009). Automated document metadata extraction. *Journal of Information Science*, Vol. 35 No. 5, pp. 563-570.

American Statistical Association. (2016). *Past winners of the John M. Chambers Statistical Software Award*. [online] Available at: http://stat-computing.org/awards/jmc/winners.html [Accessed 3 Apr. 2019].

Bird, S.; Loper, E.; and Klein, E. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Buduma, N. and Locascio, N. (2017). *Fundamentals of Deep Learning*. 1st ed. Sebastopol, CA: O'Reilly Media, Inc., pp. 53.

Burger, S. (2018). *Introduction to Machine Learning with R*. 1st ed. Sebastopol, CA: O'Reilly Media, Inc., pp. 107.

Chio, C. and Freeman, D. (2018). *Machine Learning and Security*. 1st ed. Sebastopol, CA: O'Reilly Media, Inc., pp.40, 53

Churchill, C. (2019). Outsourcing Document Management. *INFONOMICS*, Vol. 24, No. 2, p.46.

Conway, D. and White, J. (2012). *Machine learning for hackers*. Beijing: O'Reilly, p.vii.

Craig, N. and Sommerville, J. (2006). "Information management systems on construction projects: Case reviews." *Records Management Journal*, Vol. 16, No. 3, pp. 131-148, DOI: 10.1108/09565690610713192.

Crystal, A. and Land, P. (2003). *Metadata and Search*. Global Corporate Circle DCMI 2003 Workshop. [online] Seattle, Washington, USA: Dublin Core Metadata Initiative. Available at: http://dublincore.org/groups/corporate/Seattle/ [Accessed 21 Jan. 2019].

Degerstedt, A. (2000): *Inventering och utvärdering av elektroniska dokumenthanteringssystem i byggprocessen*. M.Sc. Thesis, Royal Institute of Technology, Construction and management Economics, Stockholm.

Encyclopedia Britannica. (2019). *Zipf's law | probability*. [online] Available at: https://www.britannica.com/topic/Zipfs-law [Accessed 5 Apr. 2019].

Grech, S., Holburn, J., Kelly, D., Beastall, P., Matthews, O., Barlow, M., Barbaroux, M., Gabrielczyk, M., Winchcomb, T., Clemoes, J. and Videtta, N. (2018). AI: Understanding and Harnessing the Potential. In: *Mobile World Congress 2018*. Cambridge, UK: Cambridge Consultants, pp. 31, 35-38

Hammer, M., and Hershman, L. W. (2010). *Faster, cheaper, better: The 9 levers for transforming how work gets done*. Crown Business.

Hevner, A. R. (2007). The three cycle view of design science research. *Scandinavian Journal of Information Systems*, Vol. 19, No. 2 pp. 87.

Hevner, A. and Chatterjee, S. (2010). *Design Research in Information Systems*, volume 22 of *Integrated Series in Information Systems*. Springer US, Boston, MA.

Hevner, A.R. et al. 2004. Design science in information systems research. *MIS Q*. Vol. 28, No. 1, pp. 75–105.

Hjelt, M. and Björk, B. (2007). End user attitudes towards EDM use in construction project work - a case study. *American Society for Civil Engineering Journal of Computing in Civil Engineering*, Vol 21, No 4, pp. 289-300

Hui Han, Giles, C., Manavoglu, E., Hongyuan, Z., Zhenyue, Z. and Fox, E. (n.d.). Automatic document metadata extraction using support vector machines. *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*

Hynninen, H., Jäske, P. and Tiili, M. (2015). *Sähköisen asioinnin lisäämisen tavoitteiden toteutuminen ja taloudelliset vaikutukset*. [online] Helsinki: Helsingin kaupunki, Tarkastusvirasto, p. 6, 7, 29, 32. Available at: https://www.arviointikertomus.fi/sites/default/files/pdf/article-memo/2016/arviointimuistio_sahkoinen_asiointi.pdf [Accessed 19 Jan. 2019].

Jervis, M., Masoodian, M., Evaluation of an Integrated Paper and Digital Document Management System.. *13th International Conference on Human-Computer Interaction (INTERACT)*, Sep 2011, Lisbon, Portugal. Springer, Lecture Notes in Computer Science, LNCS-6948 (Part III), pp.100-116, 2011,
Human-Computer Interaction – INTERACT 2011.
<10.1007/978-3-642-23765-2_8>. <hal-01591821>

Jones, S. (2012). eGovernment Document Management System: A case analysis of risk and reward. *International Journal of Information Management*, Vol. 32, No. 4, pp. 396-400.

Keras.io. (2019). *Why use Keras - Keras Documentation*. [online] Available at: https://keras.io/why-use-keras/ [Accessed 3 Apr. 2019].

Keras.io. (2019). *Guide to the Sequential model- Keras Documentation*. [online] Available at: https://keras.io/getting-started/sequential-model-guide/ [Accessed 3 Apr. 2019].

Kirk, M. (2017). *Thoughtful Machine Learning with Python*. 1st ed. Sebastopol, CA: O'Reilly Media, Inc., pp.111-117, 193, 197 - 199.

Knutas, A. (2016). Increasing Beneficial Interactions in a Computer-Supported Collaborative Environment. *Acta Universitatis Lappeenrantaensis*, [online] 718. Available at: http://urn.fi/URN:ISBN:978-952-335-007-6 [Accessed 25 Jan. 2019].

Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In: *International Conference on Machine Learning*. [online] Mountain View, CA: Google Inc., pp.1-8. Available at: https://ai.google/research/pubs/pub44930 [Accessed 19 Mar. 2019].

Liddy, E.D., Sutton, S., Allen, E., Harwell, S., Corieri, S. and Yilmazel, O, Automatic metadata generation and evaluation, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Tampere, Finland, 2002) pp. 401–402

Linear Accelerator Laboratory. (2015). *HEP meets ML award*. [online] Available at: https://higgsml.lal.in2p3.fr/prizes-and-award/award/ [Accessed 3 Apr. 2019].

Löwnertz, K. (1998): *Change and Exchange – Electronic document management in building design*. Licentiate Thesis, Royal Institute of Technology, Construction and management Economics, Stockholm, Sweden.

Ma, Z., Lu, N., and Wu, S. (2011). "Identification and representation of information resources for construction firms." *Advanced Engineering Informatics*, Vol. 25, No. 4, pp. 612-624, DOI: 10.1016/j.aei.2011. 08.008.

Manning, C., Raghavan, P. and Schütze, H. (2008). *Introduction to information retrieval*. 1st ed. New York: Cambridge University Press, p.22.

Merriam-webster.com. *Definition of METADATA*. [online] Available at: https://www.merriam-webster.com/dictionary/metadata [Accessed 30 Dec. 2018].

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In: *International Conference on Learning Representations*. [online] Mountain View, CA: Google Inc., pp.1-11. Available at: https://ai.google/research/pubs/pub41224 [Accessed 19 Mar. 2019].

The Ministry of Finance. (2019). *Digital services*. [online] Available at: https://vm.fi/en/digital-services [Accessed 14 Jan. 2019].

Moor, J. (2006). The Dartmouth College Artificial Intelligence Conference: The Next Fifty years, *AI Magazine*, Vol 27, No., 4, Pp. 87-9, 2006

Morgan, P. (2018). *Machine learning is changing the rules*. Sebastopol, CA: O'Reilly Media, p.2, 4, 43.

Nikkilä, M. (2017). *Sähköinen asiointi – onko ihan pakko?*. [online] The Ministry of Finance. Available at: https://vm.fi/artikkeli/-/asset_publisher/sahkoinen-asiointi-onko-ihan-pakko- [Accessed 19 Jan. 2019].

Nilsson, N. J. (2009). *The Quest for Artificial Intelligence*. Cambridge: Cambridge Univ. Press, pp.73, 141-154, 495-532. Web version available at: https://ai.stanford.edu/~nilsson/QAI/qai.pdf [Accessed 7 Mar 2019]

Nltk.org. (2019). *Natural Language Toolkit Documentation*. [online] Available at: https://www.nltk.org/ [Accessed 3 Apr. 2019].

O'Brien, W (2000): Implementation Issues in Project Web-Sites: A Practitioner's Viewpoint. *ASCE Journal of Management in Engineering*, Vol. 16, No.3, pp. 34-39.

Osinga, D. (2018). *Deep Learning Cookbook - Practical Recipes to Get Started Quickly*. 1st ed. Sebastopol, CA: O'Reilly, p.viii.

Pajukoski, M. (2004). *Sähköinen asiointi sosiaali- ja terveydenhuollossa*. Helsinki: Stakes, p.28.

Pathirage, C. P., Amaratunga, D. G., and Haigh, R. P. (2007). "Tacit knowledge and organisational performance: Construction industry perspective." *Journal of Knowledge Management*, Vol. 11, No. 1, pp. 115-126, DOI: 10.1108/13673270710728277.

Patterson, J. and Gibson, A. (2017). *Deep learning A Practitioner's Approach*. 2nd ed. Sebastopol, CA: O'Reilly Media.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, Vol. 24, No 3, pp. 45–77.

Pustejovsky, J. and Stubbs, A. (2012). *Natural Language Annotation for Machine Learning*. 1st ed. Sebastopol, CA: O'Reilly Media, Inc., p.IX, 4-5, 94-103, 140-167

Rajaraman, A., Ullman, J. and Leskovec, J. (2014). *Mining of Massive Datasets*. 2nd ed. Cambridge, United Kingdom: Cambridge University Press, pp.1-17.

Řehůřek, R. (2019). *gensim: topic modelling for humans*. [online] Gensim homepage. Available at: https://radimrehurek.com/gensim/about.html [Accessed 3 Apr. 2019].

Řehůřek, R. and Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*. Valletta, Malta: University of Malta, 2010. p. 46--50, 5 pp. ISBN 2-9517408-6-7.

Riley, J. (2004). *Understanding metadata*. Bethesda, Md.: National Information Standards Organization (NISO), p.1.

Riley, J. (2017). *Understanding metadata - What is metadata and what is it for?*. Baltimore, MD: National Information Standards Organization (NISO), pp.1, 7.

Russell, S. and Norvig, P. (2010). *Artificial intelligence - A Modern Approach*. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, pp. 17, 27, 30, 860-887

Sabour, S., Frosst, N., Hinton, G.E. (2017). *Dynamic Routing Between Capsules*. Available at: https://arxiv.org/abs/1710.09829 [Accessed 22 Mar. 2019].

Scikit-learn.org. (2019). *scikit-learn documentation*. [online] Available at: https://scikit-learn.org/stable/ [Accessed 3 Apr. 2019].

Sellen, A.J., Harper, R.H.R.: The Myth of the Paperless Office. MIT Press, Cambridge, MA, USA (2003)

Shin, Y. (2015). *Designing a system prototype for construction document management using automated tagging and visualization*, MSc Thesis, Seoul National University, Seoul, Korea.

Soibelman, L., Wu, J., Caldas, C., Brilakis, I., and Lin, K. Y. (2008). "Management and analysis of unstructured construction data types." *Advanced Engineering Informatics*, Vol. 22, No. 1, pp. 15-27, DOI: 10.1016/j.aei.2007.08.011.

Solomonoff, R.J. (1985). The Time Scale of Artificial Intelligence; Reflections on Social Effects, *Human Systems Management*, Vol 5 1985, Pp 149-153

Sprehe, J. T. (2004) A Framework for EDMS/ERMS Integration. *Information Management Journal*, Vol. 38, No. 6, pp. 54-62

Sulankivi, K., Lakka, A. and Luedke, M. (2002): *Projektin hallinta sähköisen tiedonsiirron ympäristössä*. VTT Publications, Espoo, Finland.

Tornadoweb.org. (2019). *Tornado Web Server — Tornado 6.0.2 documentation*. [online] Available at: https://www.tornadoweb.org/en/stable/ [Accessed 6 Apr. 2019].

Visma.fi. (2019). *Visma Homepage*. [online] Available at: https://www.visma.fi/ [Accessed 24 Jan. 2019].

Weber, B. (2018). *Deploying Keras Deep Learning Models with Java*. [online] Towards Data Science. Available at: https://towardsdatascience.com/deploying-keras-deep-learning-models-with-java-62d80464f34a [Accessed 13 Apr. 2019].

Webster, M. (2015). *Addressing the Document Disconnect*. [online] Framingham, MA: International Data Corporation. Available at: https://acrobat.adobe.com/content/dam/doc-cloud/en/pdfs/idc-adobe-document-disconnect-whitepaper-global-ie-final.pdf [Accessed 14 Feb. 2019].

Wolf, K. (2018). *Where EDMS Fails Data Integrity Pitfalls To Avoid In Metadata For Life Science Products*. [online] Outsourcedpharma.com. Available at: https://www.outsourcedpharma.com/doc/where-edms-fails-data-integrity-pitfalls-to-avoid-in-metadata-for-life-science-products-0001 [Accessed 22 Feb. 2019].

Yilmazel, O., Finneran, C.M., and Liddy, E.D. MetaExtract: an NLP system to automatically assign metadata, *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries* (Tuscan, AZ, USA, 2004) pp. 241–242

WIPO (2019). *WIPO Technology Trends – Artificial Intelligence*. 1st ed. Geneva, Switzerland: World Intellectual Property Organization (WIPO), p.1. Available at: https://www.wipo.int/tech_trends/en/artificial_intelligence/ [Accessed 24 Feb. 2019].

APPENDICES

APPENDIX A. Document types according to different machine learning models. Probability column represents the word based ANN classifier's

	Document type according to classifier			
Reported type	**ANN1 Word based**	**ANN2 Character based**	**Random forest TF-IDF boosted**	**ANN1 Probability**
asiakirja	raportti	raportti	raportti	53,8 %
asiakirja	raportti	raportti	raportti	43,7 %
asiakirja	raportti	raportti	raportti	52,8 %
asiakirja	raportti	raportti	raportti	50,2 %
asiakirja	raportti	raportti	dokumentaatio	98,1 %
asiakirja	raportti	raportti	dokumentaatio	94,8 %
asiakirja	raportti	raportti	dokumentaatio	97,4 %
asiakirja	raportti	raportti	dokumentaatio	96,3 %
asiakirja	raportti	hakemus	dokumentaatio	64,0 %
asiakirja	ilmoitus	dokumentaatio	hakemus	39,1 %
asiakirja	suunnitelma	tilaus	dokumentaatio	73,8 %
asiakirja	ilmoitus	ilmoitus	dokumentaatio	72,8 %
asiakirja	hakemus	ilmoitus	dokumentaatio	91,1 %
asiakirja	hakemus	ilmoitus	dokumentaatio	62,5 %
asiakirja	hakemus	hakemus	dokumentaatio	68,1 %
asiakirja	ilmoitus	dokumentaatio	dokumentaatio	63,8 %
asiakirja	hakemus	ilmoitus	dokumentaatio	80,1 %
asiakirja	hakemus	ilmoitus	dokumentaatio	71,1 %
asiakirja	asiakirja	hakemus	dokumentaatio	82,6 %
asiakirja	asiakirja	hakemus	dokumentaatio	98,1 %
asiakirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
asiakirja	pöytäkirja	dokumentaatio	pöytäkirja	46,9 %
asiakirja	hakemus	raportti	hakemus	66,8 %
dokumentaatio	asiakirja	ilmoitus	hakemus	71,0 %

dokumentaatio	hakemus	ilmoitus	dokumentaatio	31,4 %
dokumentaatio	asiakirja	raportti	hakemus	48,3 %
dokumentaatio	ilmoitus	raportti	hakemus	53,2 %
dokumentaatio	asiakirja	dokumentaatio	dokumentaatio	61,1 %
dokumentaatio	asiakirja	dokumentaatio	dokumentaatio	46,7 %
dokumentaatio	pöytäkirja	muistio	hakemus	67,0 %
dokumentaatio	pöytäkirja	ilmoitus	hakemus	58,1 %
dokumentaatio	dokumentaatio	dokumentaatio	hakemus	66,4 %
dokumentaatio	dokumentaatio	hakemus	hakemus	88,7 %
dokumentaatio	raportti	hakemus	dokumentaatio	45,0 %
dokumentaatio	hakemus	raportti	hakemus	32,7 %
dokumentaatio	dokumentaatio	hakemus	dokumentaatio	97,0 %
dokumentaatio	ilmoitus	ilmoitus	dokumentaatio	70,5 %
dokumentaatio	dokumentaatio	hakemus	dokumentaatio	38,3 %
dokumentaatio	hakemus	raportti	dokumentaatio	50,6 %
dokumentaatio	ilmoitus	ilmoitus	ilmoitus	78,1 %
dokumentaatio	hakemus	pöytäkirja	dokumentaatio	49,3 %
dokumentaatio	dokumentaatio	ilmoitus	dokumentaatio	36,7 %
dokumentaatio	raportti	ilmoitus	dokumentaatio	79,8 %
dokumentaatio	dokumentaatio	pöytäkirja	dokumentaatio	73,3 %
dokumentaatio	hakemus	hakemus	dokumentaatio	35,8 %
dokumentaatio	hakemus	tilaus	dokumentaatio	48,1 %
dokumentaatio	pöytäkirja	pöytäkirja	hakemus	94,5 %
dokumentaatio	pöytäkirja	dokumentaatio	dokumentaatio	32,7 %
dokumentaatio	hakemus	dokumentaatio	dokumentaatio	60,4 %
dokumentaatio	suunnitelma	suunnitelma	hakemus	60,1 %
dokumentaatio	dokumentaatio	hakemus	dokumentaatio	55,5 %
dokumentaatio	hakemus	hakemus	dokumentaatio	37,2 %
dokumentaatio	suunnitelma	hakemus	dokumentaatio	64,2 %
dokumentaatio	dokumentaatio	dokumentaatio	dokumentaatio	60,4 %
dokumentaatio	hakemus	dokumentaatio	dokumentaatio	64,0 %
dokumentaatio	ilmoitus	dokumentaatio	dokumentaatio	64,8 %
dokumentaatio	asiakirja	raportti	dokumentaatio	32,8 %

dokumentaatio	ilmoitus	dokumentaatio	ilmoitus	50,3 %
dokumentaatio	hakemus	muistio	hakemus	31,0 %
dokumentaatio	asiakirja	muistio	asiakirja	83,1 %
dokumentaatio	dokumentaatio	dokumentaatio	dokumentaatio	69,4 %
hakemus	raportti	hakemus	hakemus	88,3 %
hakemus	hakemus	hakemus	hakemus	69,9 %
hakemus	hakemus	hakemus	hakemus	98,1 %
hakemus	hakemus	hakemus	hakemus	99,9 %
hakemus	hakemus	hakemus	hakemus	99,7 %
hakemus	hakemus	hakemus	hakemus	99,8 %
hakemus	hakemus	hakemus	hakemus	99,9 %
hakemus	hakemus	hakemus	hakemus	100,0 %
hakemus	hakemus	hakemus	hakemus	99,2 %
hakemus	hakemus	hakemus	hakemus	98,9 %
hakemus	hakemus	hakemus	hakemus	97,9 %
hakemus	hakemus	hakemus	hakemus	89,8 %
hakemus	hakemus	hakemus	hakemus	74,6 %
hakemus	hakemus	hakemus	hakemus	99,0 %
hakemus	hakemus	hakemus	hakemus	94,1 %
hakemus	hakemus	hakemus	hakemus	66,9 %
hakemus	hakemus	hakemus	hakemus	54,3 %
hakemus	hakemus	hakemus	hakemus	98,6 %
hakemus	hakemus	hakemus	hakemus	96,7 %
hakemus	hakemus	hakemus	hakemus	98,5 %
hakemus	hakemus	hakemus	hakemus	95,4 %
hakemus	hakemus	hakemus	hakemus	97,4 %
hakemus	hakemus	hakemus	hakemus	96,7 %
hakemus	hakemus	hakemus	hakemus	99,9 %
hakemus	hakemus	hakemus	hakemus	85,1 %
hakemus	hakemus	hakemus	hakemus	98,6 %
hakemus	hakemus	päätös	hakemus	91,0 %
hakemus	hakemus	hakemus	hakemus	99,8 %
hakemus	raportti	päätös	hakemus	95,4 %

ilmoitus	hakemus	hakemus	dokumentaatio	49,6 %
ilmoitus	ilmoitus	hakemus	ilmoitus	96,4 %
ilmoitus	dokumentaatio	pöytäkirja	dokumentaatio	73,3 %
ilmoitus	dokumentaatio	dokumentaatio	dokumentaatio	31,0 %
ilmoitus	ilmoitus	ilmoitus	raportti	37,1 %
ilmoitus	ilmoitus	hakemus	hakemus	73,0 %
ilmoitus	hakemus	hakemus	ilmoitus	95,0 %
ilmoitus	raportti	ilmoitus	dokumentaatio	43,2 %
ilmoitus	ilmoitus	muistio	raportti	36,5 %
ilmoitus	ilmoitus	muistio	dokumentaatio	59,8 %
ilmoitus	ilmoitus	muistio	dokumentaatio	68,5 %
ilmoitus	dokumentaatio	dokumentaatio	dokumentaatio	48,7 %
ilmoitus	dokumentaatio	muistio	dokumentaatio	83,9 %
ilmoitus	dokumentaatio	pöytäkirja	raportti	79,3 %
muistio	päätös	dokumentaatio	dokumentaatio	27,1 %
muistio	pöytäkirja	muistio	muistio	50,9 %
muistio	muistio	muistio	muistio	53,6 %
muistio	muistio	pöytäkirja	muistio	84,9 %
muistio	pöytäkirja	pöytäkirja	muistio	84,7 %
muistio	raportti	pöytäkirja	ilmoitus	89,7 %
muistio	pöytäkirja	muistio	hakemus	67,0 %
muistio	pöytäkirja	muistio	hakemus	67,0 %
muistio	muistio	muistio	muistio	95,4 %
muistio	pöytäkirja	pöytäkirja	dokumentaatio	46,0 %
muistio	pöytäkirja	muistio	muistio	51,9 %
muistio	päätös	muistio	muistio	59,4 %
muistio	muistio	muistio	muistio	87,3 %
päätös	hakemus	hakemus	ilmoitus	98,3 %
päätös	päätös	päätös	päätös	99,2 %
pöytäkirja	pöytäkirja	muistio	muistio	50,9 %
pöytäkirja	muistio	muistio	muistio	53,6 %
pöytäkirja	muistio	pöytäkirja	muistio	84,9 %
pöytäkirja	pöytäkirja	pöytäkirja	muistio	84,7 %

pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	muistio	pöytäkirja	muistio	89,5 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	99,5 %
pöytäkirja	päätös	pöytäkirja	dokumentaatio	88,9 %
pöytäkirja	muistio	hakemus	muistio	90,1 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	98,7 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	98,7 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	99,9 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
pöytäkirja	pöytäkirja	pöytäkirja	pöytäkirja	100,0 %
raportti	raportti	hakemus	dokumentaatio	55,3 %
raportti	raportti	hakemus	dokumentaatio	45,0 %
raportti	dokumentaatio	dokumentaatio	dokumentaatio	48,7 %
raportti	dokumentaatio	muistio	dokumentaatio	83,9 %
raportti	ilmoitus	muistio	raportti	36,5 %
raportti	hakemus	dokumentaatio	dokumentaatio	60,4 %
raportti	muistio	hakemus	hakemus	28,7 %
raportti	raportti	raportti	raportti	100,0 %
raportti	raportti	raportti	raportti	100,0 %
suunnitelma	dokumentaatio	muistio	dokumentaatio	68,1 %
suunnitelma	ilmoitus	dokumentaatio	dokumentaatio	40,5 %
suunnitelma	raportti	muistio	dokumentaatio	55,6 %
suunnitelma	raportti	muistio	raportti	51,7 %
suunnitelma	dokumentaatio	dokumentaatio	dokumentaatio	98,2 %

suunnitelma	dokumentaatio	hakemus	dokumentaatio	64,0 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	95,7 %
suunnitelma	raportti	hakemus	dokumentaatio	89,3 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	85,1 %
suunnitelma	raportti	suunnitelma	dokumentaatio	41,4 %
suunnitelma	dokumentaatio	dokumentaatio	dokumentaatio	61,5 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	80,5 %
suunnitelma	raportti	hakemus	dokumentaatio	66,2 %
suunnitelma	pöytäkirja	pöytäkirja	hakemus	53,6 %
suunnitelma	pöytäkirja	pöytäkirja	raportti	61,4 %
suunnitelma	raportti	pöytäkirja	raportti	68,1 %
suunnitelma	asiakirja	muistio	raportti	65,6 %
suunnitelma	raportti	hakemus	raportti	75,7 %
suunnitelma	asiakirja	pöytäkirja	hakemus	57,5 %
suunnitelma	asiakirja	pöytäkirja	raportti	87,7 %
suunnitelma	ilmoitus	hakemus	suunnitelma	64,5 %
suunnitelma	pöytäkirja	pöytäkirja	raportti	49,2 %
suunnitelma	dokumentaatio	hakemus	suunnitelma	74,3 %
suunnitelma	suunnitelma	hakemus	dokumentaatio	51,9 %
suunnitelma	pöytäkirja	pöytäkirja	dokumentaatio	77,6 %
suunnitelma	dokumentaatio	dokumentaatio	dokumentaatio	98,2 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	85,1 %
suunnitelma	raportti	hakemus	dokumentaatio	85,9 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	95,7 %
suunnitelma	raportti	suunnitelma	dokumentaatio	41,4 %
suunnitelma	raportti	hakemus	dokumentaatio	89,3 %
suunnitelma	dokumentaatio	dokumentaatio	dokumentaatio	61,5 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	64,0 %
suunnitelma	dokumentaatio	hakemus	dokumentaatio	80,5%