# STATISTICAL NATURAL LANGUAGE PROCESSING METHODS FOR INTELLIGENT PROCESS AUTOMATION

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Philosophie
an der Ludwig-Maximilians-Universität München

vorgelegt von
Alena Moiseeva
aus Moskau, Russische Föderation

München 2019

Erstgutachter: Prof. Dr. Hinrich Schütze

1.Korreferent: Prof. Dr. Alexander M. Fraser
2.Korreferent: Prof. Dr. Rudolf Seiler

TAG DER MÜNDLICHEN PRÜFUNG: 19.05.2020

# ABSTRACT

Nowadays, digitization is transforming the way businesses work. Recently, Artificial Intelligence (AI) techniques became an essential part of the automation of business processes: In addition to cost advantages, these techniques offer fast processing times and higher customer satisfaction rates, thus ultimately increasing sales. One of the intelligent approaches for accelerating digital transformation in companies is the Robotic Process Automation (RPA). A RPA-system is a software tool that robotizes routine and time-consuming responsibilities such as email assessment, various calculations, or creation of documents and reports (Mohanty and Vyas, 2018). Its main objective is to organize a smart workflow and therethrough to assist employees by offering them more scope for cognitively demanding and engaging work.

Intelligent Process Automation (IPA) offers all these advantages as well; however, it goes beyond the RPA by adding AI components such as Machine- and Deep Learning techniques to conventional automation solutions. Previously, IPA approaches were primarily employed within the computer vision domain. However, in recent times, Natural Language Processing (NLP) became one of the potential applications for IPA as well due to its ability to understand and interpret human language. Usually, NLP methods are used to analyze large amounts of unstructured textual data and to respond to various inquiries. However, one of the central applications of NLP within the IPA domain – are conversational interfaces (e.g., chatbots, virtual agents) that are used to enable human-to-machine communication. Nowadays, conversational agents gain enormous demand due to their ability to support a large number of users simultaneously while communicating in a natural language. The implementation of a conversational agent comprises multiple stages and involves diverse types of NLP sub-tasks, starting with natural language understanding (e.g., intent recognition, named entity extraction) and going towards dialogue management (i.e., determining the next possible bots action) and response generation. Typical dialogue system for IPA purposes undertakes straightforward customer support requests (e.g., FAQs), allowing human workers to focus on more complicated inquiries.

In this thesis, we are addressing two potential Intelligent Process Automation (IPA) applications and employing statistical Natural Language Processing (NLP) methods for their implementation.

The first block of this thesis (Chapter 2 – Chapter 4) deals with the development of a *conversational agent* for IPA purposes within the e-learning domain. As already mentioned, chatbots are one of the central applications for the IPA domain since they can effectively perform time-consuming tasks while communicating in a natural lan-

guage. Within this thesis, we realized the IPA conversational bot that takes care of routine and time-consuming tasks regularly performed by human tutors of an online mathematical course. This bot is deployed in a real-world setting within the OMB+ mathematical platform. Conducting experiments for this part, we observed two possibilities to build the conversational agent in industrial settings – first, with purely rule-based methods, considering the missing training data and individual aspects of the target domain (i.e., e-learning). Second, we re-implemented two of the main system components (i.e., Natural Language Understanding (NLU) and Dialogue Manager (DM) units) using the current state-of-the-art deep-learning architecture (i.e., Bidirectional Encoder Representations from Transformers (BERT)) and investigated their performance and potential use as a part of a hybrid model (i.e., containing both rule-based and machine learning methods).

The second part of the thesis (Chapter 5 – Chapter 6) considers an IPA subproblem within the *predictive analytics* domain and addresses the task of scientific trend forecasting. Predictive analytics forecasts future outcomes based on historical and current data. Therefore, using the benefits of advanced analytics models, an organization can, for instance, reliably determine trends and emerging topics and then manipulate it while making significant business decisions (i.e., investments). In this work, we dealt with the trend detection task – specifically, we addressed the lack of publicly available benchmarks for evaluating trend detection algorithms. We assembled the benchmark for the detection of both scientific *trends* and *downtrends* (i.e., topics that become less frequent overtime). To the best of our knowledge, the task of downtrend detection has not been addressed before. The resulting benchmark is based on a collection of more than one million documents, which is among the largest that has been used for trend detection before, and therefore, offers a realistic setting for the development of trend detection algorithms.

# ZUSAMMENFASSUNG

Robotergesteuerte Prozessautomatisierung (RPA) ist eine Art von Software-Bots, die manuelle menschliche Tätigkeiten wie die Eingabe von Daten in das System, die Anmeldung in Benutzerkonten oder die Ausführung einfacher, aber sich wiederholender Arbeitsabläufe nachahmt (Mohanty and Vyas, 2018). Einer der Hauptvorteile und gleichzeitig Nachteil der RPA-bots ist jedoch deren Fähigkeit, die gestellte Aufgabe punktgenau zu erfüllen. Einerseits ist ein solches System in der Lage, die Aufgabe akkurat, sorgfältig und schnell auszuführen. Andererseits ist es sehr anfällig für Veränderungen in definierten Szenarien. Da der RPA-Bot für eine bestimmte Aufgabe konzipiert ist, ist es oft nicht möglich, ihn an andere Domänen oder sogar für einfache Änderungen in einem Arbeitsablauf anzupassen (Mohanty and Vyas, 2018). Diese Unfähigkeit, sich an veränderte Bedingungen anzupassen, führte zu einem weiteren Verbesserungsbereich für RPA-bots – den Intelligenten Prozessautomatisierungssystemen (IPA).

IPA-Bots kombinieren RPA mit Künstlicher Intelligenz (AI) und können komplexe und kognitiv anspruchsvollere Aufgaben erfüllen, die u.A. Schlussfolgerungen und natürliches Sprachverständnis erfordern. Diese Systeme übernehmen zeitaufwändige und routinemäßige Aufgaben, ermöglichen somit einen intelligenten Arbeitsablauf und befreien Fachkräfte für die Durchführung komplizierterer Aufgaben. Bisher wurden die IPA-Techniken hauptsächlich im Bereich der Bildverarbeitung eingesetzt. In der letzten Zeit wurde die natürliche Sprachverarbeitung (NLP) jedoch auch zu einem der potenziellen Anwendungen für IPA, und zwar aufgrund von der Fähigkeit, die menschliche Sprache zu interpretieren. NLP-Methoden werden eingesetzt, um große Mengen an Textdaten zu analysieren und auf verschiedene Anfragen zu reagieren. Auch wenn die verfügbaren Daten unstrukturiert sind oder kein vordefiniertes Format haben (z.B. E-Mails), oder wenn die in einem variablen Format vorliegen (z.B. Rechnungen, juristische Dokumente), dann werden ebenfalls die NLP Techniken angewendet, um die relevanten Informationen zu extrahieren, die dann zur Lösung verschiedener Probleme verwendet werden können.

NLP im Rahmen von IPA beschränkt sich jedoch nicht auf die Extraktion relevanter Daten aus Textdokumenten. Eine der zentralen Anwendungen von IPA sind Konversationsagenten, die zur Interaktion zwischen Mensch und Maschine eingesetzt werden. Konversationsagenten erfahren enorme Nachfrage, da sie in der Lage sind, eine große Anzahl von Benutzern gleichzeitig zu unterstützen, und dabei in einer natürlichen Sprache kommunizieren. Die Implementierung eines Chatsystems umfasst verschiedene Arten von NLP-Teilaufgaben, beginnend mit dem Verständnis der natürlichen Sprache (z.B. Absichtserkennung, Extraktion von Entitäten) über das Dialogmanage-

ment (z.B. Festlegung der nächstmöglichen Bot-Aktion) bis hin zur Response-Generierung. Ein typisches Dialogsystem für IPA-Zwecke übernimmt in der Regel unkomplizierte Kundendienstanfragen (z.B. Beantwortung von FAQs), so dass sich die Mitarbeiter auf komplexere Anfragen konzentrieren können.

Diese Dissertation umfasst zwei Bereiche, die durch das breitere Thema vereint sind, nämlich die Intelligente Prozessautomatisierung (IPA) unter Verwendung statistischer Methoden der natürlichen Sprachverarbeitung (NLP).

Der erste Block dieser Arbeit (Kapitel 2 – Kapitel 4) befasst sich mit der Impementierung eines *Konversationsagenten* für IPA-Zwecke innerhalb der E-Learning-Domäne. Wie bereits erwähnt, sind Chatbots eine der zentralen Anwendungen für die IPA-Domäne, da sie zeitaufwändige Aufgaben in einer natürlichen Sprache effektiv ausführen können. Der IPA-Kommunikationsbot, der in dieser Arbeit realisiert wurde, kümmert sich ebenfalls um routinemäßige und zeitaufwändige Aufgaben, die sonst von Tutoren in einem Online-Mathematikkurs in deutscher Sprache durchgeführt werden. Dieser Bot ist in der täglichen Anwendung innerhalb der mathematischen Plattform OMB+ eingesetzt. Bei der Durchführung von Experimenten beobachteten wir zwei Möglichkeiten, den Konversationsagenten im industriellen Umfeld zu entwickeln – zunächst mit rein regelbasierten Methoden, unter Bedingungen der fehlenden Trainingsdaten und besonderer Aspekte der Zieldomäne (d.h. E-Learning). Zweitens haben wir zwei der Hauptsystemkomponenten (Sprachverständnismodul, Dialog-Manager) mit dem derzeit fortschrittlichsten Deep Learning Algorithmus reimplementiert und die Performanz dieser Komponenten untersucht.

Der zweite Teil der Doktorarbeit (Kapitel 5 – Kapitel 6) betrachtet ein IPA-Problem innerhalb des *Vorhersageanalytik*-Bereichs. Vorhersageanalytik zielt darauf ab, Prognosen über zukünftige Ergebnisse auf der Grundlage von historischen und aktuellen Daten zu erstellen. Daher kann ein Unternehmen mit Hilfe der Vorhersagesysteme z.B. die Trends oder neu entstehende Themen zuverlässig bestimmen und diese Informationen dann bei wichtigen Geschäftsentscheidungen (z.B. Investitionen) einsetzen. In diesem Teil der Arbeit beschäftigen wir uns mit dem Teilproblem der Trendprognose – insbesondere mit dem Fehlen öffentlich zugänglicher Benchmarks für die Evaluierung von Trenderkennungsalgorithmen. Wir haben den Benchmark zusammengestellt und veröffentlicht, um sowohl *Trends* als auch *Abwärtstrends* zu erkennen. Nach unserem besten Wissen ist die Aufgabe der Abwärtstrenderkennung bisher nicht adressiert worden. Der resultierende Benchmark basiert auf einer Sammlung von mehr als einer Million Dokumente, der zu den größten gehört, die bisher für die Trenderkennung verwendet wurden, und somit einen realistischen Rahmen für die Entwicklung von Trenddetektionsalgorithmen bietet.

# ACKNOWLEDGEMENTS

This thesis was possible due to several people who contributed significantly to my work. I owe my gratitude to each of them for guidance, encouragement, and inspiration during my doctoral study.

First of all, I would like to thank my advisor, *Prof. Dr. Hinrich Schütze*, for his mentorship, support, and thoughtful guidance. My special gratitude goes for *Prof. Dr. Ruedi Seiler* and his fantastic team at Integral-Learning GmbH. I am thankful for your valuable input, interest in our collaborative work, and for your continuous support. I would also like to show gratitude to *Prof. Dr. Alexander Fraser* for co-advising, support and helpful feedback.

Finally, none of this would be possible without my family's support: there are no words to express the depth of gratitude I feel for them. My earnest appreciation goes to my husband, Dietrich: thank you for all the time you were by my side and for the encouragement you provided me through these years. I am very grateful to my parents for their understanding and backing at any moment. I appreciate the love of my grandparents: it is hard to realize how significant their care is for me. Besides that, I am also grateful to my sister Alexandra, elder brother Denis, and parents-in-law, for merely being always there for me.

# CONTENTS

List of Figures XI
List of Tables XII
Acronyms XIII
1 INTRODUCTION 1
1.1 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
I E-LEARNING CONVERSATIONAL ASSISTANT 5
2 FOUNDATIONS 7
2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.2 Conversational Assistants . . . . . . . . . . . . . . . . . 9
2.2.1 Taxonomy . . . . . . . . . . . . . . . . . . . . . 9
2.2.2 Conventional Architecture . . . . . . . . . . . . . 12
2.2.3 Existing Approaches . . . . . . . . . . . . . . . . 15
2.3 Target Domain & Task Definition . . . . . . . . . . . . . 18
2.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3 MULTIPURPOSE IPA VIA CONVERSATIONAL ASSISTANT 21
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 21
3.1.1 Outline and Contributions . . . . . . . . . . . . 22
3.2 Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.2.1 OMB+ Design . . . . . . . . . . . . . . . . . . . . 23
3.2.2 Preprocessing . . . . . . . . . . . . . . . . . . . . 23
3.2.3 Natural Language Understanding . . . . . . . . 25
3.2.4 Dialogue Manager . . . . . . . . . . . . . . . . . 26
3.2.5 Meta Policy . . . . . . . . . . . . . . . . . . . . . 30
3.2.6 Response Generation . . . . . . . . . . . . . . . . 32
3.3 Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . 37
3.3.1 Automated Evaluation . . . . . . . . . . . . . . . 37
3.3.2 Human Evaluation & Error Analysis . . . . . . 37
3.4 Structured Dialogue Acquisition . . . . . . . . . . . . . 39
3.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . 39
3.6 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . 40
4 DEEP LEARNING RE-IMPLEMENTATION OF UNITS 41
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 41
4.1.1 Outline and Contributions . . . . . . . . . . . . 42
4.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . 43
4.3 BERT: Bidirectional Encoder Representations from Transformers . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
4.4 Data & Descriptive Statistics . . . . . . . . . . . . . . . . 46
4.5 Named Entity Recognition . . . . . . . . . . . . . . . . . 48
4.5.1 Model Settings . . . . . . . . . . . . . . . . . . . 49
4.6 Next Action Prediction . . . . . . . . . . . . . . . . . . . 49
4.6.1 Model Settings . . . . . . . . . . . . . . . . . . . 49
4.7 Evaluation and Results . . . . . . . . . . . . . . . . . . . 50
4.8 Error Analysis: . . . . . . . . . . . . . . . . . . . . . . . . 58

4.9 Summary . . . . . . . . . . . . . . . . . . . . . . . . . 58
4.10 Future Work . . . . . . . . . . . . . . . . . . . . . . . . 59

II CLUSTERING-BASED TREND ANALYZER 61
5 FOUNDATIONS 63
5.1 Motivation and Challenges . . . . . . . . . . . . . . . . 63
5.2 Detection of Emerging Research Trends . . . . . . . . . 64
5.2.1 Methods for Topic Modeling . . . . . . . . . . . 65
5.2.2 Topic Evolution of Scientific Publications . . . . 67
5.3 Summary . . . . . . . . . . . . . . . . . . . . . . . . . 68
6 BENCHMARK FOR SCIENTIFIC TREND DETECTION 71
6.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . 71
6.1.1 Outline and Contributions . . . . . . . . . . . . 72
6.2 Corpus Creation . . . . . . . . . . . . . . . . . . . . . 73
6.2.1 Underlying Data . . . . . . . . . . . . . . . . . 73
6.2.2 Stratification . . . . . . . . . . . . . . . . . . . 73
6.2.3 Document representations & Clustering . . . . . 74
6.2.4 Trend & Downtrend Estimation . . . . . . . . . 75
6.2.5 (Down)trend Candidates Validation . . . . . . . 75
6.3 Benchmark Annotation . . . . . . . . . . . . . . . . . 77
6.3.1 Inter-annotator Agreement . . . . . . . . . . . . 77
6.3.2 Gold Trends . . . . . . . . . . . . . . . . . . . . 78
6.4 Evaluation Measure for Trend Detection . . . . . . . . . 79
6.5 Trend Detection Baselines . . . . . . . . . . . . . . . . 80
6.5.1 Configurations . . . . . . . . . . . . . . . . . . 81
6.5.2 Experimental Setup and Results . . . . . . . . . 81
6.6 Benchmark Description . . . . . . . . . . . . . . . . . 82
6.7 Summary . . . . . . . . . . . . . . . . . . . . . . . . . 83
6.8 Future Work . . . . . . . . . . . . . . . . . . . . . . . . 83
7 DISCUSSION AND FUTURE WORK 85

BIBLIOGRAPHY 87

# LIST OF FIGURES


| | | |
|---|---|---|
| Figure 1 | Taxonomy of dialogue systems. | 10 |
| Figure 2 | Example of a conventional dialogue architecture. | 12 |
| Figure 3 | *OMB+* online learning platform. | 24 |
| Figure 4 | Ruled-Based System. Dialogue flow | 36 |
| Figure 5 | Macro F1 evaluation. NAP – default model. Comparison between German and multilingual BERT. | 52 |
| Figure 6 | Macro F1 test. NAP – default model. Comparison between German and multilingual BERT. | 53 |
| Figure 7 | Macro F1 evaluation. NAP – extended model. | 54 |
| Figure 8 | Macro F1 test. NAP – extended model. | 55 |
| Figure 9 | Macro F1 evaluation. NER – Comparison between German and multilingual BERT. | 56 |
| Figure 10 | Macro F1 test. NER – Comparison between German and multilingual BERT. | 57 |
| Figure 11 | (Down)trend detection. Overall distribution of papers in the entire dataset. | 73 |
| Figure 12 | A trend candidate (positive slope); Topic: Sentiment and Emotion Analysis (using Social Media Channels) | 76 |
| Figure 13 | A downtrend candidate (negative slope); Topic: XML | 76 |

# LIST OF TABLES


| | | |
|---|---|---|
| Table 1 | Natural language representation of a sentence. | 13 |
| Table 2 | Rule 1: Admissible configurations. | 27 |
| Table 3 | Rule 2: Admissible configurations. | 27 |
| Table 4 | Rule 3: Admissible configurations. | 27 |
| Table 5 | Rule 4: Admissible configurations. | 28 |
| Table 6 | Rule 5: Admissible configurations. | 28 |
| Table 7 | Case 1: ID completeness. | 31 |
| Table 8 | Case 2: ID completeness. | 31 |
| Table 9 | Showcase 1: Short flow. | 33 |
| Table 10 | Showcase 2: Organisational question. | 33 |
| Table 11 | Showcase 3: Fallback and human-request policies. | 34 |
| Table 12 | Showcase 4: Contextual question. | 34 |
| Table 13 | Showcase 5: Long flow. | 35 |
| Table 14 | General statistics for conversational dataset. | 46 |
| Table 15 | Detailed statistics on possible systems actions in conversational dataset. | 48 |
| Table 16 | Detailed statistics on possible named entities in conversational dataset. | 48 |
| Table 17 | F1-results for the Named Entity Recognition task. | 51 |
| Table 18 | F1-results for the Next Action Prediction task. | 51 |
| Table 19 | Average dialogue accuracy computed for the Next Action Prediction task. | 51 |
| Table 20 | Trend detection. List of gold trends. | 79 |
| Table 21 | Trend detection. Example of MAP evaluation. | 80 |
| Table 22 | Trend detection. Ablation results. | 82 |

AI Artificial Intelligence

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

BiLSTM Bidirectional Long Short Term Memory

CRF Conditional Random Fields

DM Dialogue Manager

DST Dialogue State Tracker

DSTC Dialogue State Tracking Challenge

EC Equivalence Classes

ES ElasticSearch

GM Gaussian Models

IAA Inter Annotator Agreement

ID Informational Dictionary

IPA Intelligent Process Automation

ITS Intelligent Tutoring System

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

LSTM Long Short Term Memory

MAP Mean Average Precision

MER Mutually Exclusive Rules

NAP Next Action Prediction

NER Named Entity Recognition

NLG Natural Language Generation

NLP Natural Language Processing

NLU Natural Language Understanding

OMB+ Online Mathematik Brückenkurs

PL Policy Learning

pLSA probabilistic Latent Semantic Analysis

REST Representational State Transfer

RNN Recurrent Neural Network

RPA Robotic Process Automation

RegEx Regular Expressions

TL Transfer Learning

ToDS Task-oriented Dialogue System

# 1 INTRODUCTION

Robotic Process Automation (RPA) is a type of software bots that imitates manual human activities like entering data into a system, logging into accounts, or executing simple but repetitive workflows (Mohanty and Vyas, 2018). In other words, RPA-systems allow business applications to work without human involvement by replicating human worker activities. A further benefit of RPA-bots is that they are much cheaper compared to the employment of a human worker, and they are also able to work around-the-clock. Overall advantages of software bots are hugely appealing and include cost savings, accuracy and compliance while executing processes, improved responsiveness (i.e., bots are faster than human workers), and finally, such bots are usually agile and multi-skilled (Ivančić, Vugec, and Vukšić, 2019).

However, one of the main benefits and at the same time – drawbacks of the RPA-systems, is their ability to precisely fulfill the assigned task. On the one hand, the bot can carry out the task accurately and diligently. On the other hand, it is highly susceptible to changes in defined scenarios. Being designed for a particular task, the RPA-bot is often not adaptable to other domains or even simple changes in a workflow (Mohanty and Vyas, 2018). This inability to adapt to changing conditions gave rise to a further area of improvement for RPA-bots – Intelligent Process Automation (IPA) systems.

IPA-bots combine Robotic Process Automation (RPA) with Artificial Intelligence (AI) and thus can perform complex and more cognitively demanding tasks that require reasoning and language understanding. Hence, IPA-bots went beyond automating simple "click tasks" (as in RPA) and can accomplish jobs more intelligently – employing machine learning algorithms and advanced analytics. Such IPA-systems undertake time-consuming and routine tasks, and thus enable smart workflows and free up skilled workers to accomplish higher-value activities.

Previously, IPA techniques were primarily employed within the computer vision domain. Starting with the automation of simple display click-tasks and going towards sophisticated self-driving cars with their ability to interpret a stream of situational data obtained from sensors and cameras. However, in recent times, Natural Language Processing (NLP) became one of the potential applications for IPA as well due to its ability to understand and interpret human language. NLP methods are used to analyze large amounts of textual data and to respond to various inquiries. Furthermore, if the available data is unstructured and does not have any predefined format (i.e., customer emails), or if it is available in a variable format (i.e., invoices, legal documents), then the Natural Language Processing (NLP) techniques are applied to extract the multiple relevant information. This data then can be utilized to solve distinct problems (i.e., natural language understanding, predictive analytics) (Mohanty and Vyas, 2018). For instance, in the case of predictive analytics, such a bot would make forecasts about the future using current and historical data and therethrough assist with sales or investments.

However, NLP in the context of Intelligent Process Automation (IPA) is not limited to the extraction of relevant data and insights from textual documents. One of the central applications of NLP within the IPA domain – are conversational interfaces (e.g., chatbots) that are used to enable human-to-machine interaction. This type of software is commonly used for customer support or within recommendation- and booking systems. Conversational agents gain enormous demand due to their ability to simultaneously support a large number of users while communicating in a natural language. In a conventional chatbot system, a user provides input in a natural language, the bot then processes this query to extract potentially useful information and responds with a relevant reply. This process comprises multiple stages and involves different types of NLP subtasks, starting with Natural Language Understanding (NLU) (e.g., intent recognition, named entity extraction) and going towards dialogue management (i.e., determining the next possible bots action, considering the dialogue history) and response generation (e.g., converting the semantic representation of next bots action to a natural language utterance). Typical dialogue system for Intelligent Process Automation (IPA) purposes usually undertakes shallow customer support requests (e.g., answering of FAQs), allowing human workers to focus on more sophisticated inquiries.

Natural Language Processing (NLP) techniques may also be used indirectly for IPA purposes. There are attempts to automatically identify and classify tasks extracted from textual process descriptions as manual, user, or automated. The goal of such an NLP application is to reduce the effort required to identify suitable candidates for further robotic process automation (Friedrich, Mendling, and Puhlmann, 2011; Leopold, Aa, and Reijers, 2018).

Intelligent Process Automation (IPA) is an emerging technology and evolves mainly due to the recognized potential benefit of combining Robotic Process Automation (RPA) and Artificial Intelligence (AI) techniques to solve industrial problems (Ivančić, Vugec, and Vukšić, 2019; Mohanty and Vyas, 2018). Industries are typically interested in being responsive to customers and markets (Mohanty and Vyas, 2018). Thus, most customer support applications need to be real-time active and highly robust to achieve higher client satisfaction rates. The manual accomplishment of such tasks is often time-consuming, expensive and error-prone. Furthermore, due to the massive amounts of textual data available for analysis, it seems not to be feasible to process this data entirely by human means.

## 1.1 OUTLINE

This thesis covers two topics united by the broader theme, which is Intelligent Process Automation (IPA) utilizing statistical Natural Language Processing (NLP) methods.

THE FIRST BLOCK OF THE THESIS (Chapter 2 – Chapter 4) addresses the challenge of implementing a *conversational agent* for Intelligent Process Automation (IPA) purposes within the e-learning field. As mentioned above, chatbots are one of the central applications of the IPA domain, since they can effectively perform time-consuming tasks requiring natural language understanding, allowing human workers to concentrate on more valuable tasks. The IPA conversational bot that was realized within this thesis takes care of routine and time-consuming tasks performed by human tutors within an online mathematical course. This bot is deployed in a real-world setting and interact with students within the OMB+ mathematical platform. Conducting experiments, we observed two possibilities to build the conversational agent in industrial settings – first, with purely rule-based methods, considering the missing training data and individual aspects of the target domain (i.e., e-learning). Second, we reimplemented two of the main system components (i.e., Natural Language Understanding (NLU) and Dialogue Manager (DM) units) using the current state-of-the-art deep-learning algorithm (i.e., Bidirectional Encoder Representations from Transformers (BERT)) and investigated their performance and potential use as a part of a hybrid model (i.e., containing both rule-based and machine learning methods).

THE SECOND BLOCK OF THE THESIS (Chapter 5 – Chapter 6) considers an Intelligent Process Automation (IPA) problem within the *predictive analytics* domain and addresses the task of scientific trend detection. Predictive analytics makes forecasts about future outcomes based on historical and current data. Hence, organizations can benefit through the advanced analytics models by detecting trends and their behaviors, and then employing this information while making crucial business decisions (i.e., investments). In this work, we deal with the subproblem of trend detection task – specifically, we address the lack of publicly available benchmarks for the evaluation of trend detection algorithms. Therefore, we assembled the benchmark for both *trends* and *downtrends* (i.e., topics that become less frequent overtime) detection. To the best of our knowledge, the task of downtrend detection has not been well addressed before. Furthermore, the resulting benchmark is based on a collection of more than one million documents, which is among the largest that has been used for trend detection before and therefore offers a realistic setting for developing trend detection algorithms.

All main chapters contain their specific introduction, contribution list, related work section, and summary.

Part I

# E-LEARNING CONVERSATIONAL ASSISTANT

# 2 FOUNDATIONS

In this block, we address the challenge of designing and implementing a conversational assistant for Intelligent Process Automation (IPA) purposes within the e-learning domain. The system aims to support human tutors of the Online Mathematik Brückenkurs (OMB+) platform during the tutoring round by undertaking routine and time-consuming responsibilities. Specifically, the system intelligently automates the process of information accumulation and validation, that precede every conversation. Therethrough, the IPA-bot frees up tutors and allows them to concentrate on more complicated and cognitively demanding tasks.

In this chapter, we introduce the fundamental concepts required by the topic of the first part of the thesis. We begin with the foundations to conversational assistants in Section 2.2, where we present the taxonomy of existing dialogue systems in Section 2.2.1, give an overview of the conventional dialogue architecture in Section 2.2.2, with its most essential components, and describe the existing approaches for building a conversational agent in Section 2.2.3. We finally introduce the target domain for our experiments in Section 2.3.

## 2.1 INTRODUCTION

Conversational Assistants - a type of software that interacts with people via written or spoken natural language, have become widely popularized in recent times. Today's conversational assistants support a broad range of everyday skills, including, but not limited to, creating a reminder, answering factoid questions, and controlling smart home devices.

Initially, the notion of artificial companion capable of conversing in a natural language was anticipated by Alan Turing in 1949 (Turing, 1949). Still, the first significant step in this direction was done in 1966 with the appearance of Weizenbaums system *ELIZA* (Weizenbaum, 1966). Its successor was the *A.L.I.C.E.* (Artificial Linguistic Internet Computer Entity) – a natural language processing dialogue system that conversed with a human by applying heuristical pattern matching rules. Despite being remarkably popular and technologically advanced for their days, both systems were unable to pass the Turing test (Alan, 1950). Following a long silent period, the next wave of growth of intelligent agents was caused by the advances in Natural Language Processing (NLP). This was enabled by access to low-cost memory, higher processing speed, vast amounts of available data, and sophisticated machine learning algorithms. Hence, one of the most notable leaps in the history of conversational agents began in the year 2011 with *intelligent personal assistants*. In that time *Apples Siri*, followed by *Microsofts Cortana* and *Amazons Alexa* in 2014, and then *Google Assistant* in 2016, came to the scene. The main focus of these agents was not to mimic humans, as in ELIZA or A.L.I.C.E, but to assist a human by answering simple factoid questions and setting notification tasks across a broad range of content. Another type of conversational agent, called Task-oriented Dialogue System (ToDS), became a focus of industries in the year 2016. For the most part, these bots aimed to support customer service (Cui et al., 2017) and respond to Frequently Asked Questions (Liao et al., 2018). Their conversational style provided more depth than those of personal assistants, but a narrower range, since they were patterned for task solving and not for open-domain discussions.

One of the central advantages of conversational agents and the reason why they are attracting considerable interest is their ability to give attention to several users simultaneously while supporting natural language communication. Due to this, conversational assistants gain momentum in numerous kinds of practical support applications (i.e., customer support), where a high number of users are involved at the same time. Classical engagement of human assistants is very cost-efficient, and such support is usually not full-time accessible. Therefore, to automate human assistance is desirable but also an ambitious task. Due to the lack of solid Natural Language Understanding (NLU), advancing beyond primary tasks is still challenging (Luger and Sellen, 2016), and even the most prosperous dialogue systems often fail to meet user's expectations (Jain et al., 2018). Significant challenges involved in the design and implementation of a conversational agent varying from domain engineering to Natural Language Understanding (NLU), and designing of an adaptive domain-specific conversational flow. The quintessential problems and "how-to" questions while building conversational assistants include:

- How to deal with *variability* and *flexibility* of language?
- How to get high-quality *in-domain data*?
- How to build *meaningful representations*?
- How to integrate *commonsense* and *domain knowledge*?
- How to build a *robust* system?

Besides that, current Natural Language Processing (NLP) and Artificial Intelligence (AI) techniques have weaknesses when applied to domain-specific task-oriented problems, especially if underlying data comes from an industrial source. As machine learning techniques heavily rely on structured and cleaned training data to deliver consistent performance, typically noisy real-world data without access to additional knowledge databases negatively impacts the classification accuracy.

Despite that, conversational agents could be successfully employed to solve less cognitive but still highly relevant tasks. In this case, a dialogue agent could be operated as a part of the Intelligent Process Automation (IPA) system, where it takes care of straightforward but routine and time-consuming functions performed in a natural language while allowing a human worker to focus on more cognitive demanding processes.

## 2.2 CONVERSATIONAL ASSISTANTS

Development and design of a particular dialogue system usually vary by its objectives, among the most fundamental are:

- **Purpose**: Should it be an open-ended dialogue system (e.g., chit-chat), task-oriented system (e.g., customer support) or a personal assistant (e.g., Google Assistant)?;
- **Skills**: What kind of behavior should a system demonstrate?
- **Data**: Which data is required to develop (resp. train) specific blocks and how could it be assembled and curated?
- **Deployment**: Will the final system be integrated into a particular software platform? What are the challenges in choosing suitable tools? How will the deployment happen?
- **Implementation**: Should it be a rule-based, information-retrieval, machine-learning, or a hybrid system?

In the following chapter, we will take a look at the most common taxonomies of dialogue systems, and different design and implementation strategies.

### 2.2.1 *Taxonomy*

According to the *goals*, conversational agents could be classified into two main categories: *task-oriented systems* (e.g., for customer support), and *social bots* (e.g., for chit-chat purposes). Furthermore, systems also vary regarding their implementation characteristics. Below we provide a detailed overview of the most common variations.

TASK-ORIENTED DIALOGUE SYSTEMS are famous for their ability to lead a dialogue with the ultimate goal of solving a user's problem (see Figure 1). A customer support agent or a flight/restaurant booking system exemplify this class of conversational assistants. Such systems are more meaningful and useful as those with an entirely social goal; however, they usually affect more complications during the implementation stage. Since Task-oriented Dialogue System (ToDS) require a precise understanding of the user's intent (i.e., goal), the Natural Language Understanding (NLU) segment must perform flawlessly.

Besides that, if applied in an industrial setting, ToDS are usually built modular, which means they mostly rely on handcrafted rules and predefined templates, and are restricted in their abilities. *Personal assistants* (i.e., Google Assistant) are members of the task-oriented family as well. Though, compared to standard agents, they involve more social conversation and could be used for entertainment purposes (e.g., *"Ok Google, tell me a joke!"*). The latter is typically not available in conventional task-oriented systems.

SOCIAL BOTS AND CHIT-CHAT SYSTEMS in contrast, regularly do not hold any specific goal and are designed for small-talk and entertainment conversations (see Figure 1). Those agents are usually end-to-end trainable and highly data-driven, which means they require large amounts of training data. However, due to entirely learnable training, such systems often produce inappropriate responses, making them extremely unreliable for practical applications. Alike chit-chat systems, *social bots* are rich in social conversation; however, they possess an ability to solve uncomplicated user requests and conduct a more meaningful dialogue (Fang et al., 2018). Those requests are not the same, as in task-oriented systems and are mostly pointed to answer simple factoid questions.

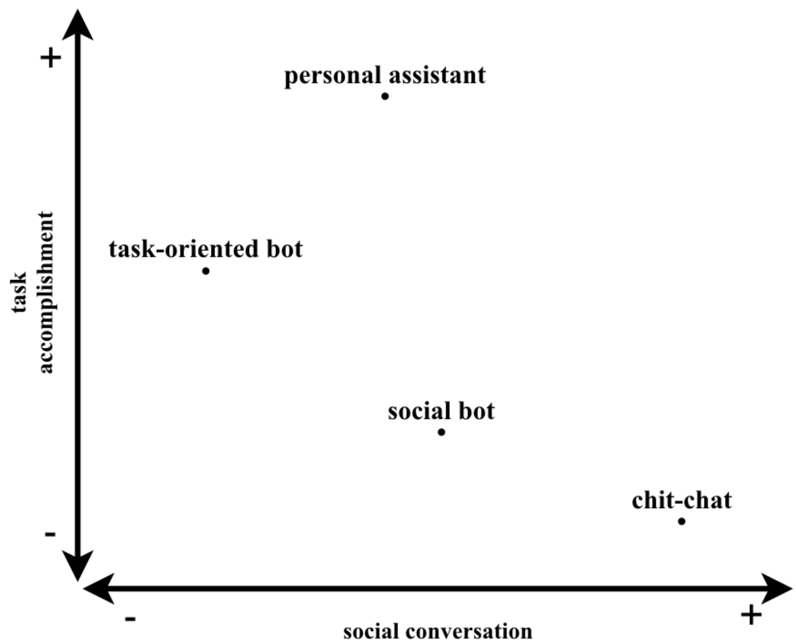


Figure 1: Taxonomy of dialogue systems according to the measure of their task accomplishment and social contribution. Figure originates from Fang et al., 2018.

The aforementioned system types could be decomposed farther according to the implementation characteristics:

**OPEN DOMAIN & CLOSED DOMAIN:** A Task-oriented Dialogue System (ToDS) is typically built within a *closed domain* setting. The space of potential inputs and outputs is restricted since the system attempts to achieve a specific goal. Customer support or shopping assistants are cases of closed domain problems (Wen et al., 2016). These systems do not need to be able to talk about off-site topics, but they have to fulfill their particular tasks as efficiently as possible.

Whereas, chit-chat systems do not assume a well-defined goal or intention and thus, could be built within an *open domain*. This means the conversation can progress in any direction and without any or minimal functional purpose (Serban et al., 2016). However, the infinite number of various topics and the fact that a certain amount of commonsense is required to create reasonable responses makes an open-domain setting a hard problem.

**CONTENT-DRIVEN & USER-CENTRIC:** Social bots and chit-chat systems are typically utilizing a *content-driven* dialogue manner. That means that their responses are based on the large and dynamic content derived from the daily web mining and knowledge graphs. This approach builds a dialogue based on trendy content from diverse sources, and thus, it is an ideal way for social bots to catch users' attention and bring a user into the dialogue.

*User-centric* approaches, in turn, assume precise language understanding to detect the sentiment of users' statements. According to the sentiment, the dialogue manager learns users' personalities, handles rapid topic changes, and tracks engagement. In this case, the dialogue builds on specific user interests.

**LONG & SHORT CONVERSATIONS** Models with a *short conversation* style attempt to create a single response to a single input (e.g., weather forecast). In the case with a *long conversation*, a system has to go through multiple turns and to keep track of what has been said before (i.e., dialogue history). The longer the conversation, the more challenging it is to train a system. Customer support conversations are typically long conversational threads with multiple questions.

**RESPONSE GENERATION** One of the essential elements of a dialogue system is a response generation. This could be either done in a *retrieval-based* manner or by using *generative models* (i.e., Natural Language Generation (NLG)).

The former usually utilizes predefined responses and heuristics to select a relevant response based on the input and context. The heuristic could be a rule-based expression match (i.e., Regular Expressions (RegEx)) or an ensemble of machine learning classifiers (e.g., entity extraction, prediction of next action). Such systems do not generate any original text; instead, they select a response from a fixed set of predefined candidates. These models are usually used in an industrial setting, where the system must show high robustness, performance and interpretability of the outcomes.

The latter, in contrast, do not rely on predefined replies. Since such models are typically based on (deep-) machine learning techniques, they attempt to generate new responses from scratch. This is, however, a complicated task, and is mostly used for vanilla chit-chat systems or in a research domain, where the structured training data is available.

### 2.2.2 *Conventional Architecture*

In this and the following subsection, we review the pipeline and methods for Task-oriented Dialogue System (ToDS). Such agents are typically composed of several components (Figure 2) addressing diverse tasks required to facilitate a natural language dialogue (Williams, Raux, and Henderson, 2016).

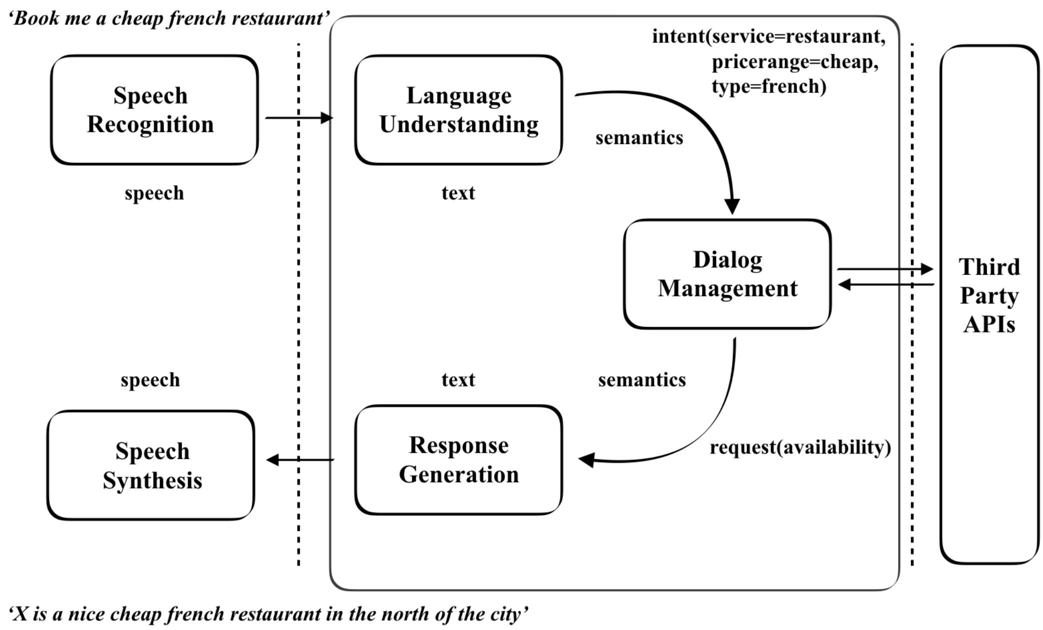


Figure 2: Example of a conventional dialogue architecture with its main components in the bounding box: Language Understanding (NLU), Dialogue Manager (DM), Response Generation (NLG). Figure originates from NAACL 2018 Tutorial, Su et al., 2018.

A conventional dialogue architecture implements this with the following components:

**NATURAL LANGUAGE UNDERSTANDING UNIT** has the following objectives: it parses the user input, classifies the *domain* (i.e., customer support, restaurant booking; not required for a single domain systems) and *intent* (i.e., find-flight, book-taxi), and extracts *entities* (i.e., Italian food, Berlin).

Generally, NLU unit maps every user utterance into semantic frames, that are predefined according to different scenarios. Table 1 illustrates an example of sentence representation, where the departure-location (*Berlin*) and the arrival-location (*Munich*) are specified as slot values, and the domain (*airline travel*) and intent (*find-flight*) are specified as well. Typically, there are two kinds of possible representations. The first one is the utterance category, represented in the form of the user's intent. The second could be the word-level information extraction in the form of Named Entity Recognition (NER) or Slot Filling tasks.

| **Sentence** | *Find* | *flight* | *from* | *Berlin* | *to* | *Munich* | *today* |
|---|---|---|---|---|---|---|---|
| **Slots/Concepts** | O | O | O | B-dept | O | B-arr | B-date |
| **Named Entity** | O | O | O | B-city | O | B-city | O |
| **Intent** | Find_Flight | | | | | | |
| **Domain** | Airline Travel | | | | | | |

Table 1: An illustrative example of a sentence representation[1].

A dialog act classification sub-module, also known as *intent classification*, is dedicated to detecting the primary user's goal at every dialogue state. It labels each user's utterance with one of the predefined intents. Deep neural networks can be directly applied to this conventional classification problem (Khanpour, Guntakandla, and Nielsen, 2016; Lee and Dernoncourt, 2016).

*Slot filling* is another sub-module for language understanding. Unlike intent detection (i.e., classification problem), slot filling is defined as a sequence labeling problem, where words in the sequence are labeled with semantic tags. The input is a sequence of words, and the output is a sequence of slots, one for every word. Variations of Recurrent Neural Network (RNN) architectures (LSTM, BiLSTM) (Deoras et al., 2015) and (attention-based) encoder-decoder architectures (Kurata et al., 2016) have been employed for this task.

DIALOGUE MANAGER UNIT is subdivided into two components – first, is a Dialogue State Tracker (DST), that holds the *conversational state*, and second, is a Policy Learning (PL), that defines the systems *next action*.

Dialogue State Tracker (DST) is the core component to guarantee a robust dialogue flow since it manages the user's intent at every turn. The conventional methods, which are broadly applied in commercial implementations, often adopt handcrafted rules to pick the most probable candidate (Goddeau et al., 1996; Wang and Lemon, 2013) among the predefined. Though, a variety of statistical approaches have emerged since the Dialogue State Tracking Challenge (DSTC)[2] was arranged by Jason D. Williams in year 2013. Among the deep-learning based approaches is the *Belief Tracker* proposed by Henderson, Thomson, and Young (2013). It employs a sliding window to produce a sequence of probability distributions over an arbitrary number of possible values (Chen et al., 2017). In the work of Mrkšić et al. (2016) authors introduced a *Neural Belief Tracker* to identify the slot-value pairs. The system used the user intents preceding the user input, the user utterance itself, and a candidate slot-value pair which it needs to decide about, as the input. Then the system iterated overall candidate slot-value pairs to determine which ones have just been expressed by the user (Chen et al., 2017). Finally, Vodolán, Kadlec, and Kleindienst proposed a *Hybrid Dialog State Tracker*, that achieved the state-of-the-art performance for DSTC2 shared task. Their model used a separate-learned decoder coupled with a rule-based system.

Based on the state representation from the Dialogue State Tracker (DST), the PL unit generates the *next system action*, that is then decoded by the Natural Language Generation (NLG) module. Usually, a rule-based agent is used to initialize the system (Yan et al., 2017) and then, supervised learning is applied to the actions generated by these rules. Alternatively, the dialogue policy can be trained end-to-end with reinforcement learning to manage the system making policies toward the final performance (Cuayáhuitl, Keizer, and Lemon, 2015).

NATURAL LANGUAGE GENERATION UNIT transforms the *next action* into a natural language response. Conventional approaches to Natural Language Generation (NLG) typically adopt a template-based style, where a set of rules is defined to map every next action to a predefined natural language response. Such approaches are simple, generally error-free and easy to control; however, their implementation is time-consuming, and the style is repetitive and hardly scalable.

Among the deep learning techniques is the work by Wen et al. (2015), that introduced neural network-based approaches to NLG with an LSTM-based structure. Later, Zhou and Huang

[2]Currently: *Dialog System Technology Challenge*

(2016) adopted this model along with the question information, semantic slot values, and dialogue act types to generate more precise answers. Finally, Wu, Socher, and Xiong (2019) proposed *Global to Local Memory Pointer Networks (GLMP)* – an architecture that incorporates knowledge bases directly into a learning framework (due to the *Memory Networks* component) and is able to re-use information from the user input (due to the *Pointer Networks* component).

In particular cases, dialogue flow can include speech recognition and speech synthesis units (if managing a spoken natural language system) and connection to third-party APIs to request information stored in external databases (e.g., weather forecast) (as depicted in Figure 2).

### 2.2.3 *Existing Approaches*

As stated above, a conventional task-oriented dialogue system is implemented in a pipeline manner. That means that once the system receives a user query, it interprets it and acts according to the dialogue state and the corresponding policy. Additionally, based on understanding, it may access the knowledge base to find the demanded information there. Finally, a system transforms a system's next action (and if given, an extracted information) into its surface form as a natural language response (Monostori et al., 1996).

Individual components (i.e., Natural Language Understanding (NLU), Dialogue Manager (DM), Natural Language Generation (NLG)) of a particular conversational system could be implemented using different approaches, starting with entirely *rule- and template-based methods*, and going towards *hybrid approaches* (using learnable components along with handcrafted units) and *end-to-end* trainable machine learning methods.

RULE-BASED APPROACHES: Though many of the latest research approaches handle Natural Language Understanding (NLU) and Natural Language Generation (NLG) units by using statistical Natural Language Processing (NLP) models (Bocklisch et al., 2017; Burtsev et al., 2018; Honnibal and Montani, 2017), most of the industrially deployed dialogue systems still utilise handcrafted features and rules for the state tracking, action prediction, intent recognition, and slot filling tasks (Chen et al., 2017; Ultes et al., 2017). So, most of the PyDial[3] framework modules offer rule-based implementations using Regular Expressions (RegEx), rule-based dialogue trackers (Henderson, Thomson, and Williams, 2014), and handcrafted policies. Also, in order to map the next action to text, Ultes et al. (2017) suggest rules along with a template-based response generation.

[3] PyDial Framework: `http://www.camdial.org/pydial/`

The rule-based approach ensures robustness and stable performance that is crucial for industrial systems that communicate with a large number of users simultaneously. However, it is highly expensive and time-consuming to deploy a real dialogue system built in that manner. The major disadvantage is that the usage of handcrafted systems is restricted to a specific domain, and possible adaptation requires extensive manual engineering.

END-TO-END LEARNING APPROACHES: Due to the recent advance of end-to-end neural generative models (Collobert et al., 2011), many efforts have been made to build an end-to-end trainable architecture for conversational agents. Rather than using the traditional pipeline (see Figure 2), the end-to-end model is conceived as a single module (Chen et al., 2017).

Wen et al. (2016), followed by Bordes, Boureau, and Weston (2016) were among the pioneers who adapted an end-to-end trainable approach for conversational systems. Their approach was, to treat dialogue learning as the problem of learning a mapping from dialogue histories to system responses, and to apply an encoder-decoder model (Sutskever, Vinyals, and Le, 2014) to train the entire system. The proposed model still had its significant limitations (Glasmachers, 2017), such as missing policies to learn a dialogue flow and the inability to incorporate additional knowledge that is especially crucial for training a task-oriented agent. To overcome the first problem, Dhingra et al. (2016) introduced an end-to-end *reinforcement learning* approach that jointly trains dialogue state tracker and policy learning units. The second weakness was addressed by Eric and Manning (2017), who augmented existing recurrent network architectures with a differentiable attention-based key-value retrieval mechanism over the entries of a knowledge base. This approach was inspired by *Key-Value Memory Networks* (Miller et al., 2016). Later on, Wu, Socher, and Xiong (2019) proposed *Global to Local Memory Pointer Networks (GLMP)* where the authors incorporated knowledge bases directly into a learning framework. In their model, a global memory encoder and a local memory decoder are employed to share external knowledge. The encoder encodes dialogue history, modifies global contextual representation, and generates a global memory pointer. The decoder first produces a sketch response with empty slots. Next, it transfers the global memory pointer to filter the external knowledge for relevant information and then instantiates the slots via the local memory pointers.

Despite having better adaptability compared to any rule-based system and being simpler to train, end-to-end approaches remain unattainable for commercial conversational agents, that operating on real-world data. A well and carefully constructed task-oriented dialogue system in an observed domain using handcrafted rules (in NLU and DST units), and predefined responses for NLG, still outperforms the end-to-end systems due to its robustness (Glasmachers, 2017; Wu, Socher, and Xiong, 2019).

HYBRID APPROACHES: Though end-to-end learning is an attractive solution for conversational systems, current techniques are data-intensive and require large amounts of dialogues to learn simple behaviors. To overcome this barrier, Williams, Asadi, and Zweig (2017) introduce *Hybrid Code Networks (HCNs)*, which is an ensemble of retrieval and trainable units. System utilizes an Recurrent Neural Network (RNN) for Natural Language Understanding (NLU) block, domain-specific knowledge that are encoded as software (i.e., API calls), and custom action templates used for Natural Language Generation (NLG) unit. Compared to existing end-to-end methods, the authors report that their approach considerably reduces the amount of data required for training (Williams, Asadi, and Zweig, 2017). According to the authors, *HCNs* achieve state-of-the-art performance on the bAbI dialog dataset (Bordes, Boureau, and Weston, 2016) and outperform two commercial customer dialogue systems[4].

Along with this work, Wen et al. (2016) propose a neural network-based model that is end-to-end trainable but still modularly connected. In their work, authors treat dialogue as a sequence to sequence mapping problem (Sutskever, Vinyals, and Le, 2014) augmented with the dialogue history and the current database search outcome (Wen et al., 2016). The authors explain the learning process as follows: At every turn, the system takes a user input represented as a sequence of tokens and converts it into two internal representations: a distributed representation of the intent and a probability distribution over slot-value pairs called the belief state (Young et al., 2013). The database operator then picks the most probable values in the belief state to form the search result. At the same time, the intent representation and the belief state are transformed and combined by a Policy Learning (PL) unit to form a single vector representing the next system action. This system action is then used to generate a sketch response token by the token. The final system response is composed by substituting the actual values of the database entries into the sketch sentence structure (Wen et al., 2016).

Hybrid models appear to replace the established rule- and template-based approaches currently utilized in an industrial setting. The performance of the most Natural Language Understanding (NLU) components (i.e., Named Entity Recognition (NER), domain and intent classification) is reasonably high to apply them for the development of commercial conversational agents. Whereas the Natural Language Generation (NLG) unit could still be realized as a template-based solution by employing predefined response candidates and predicting the next system action, employing deep learning systems.

[4]Internal Microsoft Research datasets in customer support and troubleshooting domains were employed for training and evaluation.

## 2.3 TARGET DOMAIN & TASK DEFINITION

MUMIE[5] is an open-source e-learning platform for studying and teaching mathematics and computer science. Online Mathematik Brückenkurs (OMB+) is one of the projects powered by MUMIE. Specifically, Online Mathematik Brückenkurs (OMB+)[6] is a German online learning platform that assists students who are preparing for an engineering or computer science study at a university.

The course's central purpose is to assist students in reviving their mathematical skills so that they can follow the upcoming university courses. This online course is thematically segmented into 13 parts and includes free mathematical classes with theoretical and practical content. Every chapter begins with an overview of its sections and consists of explanatory texts with built-in videos, examples, and quick checks of understanding. Furthermore, various examination modes to practice the content (i.e., training, exercises) and to check the understanding of the theory (i.e., quiz) are available. Besides that, OMB+ provides a possibility to get assistance from a human tutor. Usually, the students and tutors interact via a chat interface in written form. The language of communication is German.

The current dilemma of the OMB+ platform is that the number of students grows significantly every year, but it is challenging to find qualified tutors – and it is expensive to hire more human tutors. This results in a longer waiting period for students until their problems can be considered and processed. In general, all student questions can be grouped into three main categories: *Organizational questions* (i.e., course certificate), *contextual questions* (i.e., content, theorem) and *mathematical questions* (exercises, solutions). To assist a student with a mathematical question, a tutor has to know the following regular information: What kind of *topic* (or *subtopic*) a student has a problem with. At which *examination mode* (quiz, chapter level training or exercise, section level training or exercise, or final examination) a student is working right now. And finally, the *exact question number* and *exact problem formulation*. This means that a tutor has to ask the same onboarding questions every time a new conversation starts. This is, however, very time-consuming and could be successfully solved within an Intelligent Process Automation (IPA) system.

The work presented in Chapter 3 and Chapter 4 was enabled by the collaboration with MUMIE and Online Mathematik Brückenkurs (OMB+) team, and addresses the abovementioned problem. Within Chapter 3, we implemented an Intelligent Process Automation (IPA) dialogue system with rule- and retrieval-based algorithms that interacts with students and performs the onboarding. This system is multipurpose: on the one hand, it eases the assistance process for human tutors by reducing repetitive and time-consuming activities and, therefore, allows workers to focus on more challenging tasks. Second, interacting

[5] Mumie: https://www.mumie.net
[6] https://www.ombplus.de/

with students, it augments the resources with structured and labeled training data. The latter, in turn, allowed us to re-train specific system components utilizing machine and deep learning methods. We describe this procedure in Chapter 4.

We report that the earlier collected human-to-human data from the OMB+ platform could not be used for training a machine learning system since it is highly unstructured and contains no direct question-answer matching, which is crucial for trainable conversational algorithms. We analyzed these conversations to get insight from them that we used to define the dialogue flow.

We also do not consider the automatization of mathematical questions in this work since this task would require not only precise language understanding and extensive commonsense knowledge but also the accurate perception of mathematical notations. The tutoring process at OMB+ differentiates a lot from the standard tutoring methods. Here, instructors attempt to navigate students by giving hints and tips instead of providing an out-of-the-box solution. Existing human-to-human dialogues revealed that such kind of task could be even challenging for human tutors, since it is not always directly perceptible, what precisely the student does not understand or has difficulties with. Furthermore, dialogues containing mathematical explanations could last for a long time, and the topic (i.e., intent and final goal) can shift multiple times during the conversation.

To our knowledge, this task would not be feasible to sufficiently solve with current machine learning algorithms, especially considering the missing training data. A rule-based system would not be the right solution for this purpose as well due to a large number of complex rules which would need to be defined. Thus, in this work, we focus on the automatization of less cognitively demanding tasks, which are still relevant and must be solved in the first instance to ease the tutoring process for instructors.

## 2.4 SUMMARY

Conversational agents are a highly demanding solution for industries, due to their potential ability to support a large number of users multi-threaded while performing a flawless natural language conversation. Many messenger platforms are created, and custom chatbots in various constellations emerge daily. Unfortunately, most of the current conversational agents are often disappointing to use due to the lack of substantial natural language understanding and reasoning. Furthermore, Natural Language Processing (NLP) and Artificial Intelligence (AI) techniques have limitations when applied to domain-specific and task-oriented problems, especially if no structured training data is available.

Despite that, conversational agents could be successfully applied to solve less cognitive but still highly relevant tasks – that is, within the

Intelligent Process Automation (IPA) domain. Such IPA-bots would undertake tedious and time-consuming responsibilities to free up the knowledge worker for more complex and intricate tasks. IPA-bots could be seen as a member of goal-oriented dialogue family and are, for the most part, performed either in a rule-based manner or through hybrid models when it comes to a real-world setting.

# 3 MULTIPURPOSE IPA VIA CONVERSATIONAL ASSISTANT

> This chapter partially covers work already published at international peer-reviewed conferences. The relevant publication is Moiseeva et al., 2020. The research described in this chapter was carried out in its entirety by the author of this thesis. The other authors of the publication acted as advisers. At the time of writing this thesis, the system described in this work was deployed at the Online Mathematik Brückenkurs (OMB+) platform.

As we outlined in the introduction, Intelligent Process Automation (IPA) is an emerging technology with the primary goal of assisting the knowledge worker by taking care of repetitive, routine and low-cognitive tasks. Conversational assistants that interact with users in a natural language are a potential application for the IPA domain. Such smart virtual agents can support the user by responding to various inquiries and performing routine tasks carried out in a natural language (i.e., customer support).

In this work, we tackle the challenge of implementing an IPA conversational agent in a real-world industrial context and conditions of missing training data. Our system has two meaningful benefits: First, it decreases monotonous and time-consuming tasks and, hence, lets workers concentrate on more intellectual processes. Second, interacting with users, it augments the resources with structured and labeled training data. The latter, in turn, can be used to retrain a system utilizing machine and deep learning methods.

## 3.1 INTRODUCTION

Conversational dialogue systems can give attention to several users simultaneously while supporting natural communication. They are thus, exceptionally needed for practical assistance applications where a high number of users are involved at the same time. Regularly, dialogue agents require a lot of natural language understanding and commonsense knowledge to hold a conversation reasonably and intelligently. Therefore, nowadays, it is still challenging to substitute human assistance with an intelligent agent completely. However, a dialogue system could be successfully employed as a part of the Intelligent Process Automation (IPA) application. In this case, it would take care of simple, but routine and time-consuming tasks performed in a natural language while allowing a human worker to focus on more cognitive demanding processes.

Recent research in the dialogue generation domain is conducted by employing Artificial Intelligence (AI) techniques like machine- and deep learning (Lowe et al., 2017; Wen et al., 2016). However, those methods require high-quality data for training that is often not directly available for domain-specific and industrial problems. Even though companies gather and store vast volumes of data, it is often low-quality with regards to machine learning approaches (Daniel, 2015). Especially if it concerns dialogue data, which has to be properly structured as well annotated. Whereas, if trained on artificially created datasets (which is often the case in the research setting), the systems can solve only specific problems and are hardly adaptable to other domains. Hence, despite the popularity of deep learning end-to-end models, one still needs to rely on traditional pipelines in practical dialogue engineering, mainly while introducing a new domain.

### 3.1.1 *Outline and Contributions*

This work addresses the challenge of implementing a dialogue system for IPA purposes within the practical e-learning domain under conditions of missing training data. The chapter is structured as follows: In Section 3.2, we introduce our method and describe the design of the rule-based dialogue architecture with its main components, such as Natural Language Understanding (NLU), Dialogue Manager (DM), and Natural Language Generation (NLG) units. In Section 3.3, we present the results of our dual-focused evaluation, and Section 3.4 describes the format of collected structured data. Finally, we conclude in Section 3.5.

Our contributions within this work are as follows:

- We implemented a robust conversational system for IPA purposes within a practical e-learning domain and under the conditions of *missing training (i.e., dialogue) data*.
- The system has two main objectives:
  - First, it reduces repetitive and time-consuming activities and allows workers of the e-learning platform to focus solely on mathematical inquiries;
  - Second, by interacting with users, it augments the resources with structured and partially labeled training data for further implementation of trainable dialogue components;

## 3.2 MODEL

The central purpose of the introduced system is to interact with students at the beginning of every chat conversation and gather information on the *topic* (and *sub-topic*), *examination mode* and *level*, *question number* and *exact problem formulation*. Such onboarding process saves time for human tutors and allows them to handle mathematical questions solely. Furthermore, the system is realized in a way such that it accumulates labeled and structured dialogues in the background.

Figure 4 displays the intact conversation flow. After the system accepts user input, it analyzes it at different levels and extracts information, if such is provided. If some of the information required for tutoring, is missing, the system asks the student to provide it. When all the information points are collected, they are automatically validated and forwarded to a human tutor. The latter can then directly proceed with the assistance process. In the following, we explain the key components of the system.

### 3.2.1 *OMB+ Design*

Figure 3 illustrates the internal structure and design of the OMB+ platform. It has *topics* and *sub-topics*, as well as four various *examination modes*. Each *topic* (Figure 3, tag 1) correspond to a chapter level and always has *sub-topics* (Figure 3, tag 2), that correspond to a section level. Examination modes *training* and *exercise* are ambiguous, because they correspond to either a *chapter* (Figure 3, tag 3) or a *section* (Figure 3, tag 5) level, and it is important to differentiate between them, since they contain different types of content. The mode *final examination* (Figure 3, tag 4) always corresponds to a chapter level, whereas *quiz* (Figure 3, tag 5) can belong only to a section level. According to the design of the OMB+ platform, there are several ways of how a possible dialogue flow can proceed.

### 3.2.2 *Preprocessing*

In a natural language conversation, one may respond in various forms, making the extraction of data from user-generated text challenging due to potential misspellings or confusable spellings (i.e., *Aufgabe 1.a*, *Aufgabe 1 (a)*). Hence, to facilitate a substantial recognition of intents and extraction of entities, we normalize (e.g., correct misspellings, detect synonyms) and preprocess every user input before moving forward to the Natural Language Understanding (NLU) module.

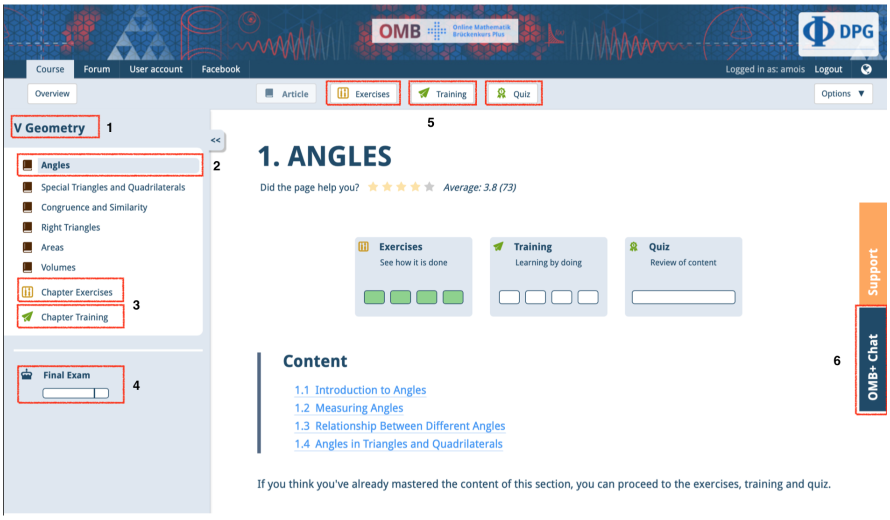


Figure 3: *OMB+* Online Learning Platform, where 1 is the *Topic* (corresponds to a *chapter level*), 2 is a *Sub-Topic* (corresponds to a *section level*), 3 is *chapter level* examination mode, 4 is the *Final Examination Mode* (available only for *chapter level*), 5 are the *Examination Modes*: Exercise, Training (available at *section levels*) and *Quiz* (available only at *section level*), and 6 is the *OMB+ Chat*.

We perform both linguistic and domain-specific preprocessing, which include the following steps:

- lowercasing and stemming of words in the input query;
- removal of stop words and punctuation;
- all mentions of X in mathematical formulations are removed to avoid confusion with roman number 10 ("X");
- in a combination of the type: word *"Kapitel/Aufgabe"* + digit written as a word (i.e *"eins"*, *"zweites"*, etc), word is replaced with a digit (*"Kapitel eins"* → *"Kapitel 1"*; *"im fünften Kapitel"* → *"im Kapitel 5"*), roman numbers are replaced with a digit as well (*"Kapitel IV"* → *"Kapitel 4"*).
- detected ambiguities are normalized (e.g., *"Trainingsaufgabe"* → *"Training"*);
- recognized misspellings resp. type errors are corrected (e.g., *"Difeernzialrechnung"* → *"Differentialrechnung"*)
- permalinks are parsed and analyzed. From each permalink it is possible to extract topic, examination mode and question number;

### 3.2.3 *Natural Language Understanding*

We implement the Natural Language Understanding (NLU) unit utilizing handcrafted rules, Regular Expressions (RegEx) and Elascticsearch[1] API.

INTENT CLASSIFICATION: According to the design of the OMB+ platform, all student questions can be classified into three categories: *organizational questions* (i.e., course certificate), *contextual questions* (i.e., content, theorem) and *mathematical questions* (exercises, solutions). To classify the input query by its *intent* (i.e., category), we employ weighted key-word information in the form of handcrafted rules. We assume that specific words are explicitly associated with a corresponding intent (e.g., *theorem* or *root* denote the mathematical question, *feedback* or *certificate* denote organizational inquire). If no intent could be ordered, then it is assumed that the NLU unit was not capable of understanding, and the intent is *unknown*. In this case, the virtual agent requests the user to provide an intent manually by picking one from the mentioned three options. The queries from organizational and theoretical categories are directly handed over to a human tutor, while mathematical questions are analyzed by the automated agent for further information extraction.

ENTITY EXTRACTION: On the next step, the system retrieves the entities from a user message. We specify five following prerequisite entities, which have to be collected before the dialogue can be handed over to a human tutor: topic, sub-topic, examination mode and level, and question number. This part is implemented using ElasticSearch (ES) and RegEx. To facilitate the use of ES, we indexed the OMB+ web-page to an internal database. Besides indexing the titles of topics and sub-topics, we also provided supplementary information on possible synonyms and writing styles. We additionally filed OMB+ permalinks, which direct to the site pages. To query the resulting database, we employ the internal Elasticsearch *multi_match* function and set the *minimum_should_match* parameter to 20%. This parameter specifies the number of terms that must match for a document to be considered relevant. Besides that, we adopted *fuzziness* with the maximum edit distance set to 2 characters. The fuzzy query uses similarity based on Levenshtein edit distance (Levenshtein, 1966). Finally, the system produces a ranked list of potential matching entries found in the database within the predefined *relevance_threshold* (we set it to $\theta$=1.5). We then pick the most probable entry as the right one and select the corresponding entity from the user input.

Overall, the Natural Language Understanding (NLU) module receives the user input as a preprocessed text and examines it across all

[1] https://www.elastic.co/products/elasticsearch

predefined RegEx statements and for a match in the ElasticSearch (ES) database. We use ES to retrieve entities for the topic, sub-topic, and examination mode, whereas RegEx is used to extract the information on a question number. Every time the entity is extracted, it is filled in the Informational Dictionary (ID). The ID has the following six slots to be filled in: topic, sub-topic, examination level, examination mode, question number, and exact problem formulation (this information is requested on further steps).

### 3.2.4 *Dialogue Manager*

A conventional Dialogue Manager (DM) consists of two interacting components. The first component is the Dialogue State Tracker (DST) that maintains a representation of the current conversational state, and the second is the Policy Learning (PL) that defines the next system action (i.e., response).

In our model, every agent's *next action* is determined by the state of the previously obtained information accumulated in the Informational Dictionary (ID). For instance, if the system recognizes that the student works on the *final examination*, it also knows (defined by hand-coded logic in the predefined rules) that there is no necessity to ask for sub-topic because the final examination always corresponds to a chapter level (due to the layout of OMB+ platform). If the system recognizes that the user struggles in solving *quiz*, it has to ask for both the corresponding topic and sub-topic because the quiz always relates to a section level.

To discover all of the potential conversational flows, we implement Mutually Exclusive Rules (MER), which indicate that two events $e_1$ and $e_2$ are mutually exclusive or disjoint since they cannot both occur at the same time (i.e., the intersection of these events is empty: $P(A \cap B) = 0$). Additionally, we defined *transition* and *mapping rules*. Hence, only particular entities can co-occur, while others must be excluded from the specific rule combination. Assume, a topic $t = Geometry$. According to MER, this topic can co-occur with one of the pertaining sub-topics defined at the OMB+ (i.e., $s = Angel$). Thus, the level of the examination would be $l = Section$ (but $l \neq Chapter$), because the student already reported a sub-section, he or she is working on. In this specific case, the examination mode could either be $e \in [Training, Exercise, Quiz]$, but not $e = Final\_Examination$. This is because, $e = Final\_Examination$ can co-occur only with a topic (*chapter-level*), but not with a sub-topic (*section-level*). The question number $q$ could be arbitrary and has to co-occur with every other combination.

This allows the formal solution to be found. Assume a list of all theoretically possible dialogue states: S = [topic, sub-topic, training, exercise, chapter level, section level, quiz, final examination, question number] and for each element $s_n$ in S is true that:

$$s_n = \begin{cases} 1, & \text{if information is given;} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

This gives all general (resp. possible) dialogue states without reference to the design of the OMB+ platform. However, to make the dialogue states fully suitable for the OMB+, from the general states, we select only those, which are *valid*. To define the *validness* of the state, we specify the following five MER's:

| $R_1$: **Topic** | |
|---|---|
| $\neg$ T | T |

Table 2: Rule 1 – Admissible *topic* configurations.

Rule ($R_1$) in Table 2 denotes admissible configurations for *topic* and means that the topic could be either given (T) or not ($\neg$T).

| $R_2$: **Examination Mode** | | | |
|---|---|---|---|
| $\neg$ TR | $\neg$ E | $\neg$ Q | $\neg$ FE |
| TR | $\neg$ E | $\neg$ Q | $\neg$ FE |
| $\neg$ TR | E | $\neg$ Q | $\neg$ FE |
| $\neg$ TR | $\neg$ E | Q | $\neg$ FE |
| $\neg$ TR | $\neg$ E | $\neg$ Q | FE |

Table 3: Rule 2 – Admissible *examination mode* configurations.

Rule ($R_2$) in Table 3 denotes that either *no* information on the examination mode is given, or examination mode is Training (TR) or Exercise (E) or Quiz (Q) or Final Examination (FE), but not more than one mode at the same time.

| $R_3$: **Level** | |
|---|---|
| $\neg$ CL | $\neg$ SL |
| CL | $\neg$ SL |
| $\neg$ CL | SL |

Table 4: Rule 3 – Admissible *level* configurations.

Rule ($R_3$) in Table 4 indicates that either *no* level information is provided, or the level corresponds to chapter level (CL) or to section level (SL), but not to both at the same time.

| $R_4$: **Examination & Level** | | | |
|---|---|---|---|
| $\neg$ TR | $\neg$ E | $\neg$ SL | $\neg$ CL |
| TR | $\neg$ E | SL | $\neg$ CL |
| TR | $\neg$ E | $\neg$ SL | CL |
| TR | $\neg$ E | $\neg$ SL | $\neg$ CL |
| $\neg$ TR | E | SL | $\neg$ CL |
| $\neg$ TR | E | $\neg$ SL | CL |
| $\neg$ TR | E | $\neg$ SL | $\neg$ CL |

Table 5: Rule 4 – Admissible *examination mode* (only for Training and Exercise) and corresponding *level* configurations.

Rule ($R_4$) in Table 5 means that Training ($TR$) and Exercise ($E$) examination modes can either belong to chapter level ($CL$) or to section level ($SL$), but not to both at the same time.

| $R_5$: **Topic & Sub-Topic** | |
|---|---|
| $\neg$ T | $\neg$ ST |
| T | $\neg$ ST |
| $\neg$ T | ST |
| T | ST |

Table 6: Rule 5 – Admissible *topic* and corresponding *sub-topic* configurations.

Rule ($R_5$) in Table 6 symbolizes that we could be either given only a topic ($T$) or the combination of topic and sub-topic ($ST$) at the same time, or only sub-topic, or no information on this point at all.

We then define a *valid dialogue state*, as a dialogue state that meets all requirements of the abovementioned rules:

$$\forall s(\text{State}(s) \wedge R_{1-5}(s)) \rightarrow \text{Valid}(s) \tag{2}$$

Once we have the valid states for dialogues, we perform a *mapping* from each valid dialogue state to the *next possible systems action*. For that, we first define five *transition rules*. The order of transition rules is important.

$$T_1 := \neg T, \tag{3}$$

means that no topic ($T$) is found in the ID (i.e., could not be extracted from user input).

$$T_2 := \neg EM, \tag{4}$$

indicates that no examination mode ($EM$) is found in the ID.

$$T_3 := \text{EM, where EM} \in [\mathsf{TR}, \mathsf{E}], \tag{5}$$

denotes that the extracted examination mode ($\mathsf{EM}$) is either Training ($\mathsf{TR}$) or Exercise ($\mathsf{E}$).

$$T_4 := \begin{cases} \mathsf{T} & \neg\mathsf{ST} \quad \mathsf{TR} \quad \mathsf{SL}, \\ \mathsf{T} & \neg\mathsf{ST} \quad \mathsf{E} \quad \mathsf{SL}, \\ \mathsf{T} & \neg\mathsf{ST} \quad \mathsf{Q} \end{cases} \tag{6}$$

means that *no* sub-topic ($\mathsf{ST}$) is provided by a user, but ID either already contains the combination of topic ($\mathsf{T}$), training ($\mathsf{TR}$) and section level ($\mathsf{SL}$), or the combination of topic, exercise ($\mathsf{E}$) and section level, or the combination of topic and quiz ($\mathsf{Q}$).

$$T_5 := \neg\text{QNR}, \tag{7}$$

indicates that *no* question number ($\mathsf{QNR}$) was provided by a student (or could not be successfully extracted).

Finally, we assumed the list of possible next actions for the system:

$$A = \begin{cases} \text{Ask for Topic,} \\ \text{Ask for Examination Mode,} \\ \text{Ask for Level,} \\ \text{Ask for Sub-Topic,} \\ \text{Ask for Question Nr.} \end{cases} \tag{8}$$

Following the transition rules, we map each valid dialogue state to the possible next action $a_m$ in $A$:

$$\exists a \forall s(\text{Valid}(s) \wedge T_1(s)) \rightarrow a(s, \text{T}), \tag{9}$$

in the case where we do *not* have any topic provided, the next action is to *ask for the topic* ($\mathsf{T}$).

$$\exists a \forall s(\text{Valid}(s) \wedge T_2(s)) \rightarrow a(s, \text{EM}), \tag{10}$$

if *no* examination mode is provided by a user (or it could not be successfully extracted from the user query), the next action is defined as *ask for examination mode* ($\mathsf{EM}$).

$$\exists a \forall s(\text{Valid}(s) \wedge T_3(s)) \rightarrow a(s, \text{L}), \tag{11}$$

in case where we know the examination mode $\mathsf{EM} \in$ [Training, Exercise], we have to ask about the level (i.e., training at chapter level or training at section level), thus the next action is *ask for level* ($\mathsf{L}$).

$$\exists a \forall s(\text{Valid}(s) \wedge T_4(s)) \rightarrow a(s, \text{ST}), \tag{12}$$

if *no* sub-topic is provided, but the examination mode $\text{EM} \in$ [Training, Exercise] at *section level*, the next action is defined as *ask for subtopic* (ST).

$$\exists a \forall s(\text{Valid}(s) \wedge T_5(s)) \rightarrow a(s, \text{QNR}), \tag{13}$$

if *no* question number is provided by a user, then the next action is *ask for question number* (QNR).

Following this scheme, we generate 56 *state transitions*, which specify the next system actions. Being on a new conversation state, we compare the extracted (or updated) data in the Informational Dictionary (ID) with the valid dialogue states and select the mapped action as the next system's action.

FLEXIBILITY OF DST: The DST transitions outlined above are meant to maintain the current configuration of the OMB+ learning platform. Still, further MERs could be effortlessly generated to create new transitions. Examining this, we performed tests with the previous design of the platform, where all the topics, except for the first one, had *sub-topics*. Whereas the first topic contained both *sub-topics* and *sub-sub-topics*. We could produce the missing transitions with the described approach. The number of permissible transitions, in this case, increased from 56 to 117.

### 3.2.5 *Meta Policy*

Since the proposed system is expected to operate in a real-world setting, we had to develop additional policies that manage the dialogue flow and verify the system's accuracy. We explain these policies below.

COMPLETENESS OF INFORMATIONAL DICTIONARY: The model automatically verifies the completeness of the Informational Dictionary (ID), which is determined by the number of essential slots filled in the informational dictionary. There are 6 discrete cases when the ID is deemed to be complete. For example, if a user works on a final examination mode, the agent does not has to request a sub-topic or examination level. Hence, the ID has to be filled only with data for a topic, examination type, and question number. Whereas, if the user works on training mode, the system has to collect information about the topic, subtopic, examination level, examination mode, and the question number. Once the ID is intact, it is provided to the *verification step*. Otherwise, the agent proceeds according to the next action. The system extracts the information in each dialogue state, and therefore if the user gives updated information on any subject later in the dialogue history, the corresponding slot will be *updated* in the ID.

Below are examples of two final cases (out of six), where Informational Dictionary (ID) is considered to be complete:

$$Case1 = \begin{cases} \text{intent = Math,} \\ \text{topic} \neq \text{None,} \\ \text{exam mode = Final Examination,} \\ \text{level = None,} \\ \text{question nr.} \neq \text{None} \end{cases} \tag{14}$$

Table 7: Case 1 – Any topic, examination mode is final examination, examination level does not matter, any question number.

$$Case2 = \begin{cases} \text{intent = Math,} \\ \text{topic} \neq \text{None,} \\ \text{sub-topic} \neq \text{None,} \\ \text{exam mode} \in \text{[Training, Exercise],} \\ \text{level = Section,} \\ \text{question nr.} \neq \text{None} \end{cases} \tag{15}$$

Table 8: Case 2: Any topic, any related sub-topic, examination mode is either training or exercise, examination level is section, any question number.

VERIFICATION STEP: After the system has collected all the essential data (i.e., ID is complete), it continues to the *final verification* step. Here, the obtained data is presented to the current student in the dialogue session. The student is asked to *review the exactness* of the data, and if any entries are faulty, to correct them. The Informational Dictionary (ID) is, where necessary, updated with the user-provided data. This procedure repeats until the user confirms the correctness of the compiled data.

FALLBACK POLICY: In the cases where the system fails to retrieve information from a student's query, it re-asks a user and attempts to retrieve information from a follow-up response. The maximum number of *re-ask trials* is a parameter and set to three times ($r = 3$). If the system is still incapable of extracting information, the user input is considered as the ground truth and filled to the appropriate slot in ID. An exception to this rule applies where the user has to specify the intent manually. After three unclassified trials, a dialogues session is directly handed over to a human tutor.

HUMAN REQUEST: In every dialogue state, a user can shift to a human tutor. For this, a user can enter the *"human"* (or *"Mensch"* in German language) key-word. Therefore, every user message is investigated for the presence of this key-word.

### 3.2.6 *Response Generation*

The semantic representation of the systems *next action* is transformed into a natural language utterance. Hence, each possible action is mapped to precisely one response that is predefined in the template. Some of the responses are fixed (i.e., *"Welches Kapitel bearbeitest du gerade?")* [2]. Others have *placeholders* for custom values. In the latter case, the utterance can be formulated dependent on the Informational Dictionary (ID).

Assume predefined response: *"Deine Angaben waren wie folgt: a) Kapitel: {K}, b) Abschnitt: {A}, c) Aufgabentyp: {M}, d) Aufgabe: {Q}, e) Ebene: {E}. Habe ich dich richtig verstanden? Bitte antworte mit Ja oder Nein."*[3] The slots $[K, A, M, Q, E]$ are the placeholders for the information, stored in the ID. During the conversation, the system selects the template according to the next action and fills the slots with extracted values if required. After the extracted information is filled in the corresponding placeholders in the utterance, the final response could be as follows : *"Deine Angaben waren wie folgt: a) Kapitel: I Elementares Rechnen, b) Abschnitt: Rechenregel für Potenzen, c) Aufgabentyp: Quiz, d) Aufgabe: 1(a), e) Ebene: Abschnitt. Habe ich dich richtig verstanden? Bitte antworte mit Ja oder Nein."*[4]

Below we demonstrate five sample dialogues in original German language with variable flows:

- short dialogue, where certain information is already provided in the first student's input
- organizational question case
- contextual question case
- example with the manual intent classification.
- example with `fallback` and `human` policies
- long flow with entry correction on the final verification phase

[2] Translates to: "Which chapter are you currently working on?"
[3] Translates to: "Your details were as follows: a) Chapter, b) Section, c) Examination Mode, d) Question Number, e) Level. Did I understand you correctly? Please answer yes or no."
[4] Translates to: "Your details were as follows: a) Elementary Calculus, b) Power Laws, c) Quiz, d) Question 1 (a), e) Section Level. Did I understand you correctly? Please answer yes or no."

**Student:** Hallo, ich habe Schwierigkeiten beim **Kapitel I Elementares Rechnen**, mit der **Übungsaufgabe 1 a**

**Next-Action:** [Ask for a Level: Chapter or Section]
**Bot:** Geht es um eine Aufgabe auf Kapitel-Ebene (z.B um eine Trainingsaufgabe im Kapitel "Geometrie") oder um eine Aufgabe auf der Ebene eines Abschnitts in einem Kapitel (z.B. um eine Aufgabe zum Abschnitt "Winkel" im Kapitel "Geometrie")? Antworte bitte mit KAP für Kapitel und SEK für Abschnitt.

**Student:** Kapitel

**Next-Action:** [Final-Statement]
**Bot:** Deine Angaben waren wie folgt:
a) Kapitel: I Elementares Rechnen,
b) Aufgabentyp: Exercise,
c) Aufgabe: 1a,
d) Ebene: Kapitel.
Habe ich dich richtig verstanden? Bitte antworte mit Ja oder Nein.

**Student:** yep

**Next-Action:** [Human Hand-Over]
**Bot:** Vielen Dank, unsere menschlichen Kollegen melden sich gleich bei dir!

Table 9: Showcase 1: Short flow, most of the information was extracted from the initial user input. In bold are the extracted entities.

**Student:** Hallo, ich habe eine Frage bzgl. eines <u>Zertifikates</u> und zwar, wie kann ich es <u>austellen</u> lassen?

**Next-Action:** [Organisational Question]
**Bot:** Es scheint sich wohl um eine organisatorische Frage zu handeln! Bitte fasse deine Frage in einem kurzen Text zusammen, damit ich sie an meinen menschlichen Kollegen weiterleiten kann!

**Student:** Wie bekomme ich ein Zertifikat?

**Next-Action:** [Human Hand-Over]
**Bot:** Vielen Dank, unsere menschlichen Kollegen melden sich gleich bei dir!

Table 10: Showcase 2: Organisational Question. Underlined are the keywords which point on the organisational intent.

**Student:** hallo

**Next-Action:** [UNK]
**Bot:** Hast du eine Frage zu einer Aufgabe (MATH), zu einem Text im Kurs (TEXT) oder eine organisatorische Frage (ORG)?

**Student:** ma

**Next-Action:** [UNK]
**Bot:** Hast du eine Frage zu einer Aufgabe (MATH), zu einem Text im Kurs (TEXT) oder eine organisatorische Frage (ORG)?

**Student:** math

**Next-Action:** [Ask for Topic]
**Bot:** Welches Kapitel bearbeitest du gerade?

**Student:** Lineare Gleichungssysteme

**Next-Action:** [Ask for Examination Mode]
**Bot:** Bearbeitest du eine Übungsaufgabe, eine Trainingsaufgabe, einen Quiz oder eine Schlussprüfung?

**Student: human**

**Next-Action:** [Human Hand-Over]
**Bot:** Vielen Dank, unsere menschlichen Kollegen melden sich gleich bei dir!

Table 11: Showcase 3: Manual *intent classification*, *Fallback* and *Human Request* policies (in bold type).

**Student:** Halli-Hallo, könnte mir bitte jemand die Regel 7.2 (vertikale streckung und stauchung des graphen) erklären? Danke im Voraus!

**Next-Action:** [Contextual Question]
**Bot:** Es scheint sich wohl um eine inhaltliche Frage zu handeln! Bitte fasse deine Frage in einem kurzen Text zusammen, damit ich sie an meinen menschlichen Kollegen weiterleiten kann!

**Student:** Ich habe Schwierigkeiten beim Verstehen der Regel 7.2. Was ist genau die vertikale Streckung des Graphen?

**Next-Action:** [Human Hand-Over]
**Bot:** Vielen Dank, unsere menschlichen Kollegen melden sich gleich bei dir!

Table 12: Showcase 4: Contextual Question. Underlined are the key-words which point on the contextual intent.

**Student:** Hallo, ich kann beim **Kapitel 1** nicht weiterkommen, **Trainingsaufgabe 1 (a)**

**Next-Action:** [Ask for Level]
**Bot:** Geht es um eine Aufgabe auf Kapitel-Ebene (z.B um eine Trainingsaufgabe im Kapitel Geometrie) oder um eine Aufgabe auf der Ebene eines Abschnitts in einem Kapitel (z.B. um eine Aufgabe zum Abschnitt Winkel im Kapitel Geometrie)? Antworte bitte mit KAP für Kapitel und SEK für Abschnitt.

**Student:** ich glaube es ist eine **Sektion**

**Next-Action:** [Ask for Sub-Topic]
**Bot:** Wie heißt der Abschnitt in dem sich die Aufgabe befindet, auf die sich deine Frage bezieht?

**Student:** ich bearbeite **rechenregeln und potenzen**

**Next-Action:** [Final Statement]
**Bot:** Deine Angaben waren wie folgt:
a) Kapitel: I Elementares Rechnen,
b) Abschnitt: Rechenregel für Potenzen,
c) Aufgabentyp: Quiz,
d) Aufgabe: 1 (a),
e) Ebene: Abschnitt.
Habe ich dich richtig verstanden? Bitte antworte mit Ja oder Nein.

**Student:** nope

**Next-Action:** [Ask_For]
**Bot:** Welche der folgenden Punkten habe ich falsch erkannt? Schreibe bitte welche der folgenden Punkte es sind: a, b, c, d, e?

**Student:** c

**State-Transition:** [Correct-Fields]
**Bot:** Gib bitte die korrekte Information für c) ein:

**Student:** Training

**State-Transition:** [Final Statement]
**Bot:** Deine Angaben waren wie folgt:
a) Kapitel: I Elementares Rechnen,
b) Abschnitt: Rechenregel für Potenzen,
**c) Aufgabentyp: Training,**
d) Aufgabe: 1 (a),
e) Ebene: Abschnitt.
Habe ich dich richtig verstanden? Bitte antworte mit Ja oder Nein.

**Student:** yes

**State-Transition:** [Human Hand-Over]
**Bot:** Vielen Dank, unsere menschlichen Kollegen melden sich gleich bei dir!

Table 13: Showcase 5: Long Flow. Correction of entries. In bold type are the extracted entities.

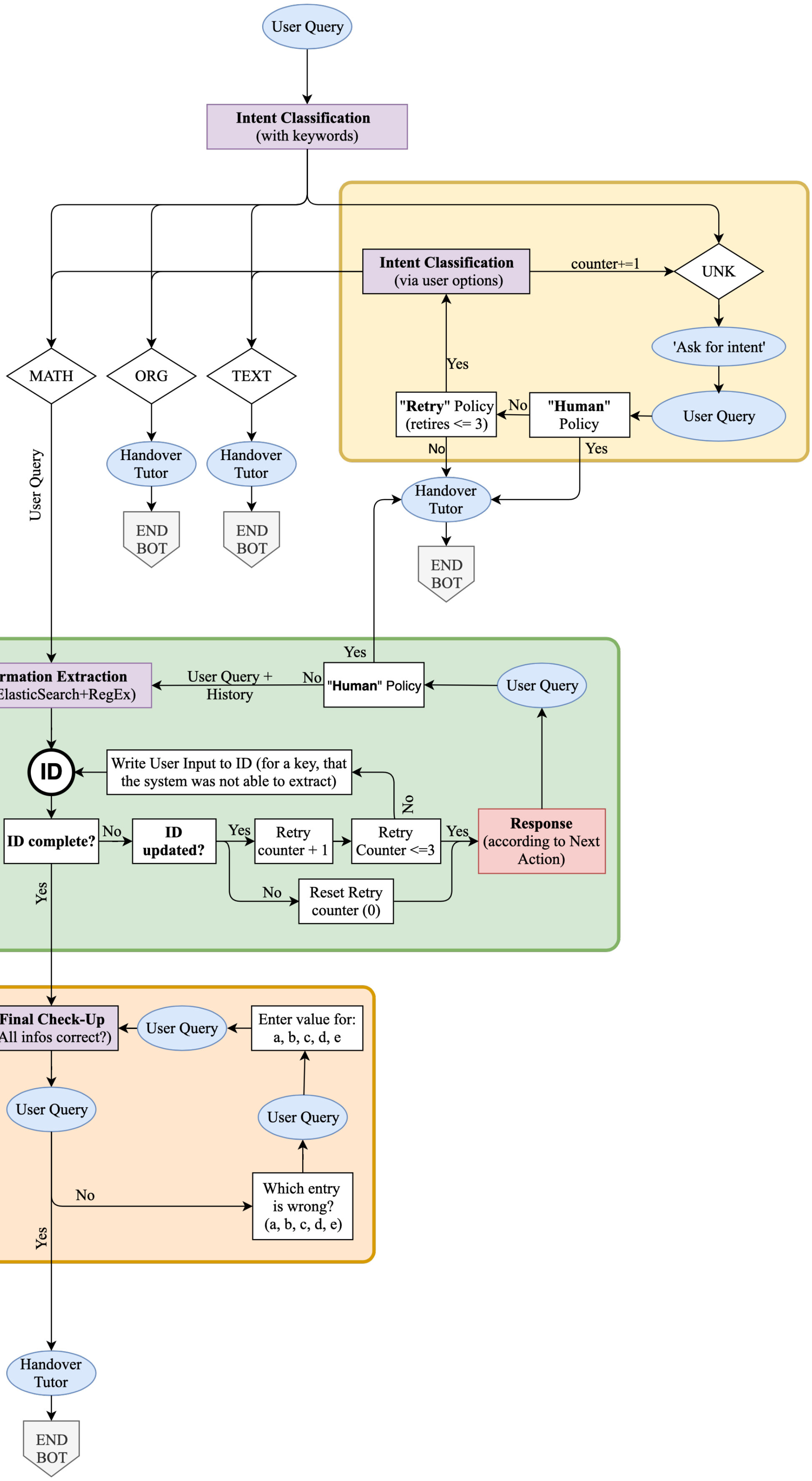


Figure 4: Dialogue Flow. Abbreviations: UNK - unknown; ID - informational dictionary; RegEx - regular expressions.

## 3.3 EVALUATION

In order to get the feedback on the quality, functionality, and usefulness of the proposed model, we evaluated it in two steps: first, with an automated method (Section 3.3.1) utilizing 130 manually annotated conversations, to verify the robustness of the system. Second, with the help of human tutors (Section 3.3.2) from OMB+, to examine the user experience. We describe the details as well as the most common errors below.

### 3.3.1 *Automated Evaluation*

To manage an automated evaluation, we manually assembled a dataset with 130 conversational flows. We predefined viable initial user questions, user answers, as well as gold system responses. These flows cover most *frequent dialogues* that we previously observed in human-to-human dialogue data. We examined our system by implementing a *self-chatting evaluation bot*. The evaluation cycle is as follows:

- The system accepts the first question from a predefined dialogue via an API request, preprocesses, and analyzes it to determine the intent and extract entities.
- Then it estimates the appropriate next action and responds accordingly.
- This response is then compared to the systems gold answer: If the predicted result is correct, then the system receives the next predefined user input from the templated dialogue and responds again, as defined above. This procedure proceeds until the dialogue terminates (i.e., ID is complete). Otherwise, the system reports an unsuccessful case number.

The system successfully passed this evaluation for all 130 cases.

### 3.3.2 *Human Evaluation & Error Analysis*

To evaluate the system on irregular cases, we carried out experiments with human tutors from the OMB+ platform. The tutors are experienced regarding rare or complex issues, ambiguous responses, misspellings, and other infrequent but still relevant problems, which occur during a natural language dialogue. In the following, we review some common errors and describe additional observations.

MISSPELLINGS AND CONFUSABLE SPELLING frequently befall in the user-generated text. Since we attempt to let the conversation remain natural to provide better user experience, we do not instruct students for formal writing. Therefore, we have to account for various writing issues. One of the prevalent problems are misspellings. German words are commonly long and can be complex, and because users often type quickly, it can lead to the wrong order of characters within a given word. To tackle this challenge, we utilized *fuzzy_match* within ES. Though, the maximum allowed edit distance in ElasticSearch (ES) is set to 2 characters, which means, that all the misspellings beyond this threshold could not be correctly identified by ES (e.g., *Differenzialrechnung* vs *Differnetialrechnung*). Another representative example would be the formulation of the section or question number. The equivalent information can be communicated in several discrete ways, which has to be considered by RegEx unit (e.g., *Aufgabe 5 a*, *Aufgabe V a*, *Aufgabe 5 (a)*). A similar problem occurs with confusable spelling (i.e.: *Differenzialrechnung* vs *Differenzialgleichung*).

We analyzed the cases stated above and improved our model by adding some of the most common misspellings to the ElasticSearch (ES) database or handling them with RegEx during the preprocessing step.

ELASTICSEARCH THRESHOLD: We observed both cases where the system failed to extract information, although the user provided it and examples where ES extracted information not mentioned in a user query. According to our understanding, these errors occur due to the *relevancy_scoring* algorithm of ElasticSearch (ES), where a document's score is a combination of textual similarity and other metadata based scores. Our examination revealed that ES mostly fails to extract the information if the user message is rather short (e.g., about 5 tokens). To overcome this difficulty, we coupled the current input $u_t$ with the dialogue history ($h = u_{t,\cdots,t-2}$). That eliminated the problem and improved the retrieval quality.

To solve the opposite problem where ES extracts false information (or information that was not mentioned in a query) was more challenging. We learned that the problem comes from short words or sub-words (i.e., suffix, prefix), which ES locates in a database and considers them credible enough. The ES documentation suggests getting rid of stop words to eliminate this behavior. However, this did not improve the search. Also, fine-tuning of ES parameters such as the *relevance_threshold*, *prefix_length*[5] and *minimum_should_match*[6] parameter did not result in notable improvements. To cope with this problem, we implemented a *verification step*, where a user can edit the erroneously retrieved data.

[5]The amount of fuzzified characters. This parameter helps to decrease the number of terms which must be reviewed.

[6]Indicates a number of terms that must match for a document to be considered relevant.

## 3.4 STRUCTURED DIALOGUE ACQUISITION

As we stated, the implemented system not only supports the human tutor by assisting students but also collects structured and labeled training data in the background. In a *trial run* of the rule-based system, we accumulated a *toy-dataset* with dialogues. The assembled dataset includes the following:

- Plain dialogues with unique dialogue-indexes;
- Plain ID information collected for the whole conversation;
- Pairs of questions (i.e., user requests) and responses (i.e., bot replies) with the unique dialogue- and turn-indexes;
- Triples in the form of *(Question, Next Action, Response)*. Information on the next system's action could be employed to train a DM unit with (deep-) machine learning algorithms;
- On every conversational state, we keep the entities, that the system was able to extract, along with their position in the utterance. This information could be used to train a custom domain-specific NER model.

## 3.5 SUMMARY

In this work, we realized a conversational agent for IPA purposes that addresses two problems: First, it decreases repetitive and time-consuming activities, which allows workers of the e-learning platform to focus solely on mathematical and hence cognitively demanding questions. Second, by interacting with users, it augments the resources with structured and labeled training data. The latter can be utilized for further implementation of learnable dialogue components.

The realization of such a system was connected with multiple challenges. Among others were missing structured conversational data, ambiguous or erroneous user-generated text, and the necessity to deal with existing corporate tools and their design.

The proposed conversational agent enabled us to accumulate structured, labeled data without any special efforts from the human (i.e., tutors) side (e.g., manual annotation and post-processing of existing data, change of the conversational structure within the tutoring process). Once we collected structured dialogues, we were able to re-train particular segments of the system with deep learning methods and achieved consistent performance for both of the proposed tasks. We report the re-implemented elements in Chapter 4.

At the time of writing this thesis, the system described in this work was deployed at the OMB+ platform.

## 3.6 FUTURE WORK

Possible extensions of our work are:

- In this work, we implemented a rule-based dialogue model from scratch and adjusted it for the specific domain (i.e., e-learning). The core of the model – is a dialogue manager that determines the current state of the conversation and the possible next action. Rule-based systems are generally considered to be hardly adaptable to new domains; however, our dialogue manager proved to be flexible to slight modifications in a workflow. One of the possible directions of future work would be, thus, the investigation of the *general adaptability* of the dialogue manager core to other scenarios and domains (e.g., different course).
- A further possible extension would we the introduction of *different languages*. The current model is implemented for the German language, but the OMB+ platform provides this course also in English and Chinese. Therefore, it would be interesting to investigate whether it is possible to adjust the existing system to other languages without drastic changes in the model?

# 4 DEEP LEARNING RE-IMPLEMENTATION OF UNITS

> This chapter partially covers work already published at international peer-reviewed conferences. The relevant publication is Moiseeva et al., 2020. The research described in this chapter was carried out in its entirety by the author of this thesis. The other authors of the publication acted as advisors.

In this chapter, we address the challenge of re-implementation of two central components of our IPA dialogue system described in Chapter 3, by employing deep learning techniques. Since the data for the training was collected in a (short) trial-run of the rule-based system, it is not sufficient to train a neural network from scratch. Therefore, we utilize Transfer Learning (TL) methods and employ the previously pre-trained Bidirectional Encoder Representations from Transformers (BERT) model that we fine-tune for two of our custom tasks.

## 4.1 INTRODUCTION

Data mining and machine learning technologies have gained notable advancement in many research fields, including but not limited to computer vision, robotics, advanced analytics, and computational linguistics (Pan and Yang, 2009; Trautmann et al., 2019; Werner et al., 2015). Besides that, in recent years, deep neural networks with their ability to accurately map from inputs to outputs (i.e., labels, images, sentences) while analyzing patterns in data, became a potential solution for many industrial automatization tasks (e.g., fake news detection, sentiment analysis).

Nevertheless, conventional supervised models still have limitations when applied to real-world data and industrial tasks. The first challenge here is a training phase since a robust model requires an immense amount of structured and labeled data. Still, there are just a few cases where such data is publicly available (e.g., research datasets), but for most of the tasks, domains and languages, the data has to be gathered over a long time. The second hurdle is that if trained on existing data from a distinct domain, a supervised model cannot generalize to conditions that are different from those faced in the training dataset. Most of the machine learning methods perform correctly only under a general assumption: the training and test data are mapped to the same feature space and the same distribution (Pan and Yang, 2009). When the distribution changes, most statistical models need to be retrained from scratch using newly obtained training samples. It is especially crucial if applying such approaches to real-world data that is usually very noisy and contains a large number of scenarios. In this case, a model would be ill-trained to make decisions about novel patterns. Furthermore, for most of the real-world applications, it is expensive or even not feasible at all to recollect the required training data to retrain the model. Hence, the possibility to transfer the knowledge obtained during the training on existing data to the new unseen domain is one of the possible solutions for industrial problems. Such a technique is called Transfer Learning (TL), and it allows dealing with the abovementioned scenarios by leveraging existing labeled data of some related tasks or domains.

### 4.1.1 *Outline and Contributions*

This work addresses the challenge of re-implementing two out of three central components in our rule-based IPA conversational assistant with deep learning methods. As we discussed in Chapter 2, hybrid dialogue models (i.e., consisting of both rule-based and trainable components) appear to replace the established rule- and template-based methods, which are, to the most part, employed in the industrial setting. To investigate this assumption and examine current state-of-the-art deep learning techniques, we conducted experiments on our custom domain-specific data. Since the available data were collected in a trial-run and the obtained dataset was relatively small to train a conventional machine learning model from scratch, we utilized the Transfer Learning (TL) approach, and fine-tuned the existing pre-trained model for our target domain.

For the experiments, we defined two tasks:

- First, we considered the NER problem in a custom domain setting. We defined a *sequence labeling* task and employed a Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) as one of the transfer learning methods widely utilized in the NLP area. We applied the model on our dataset and fine-tuned it for *seven* (7) domain-specific (i.e., e-learning) entities.
- Second, we examined the effectiveness of transfer learning for the dialogue manager core. The DM is one of the fundamental parts of the conversational agent and requires compelling learning techniques to hold the conversation intelligently. For that experiment, we defined a *classification* task and applied the BERT model to predict the systems `next action` for every given user input in a conversation. We then computed the macro F1-score for 14 possible classes (i.e., actions) and an average dialogue accuracy.

The main characteristics of both settings are:

- the set of named entities and labels (i.e., actions) between source and target domain is different;
- for both settings we utilized the BERT model in `German` and `multilingual` versions with additional parameters;
- the dataset used to train the target model is small and consists of highly domain-specific labels. The latter is often appearing in industrial scenarios.

We finally verified that the selected method performed reasonably well on both tasks: We reached the performance of 0.93 macro F1 points for Named Entity Recognition (NER) and 0.75 macro F1 points for the Next Action Prediction (NAP) task. Therefore, we conclude that both NER and NAP components could be employed to substitute or extend the existing rule-based modules.

## 4.2 RELATED WORK

As discussed in the previous section, the main benefit of Transfer Learning (TL) is that it allows the domains and tasks used during training and testing phases to be different. Initial steps in the formulation of transfer learning were made after *NeurIPS*-95[1] workshop that was focused on the demand for lifelong machine-learning methods that preserve and reuse previously learned knowledge (Chen et al., 2018). Research on transfer learning has attracted more and more attention since then, especially in the computer vision domain and, in particular, for image recognition tasks (Donahue et al., 2014; Sharif Razavian et al., 2014).

Recent studies have shown that TL can also be successfully applied within the Natural Language Processing (NLP) domain, especially on semantically equivalent tasks (Dai et al., 2019; Devlin et al., 2018; Mou et al., 2016). Several research works were carried out specifically on the Named Entity Recognition (NER) task and explored the capabilities of transfer learning when applied to different named entity categories (i.e., different output spaces). For instance, Qu et al. (2016) pre-trained a linear-chain Conditional Random Fields (CRF) on a large amount of labeled data in the source domain. Then, they introduced a two linear layer neural network that learns the difference between the source and target label distributions. Finally, the authors initialized another CRF with parameters learned in linear layers, to predict the labels of the target domain. Kim et al. (2015) conducted experiments with transferring features and model parameters between related domains, where the label classes are different but may have a semantic similarity. Their primary approach was to construct label embeddings to automatically map the source and target label classes to improve the transfer (Chen et al., 2018).

[1] Conference on Neural Information Processing Systems. Formerly NIPS.

However, there are also models trained in a manner such that they could be applied to multiple NLP tasks simultaneously. Among such architectures are `ULMFit` (Howard and Ruder, 2018), BERT (Devlin et al., 2018), `GPT2` (Radford et al., 2019) and `TransformerXL` (Dai et al., 2019) – powerful models that were recently proposed for transfer learning within the NLP domain. These architectures are called *language models* since they attempt to learn language structure from large textual collections in an unsupervised manner, and then predict the next word in a sequence based on previously observed context words. The Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) model is one of the most popular architecture among the abovementioned, and it is also one of the first that showed a close to human-level performance on the General Language Understanding Evaluation (GLUE) benchmark[2] (Wang et al., 2018), that includes tasks on sentiment analysis, question answering, semantic similarity, and language inference. The architecture of BERT consists of stacked transformer blocks that are pre-trained on a large corpus consisting of 800M words from modern English books (i.e., BooksCorpus by Zhu et al.), and $2,500$M words of text from pre-processed (i.e., without markup) English Wikipedia articles. Overall, BERT advanced the state-of-the-art performance for *eleven* (11) NLP tasks. We describe this approach in detail in the subsequent section.

## 4.3 BERT: BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

Bidirectional Encoder Representations from Transformers (BERT) architecture is divided into two steps: *pre-training* and *fine-tuning*. The pre-training step was enabled through two following tasks:

- **Masked Language Model** – the idea behind this approach is to train a deep bidirectional representation that is different from conventional language models, which are either trained left-to-right or right-to-left. To train a bidirectional model, the authors mask out a certain number of tokens in a sequence. The model is then asked to predict the original words for masked tokens. It is essential that in contrast to denoising auto-encoders (Vincent et al., 2008), the model does not need to reconstruct the entire input; instead, it attempts to predict the masked words. Since the model does not know which words exactly it will be asked about, it learns a representation for every token in a sequence (Devlin et al., 2018).
- **Next Sentence Prediction** – aims to capture relationships between two sentences $A$ and $B$. In order to train the model, the authors pre-trained a binarized next sentence prediction task, where pairs of sequences were labeled according to the criteria: $A$ and $B$ either follow each other directly in the corpus (label

[2] https://gluebenchmark.com

IsNext) or are both taken from random places (label NotNext). The model is then asked to predict whether the former or the latter is the case. The authors demonstrated that pre-training towards this task is extremely useful for both *question answering* and *natural language inference* tasks (Devlin et al., 2018).

Furthermore, BERT uses WordPiece tokenization for embeddings (Wu et al., 2016), which segments words into subword-level tokens. This approach allows the model to make additional inferences based on word structure (i.e., -ing ending denotes similar grammatical categories).

The *fine-tuning* step follows the pre-training process. For that, the BERT model is initialized with parameters obtained during the pre-training step, and a task-specific fully-connected layer is used after transformer blocks, which maps the general representations to custom labels (employing softmax operation).

SELF-ATTENTION: The key operation of BERT architecture is the *self-attention*, which is a sequence-to-sequence operation with conventional encoder-decoder structure (Vaswani et al., 2017). The encoder maps an input sequence $(x_1, x_2 \ldots, x_n)$ to a sequence of continuous representations $z = (z_1, z_2, \ldots, z_n)$. Given $z$, the decoder then generates an output sequence $(y_1, y_2, \ldots, y_m)$ of symbols one element at a time (Vaswani et al., 2017). To produce output vector $y_i$ the self attention operation takes a weighted average over all input vectors:

$$y_i = \sum_j w_{ij} x_j, \tag{16}$$

where $j$ is an index over the entire sequence and the weights sum to one (1) over all $j$. The weight $w_{ij}$ is derived from a dot product function over $x_i$ and $x_j$ (Bloem, 2019; Vaswani et al., 2017):

$$w'_{ij} = x_i^T x_j \tag{17}$$

The dot product returns a value between negative and positive infinity, and a softmax maps these values to $[0, 1]$ to ensure that they sum to one (1) over the entire sequence (Bloem, 2019; Vaswani et al., 2017):

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}} \tag{18}$$

A self-attention is not the only operation in transformers, but is the essential one, since it propagates information between vectors. Other operations in the transformer are applied to every vector in the input sequence without interactions between vectors (Bloem, 2019; Vaswani et al., 2017).

## 4.4 DATA & DESCRIPTIVE STATISTICS

The benchmark that we accumulated through the trial-run includes 300 structured and partially labeled dialogues with the average length of dialogue being *six* (6) utterances. All the conversations were held in the German language. Detailed general statistics can be found in Table 14.

| | **Value** |
|---|---|
| Max. Len. Dialogue (in utterances) | 15 |
| Avg. Len. Dialogue (in utterances) | 6 |
| Max. Len. Utterance (in tokens) | 100 |
| Avg. Len. Utterance (in tokens) | 9 |
| # Overall Unique Action Labels | 13 |
| # Overall Unique Entity Labels | 7 |
| **Train** – # Dialogues (# Utterances) | 200 (1161) |
| **Eval** – # Dialogues (# Utterances) | 50 (279) |
| **Test** – # Dialogues (# Utterances) | 50 (300) |

Table 14: General statistics for conversational dataset.

The dataset covers 14 unique *next actions* and seven (7) unique *named entities*. Detailed information on next actions and entities can be found in Table 15 and Table 16. Below we listed possible actions with the corresponding explanation and predefined mapped statements:

UNK – requests to define the intent of the question: mathematical, organizational or contextual. (i.e., fallback policy in case if the automated intent recognition unit fails.) → *"Hast du eine Frage zu einer Aufgabe (MATH), zu einem Text im Kurs (TEXT) oder eine organisatorische Frage (ORG)?"*

ORG – handovers discussion to a human tutor in case of organizational question. → *"Es scheint sich wohl um eine organisatorische Frage zu handeln! Bitte fasse deine Frage in einem kurzen Text zusammen, damit ich sie an meinen menschlichen Kollegen weiterleiten kann"*

TEXT – handovers discussion to a human tutor in case of contextual question. → *"Es scheint sich wohl um eine inhaltliche Frage zu handeln! Bitte fasse deine Frage in einem kurzen Text zusammen, damit ich sie an meinen menschlichen Kollegen weiterleiten kann."*

TOPIC – requests information on the topic. → *"Welches Kapitel bearbeitest du gerade? Antworte bitte mit der Kapitelnummer z.B.* IA, IB, II, *... oder mit dem Kapitelnamen z.B. Geometrie."*

EXAMINATION – requests information about the examination mode.→ *"Bearbeitest du eine Übungsaufgabe, eine Trainingsaufgabe, einen Quiz oder eine Schlussprüfung?"*

SUBTOPIC – requests information about the subtopic in case of quiz or section-level training/exercise modes. → *"Wie heißt der Abschnitt in dem sich die Aufgabe befindet, auf die sich deine Frage bezieht?"*

LEVEL – requests information about the level of examination mode: chapter or section? → *"Geht es um eine Aufgabe auf Kapitel-Ebene (z.B um eine Trainingsaufgabe im Kapitel Geometrie) oder um eine Aufgabe auf der Ebene eines Abschnitts in einem Kapitel (z.B. um eine Aufgabe zum Abschnitt Winkel im Kapitel Geometrie)? Antworte bitte mit KAP für Kapitel und SEK für Abschnitt."*

QUESTION NUMBER – requests information about the question number. → *"Welche Aufgabe bearbeitest du gerade? Wenn du z.B. bei Teilaufgabe* c *in der zweiten Trainingsaufgabe (oder der zweiten Übungsaufgabe) bist, antworte mit* 2c*. Bearbeitest du gerade einen Quiz oder eine Schlussprüfung, gib bitte nur die Teilaufgabe an."*

FINAL REQUEST – considers the Informational Dictionary (ID) to be complete and initiates the verification step. → *"Deine Angaben waren wie folgt .... Habe ich dich richtig verstanden? Bitte antworte mit Ja oder Nein"*

VERIFY REQUEST – requests the student to verify the assembled information. → *"Welche der folgenden Punkte habe ich falsch erkannt? Schreibe bitte welche der folgenden Punkte es sind: [a, b, c, ...][3] ... "*

CORRECT REQUEST – requests the student to provide a correct information, if there is an error in the assembled data. → *"Gib bitte die korrekte Information für [a][4], ein:"*

EXACT QUESTION – requests the student to provide the detailed explanation of the problem. → *"Bitte fasse deine Frage in einem kurzen Text zusammen, damit ich sie an meinen menschlichen Kollegen weiterleiten kann."*

HUMAN HANDOVER – finalizes the conversation. → *"Vielen Dank, unsere menschlichen Kollegen melden sich gleich bei dir!"*

[3] Variables that depend on the state of Informational Dictionary (ID).
[4] Variable that was selected as erroneous and needs to be corrected.

| **Action** | **Counts** | **Action** | **Counts** |
|---|---|---|---|
| Final Request | 321 | Unk. | 80 |
| Human Handover | 300 | Subtopic | 55 |
| Exact Question | 286 | Correct Request | 40 |
| Question Number | 176 | Verify Request | 34 |
| Examination | 175 | Org. | 17 |
| Topic | 137 | Text. | 13 |
| Level | 130 | | |

Table 15: Detailed statistics on possible systems actions. Columns Counts denote the number of occurrences of each action in the entire dataset.

| **Entity** | **Counts** |
|---|---|
| Question Nr. | 317 |
| Chapter | 311 |
| Examination | 303 |
| Subtopic | 198 |
| Level | 80 |
| Intent | 70 |

Table 16: Detailed statistics on possible named entities. Column Counts denote the number of occurrences of each entity in the entire dataset.

## 4.5 NAMED ENTITY RECOGNITION

We defined a sequence labeling task to extract custom entities from user input. We considered seven (7) viable entities (see Table 16) to be recognized by the model: *topic*, *subtopic*, *examination mode* and *level*, *question number*, *intent*, as well as the entity *other* for remaining words in the utterance. Since the data collected from the rule-based system already includes information on the entities, we able to train a domain-specific NER unit. However, since the original user-input was informal, the same information could be provided in different writing styles. That means that a single entity could have different surface forms (e.g., synonyms, writing styles). Therefore, entities that we extracted from the rule-based system were transformed into a universal standard (e.g., official chapter names). To consider all of the variable entity forms while post-labeling the original dataset, we determined generic entity names (e.g., chapter, question nr.) and mapped variations of entities from the user input (e.g., Chapter = [Elementary Calculus, Chapter I, ...]) to them. The overall dataset consists of 300 labeled dialogues, where 200 (with 1161 utterances) were employed for training, and 100 for evaluation and test sets (50 dialogues with ca. 300 utterances for each set respectively).

### 4.5.1 *Model Settings*

We conducted experiments with German and multilingual BERT implementations[5]. The capitalization of words is significant for the German language; therefore, we run the experiments on the *capitalized data* while preserving the *original punctuation*. We employed the available `base model` for both multilingual and German BERT implementations in the `cased` version. We initiated the learning rate for both models to $1e-4$, and the maximum length of the tokenized input was set to 128 tokens. We conducted the experiments multiple times with different seeds for a maximum of 50 epochs, with the training batch size set to 32. We utilized `AdamW` as the optimizer and applied early stopping if the performance did not improve significantly after 5 epochs.

## 4.6 NEXT ACTION PREDICTION

We defined a classification problem, where we aim to predict the system's next action according to the given user input. We assumed 14 custom actions (see Table 15) that we considered being our classes. While collecting the toy dataset, every input was automatically labeled by the rule-based system with the corresponding next action and the dialogue-id. Thus, no additional post-labeling was required on this step.

We investigated two settings for this task:

- **Default Setting:** Utilizing a user input and the corresponding class (i.e., next action) without any supplementary context. By default, we conduct all of our experiments in this setting.
- **Extended Setting:** Employing a user input, corresponding next action, and *previous systems action* as a source of supplementary context. For this setting, we experimented with the best performing model from the default setting.

The overall dataset consists of 300 labeled dialogues, where 200 (with 1161 utterances) were employed for training, and 100 for evaluation and test sets (50 dialogues with ca. 300 utterances for each set respectively).

### 4.6.1 *Model Settings*

For the NAP task, we carried out experiments with German and multilingual BERT implementations as well. We examined the performance of both *capitalized* and *lower-cased* inputs as well as plain and preprocessed data. For the multilingual BERT, we used the `base model` in both `cased` and `uncased` modifications. For the German BERT, we

[5] https://github.com/huggingface/transformers

utilized the `base model` in the `cased` modification only[6]. For both models, we initiated the learning rate to $4e - 5$, and the maximum length of the tokenized input was set to 128 tokens. We conducted the experiments multiple times with different seeds for a maximum of 300 epochs with the training batch size set to 32. We utilized `AdamW` as the optimizer and applied early stopping if the performance did not improve significantly after 15 epochs.

## 4.7 EVALUATION AND RESULTS

To evaluate the models, we computed *word-level* macro F1 score for the Named Entity Recognition (NER) task and *utterance-level* macro F1 score for the Next Action Prediction (NAP) task. The word-level F1 is estimated as the average of the F1 scores per class, each computed from all words in the evaluation and test sets. The outcomes for the NER task are represented in Table 17. For utterance-level F1, a single class label (i.e., next action) is taken for the whole utterance. The results for the NAP task are shown in Table 18.

We additionally computed *average dialogue accuracy* for the best performing NAP models. This score indicates how well the predicted next actions compose the conversational flow. The average dialogue accuracy was estimated for 50 conversations in the evaluation and test sets, respectively. The results are displayed in Table 19.

The obtained results for the NER task revealed that German BERT performed significantly better than the multilingual BERT model. The performance of the custom NER unit is at 0.93 macro F1 points for all possible named entities (see Table 17). In contrast, for the NAP task, the multilingual BERT model obtained better performance than the German BERT model. The best performing method in the default setting gained a macro F1 of 0.677 points for 14 possible classes, whereas the model in the extended setting performed better – its best macro F1 score is 0.752 for the equal number of classes (see Table 18).

Regarding the dialogue accuracy, the extended system fine-tuned with multilingual BERT obtained better outcomes as the default one (see Table 19). The overall conclusion for the NAP is that the capitalized setting increased the performance of the model, whereas the inclusion of punctuation has not influenced the results.

[6] Uncased pre-trained modification of model was not available.

| Task | Model | Cased | Punct. | F1 | |
|---|---|---|---|---|---|
| | | | | Eval | Test |
| NER | GER | ✓ | ✓ | **0.971** | **0.930** |
| NER | Mult | ✓ | ✓ | 0.926 | 0.905 |

Table 17: Word-level F1 for the Named Entity Recognition (NER) task. In bold: best performance for evaluation and test sets.

| Task | Model | Cased | Punct. | Extended | F1 | |
|---|---|---|---|---|---|---|
| | | | | | Eval | Test |
| NAP | GER | ✓ | × | × | 0.711 | 0.673 |
| NAP | GER | ✓ | ✓ | × | 0.701 | 0.606 |
| NAP | Mult | ✓ | ✓ | × | 0.688 | 0.625 |
| NAP | Mult | ✓ | × | × | <u>0.769</u> | <u>0.677</u> |
| NAP | Mult | ✓ | × | ✓ | **0.810** | **0.752** |
| NAP | Mult | × | ✓ | × | 0.664 | 0.596 |
| NAP | Mult | × | × | × | 0.742 | 0.502 |

Table 18: Utterance-level F1 for the Next Action Prediction (NAP) task. Underlined: best performance for evaluation and test sets for default setting (without previous action context). In bold: best performance for evaluation and test sets on extended setting (with previous action context).

| Model | Accuracy | |
|---|---|---|
| | Eval | Test |
| **NAP default** | 0.765 | 0.724 |
| **NAP extended** | **0.813** | **0.801** |

Table 19: Average dialogue accuracy computed for the Next Action Prediction (NAP) task for best preforming models. In bold: best performance for evaluation and test sets.

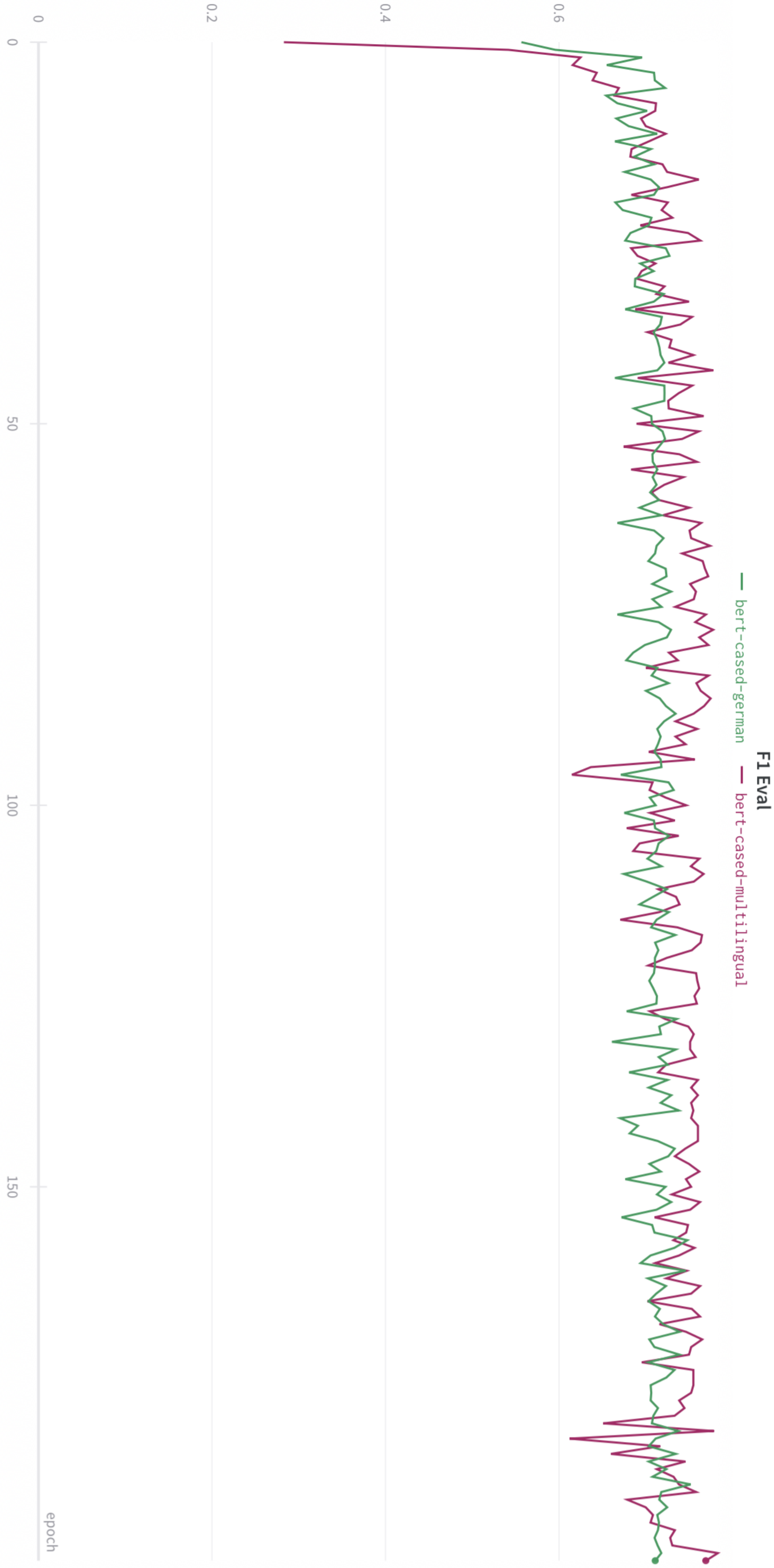


Figure 5: Macro F1 evaluation. NAP – default model. Comparison between German and multilingual BERT.

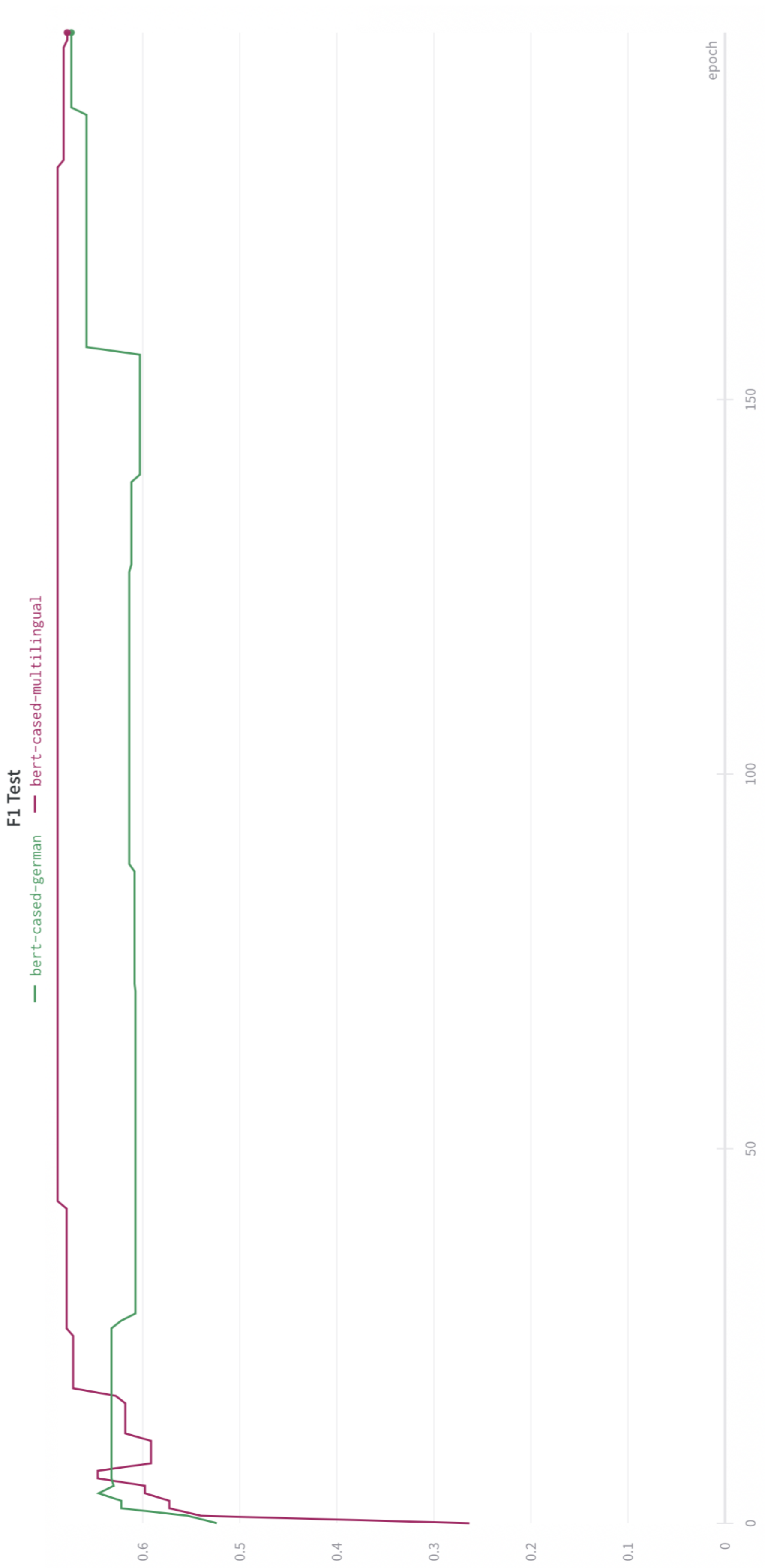


Figure 6: Macro F1 test. NAP – default model. Comparison between German and multilingual BERT.

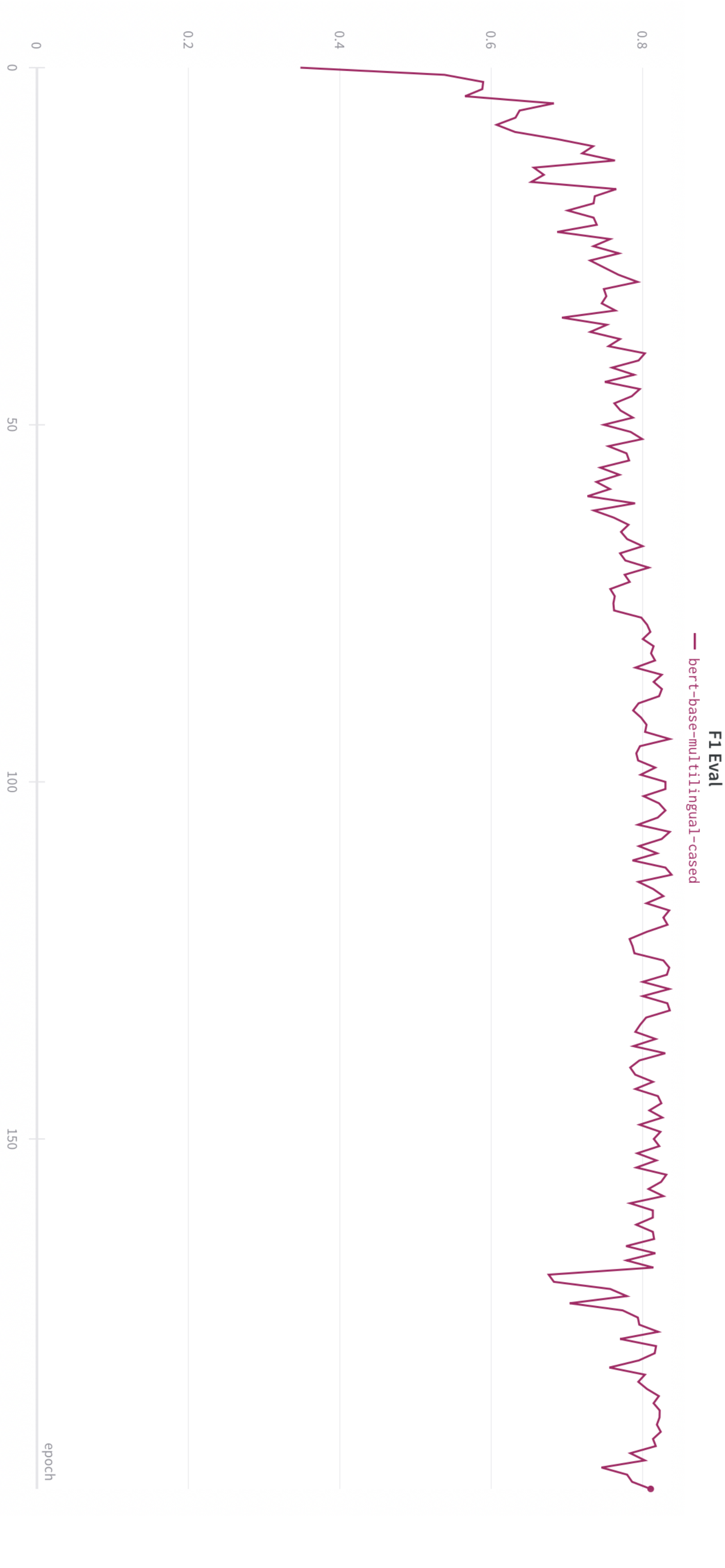


Figure 7: Macro F1 evaluation. NAP – extended model.

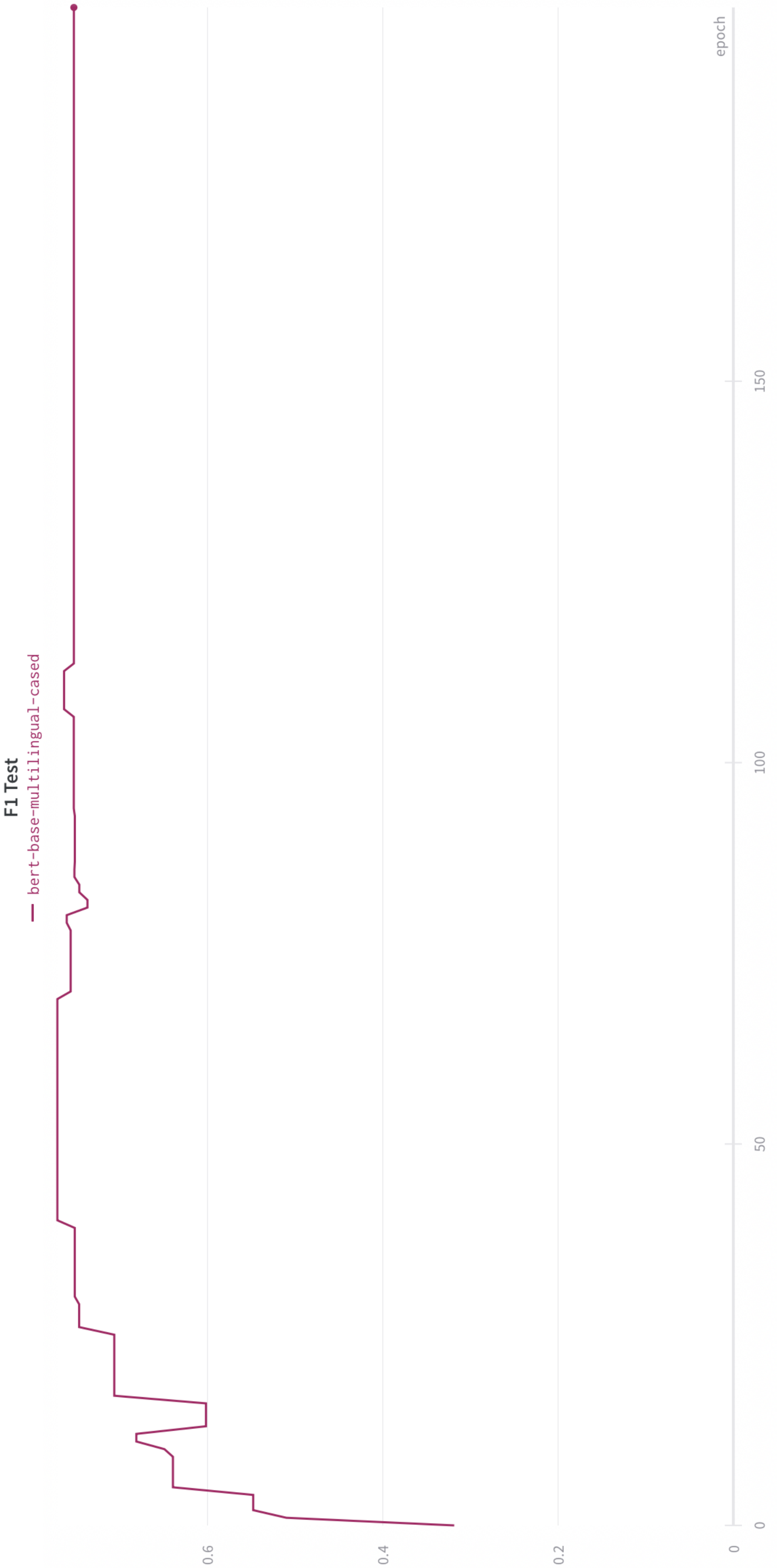


Figure 8: Macro F1 test. NAP – extended model.


Figure 9: Macro F1 evaluation. NER – Comparison between German and multilingual BERT.

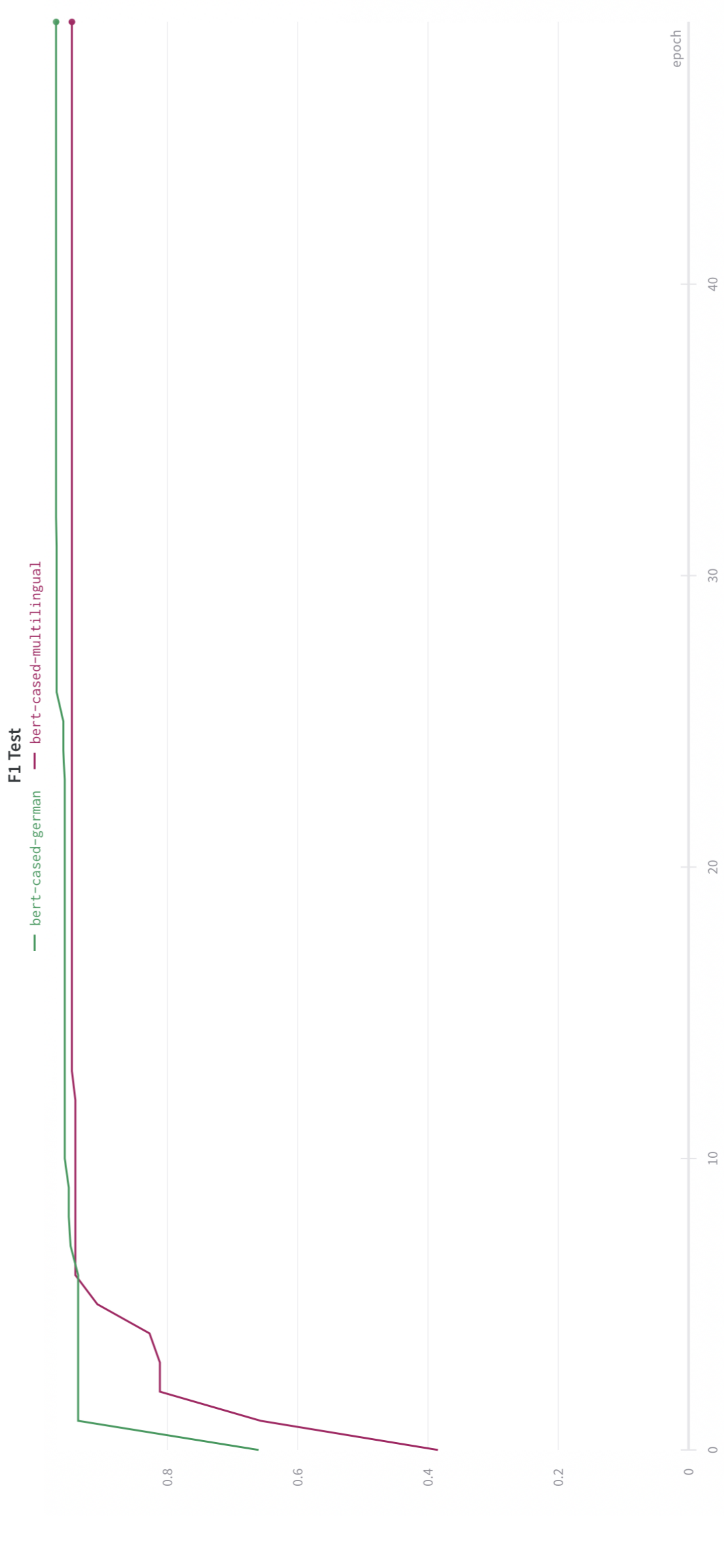


Figure 10: Macro F1 test. NER – Comparison between German and multilingual BERT.

## 4.8 ERROR ANALYSIS:

We then investigated the cases, where models failed to predict the correct action or labeled the named entity span erroneously. Below we outline the most common errors:

NEXT ACTION PREDICTION: One of the prevalent flaws in the *default model* was the mismatch between two consecutive actions – the action `question number` and `subtopic`. That is due to the position of these actions in the conversational flow. The occurrence of both actions in the dialogue is not strict and largely depends on the previous action. To explain this, we illustrate the following possible conversational scenarios:

- The system can either directly proceed with the request for the question number if the examination mode is the final examination.
- If the examination mode is training or exercise, the system has to ask about the level first (i.e., chapter or section).
- If it is a chapter-level, the system can again directly proceed with the request for the question number.
- In the case of section-level, the system has to ask about the subsection first (if this information was not provided before).

The examination of the *extended model* showed that the induction of supplementary context (i.e., previous action) improved the performance of the system by about 60%.

NAMED ENTITY RECOGNITION: The cases where the system failed cover mismatches between the labels `chapter` and `other`, and the labels `question number` and `other`. This failure arose due to the imperfectly labeled span of a multi-word named entity (e.g., *"Elementary Calculus: Proportionality, Percentage"*). The first or last words in the named entity were excluded from the span and erroneously labeled with the `other` label.

## 4.9 SUMMARY

In this chapter, we re-implemented two of three fundamental units of our IPA conversational agent – Named Entity Recognition (NER) and Dialogue Manager (DM) – using the methods of Transfer Learning (TL), specifically, the Bidirectional Encoder Representations from Transformers (BERT) model.

We believe that the obtained results are reasonably high, considering a comparatively little amount of training data we employed to fine-tune the models. Therefore, we conclude that both NAP and NER segments could be used to replace or extend the existing rule-based modules. Rule-based components, as well as ElasticSearch (ES), are limited in their capacities and not flexible to novel patterns. Whereas the trainable units generalize better, and could possibly decrease the amount of inaccurate predictions in case of accidental dialogue behavior. Furthermore, to increase the overall robustness, rule-based and trainable components could be used synchronously: when one system fails (e.g., ElasticSearch (ES) module can not extract an entity), the conversation continues on the prediction obtained from the other model.

## 4.10 FUTURE WORK

There are possible directions for future work:

- At the time of writing of this thesis, both units were trained and investigated on its own – that means, without being incorporated into the initial IPA conversational system. Thus, one of the potential directions of future work would be the embodiment of these units into the original system, and the examination of possible failure cases. Furthermore, the resulting hybrid system could then imply the constraint of additional meta policies to prevent an unexpected systems behavior.
- Further investigation could be towards the multi-language modality for the re-implemented units. Since the Online Mathematik Brückenkurs (OMB+) platform also supports English and Chinese, it would be interesting to examine whether the simple translation from target language (i.e., English, Chinese) to source language (i.e., German) would be sufficient to employ already-assembled dataset and pre-trained units. Presumably, it should be achievable in the case of the Next Action Prediction (NAP) unit but could lead to a noticeable performance drop for the Named Entity Recognition (NER) task, since there could be a considerable gap between translated and original entities spans.
- Last but not least, one of the further extensions for the IPA would be the eventual applicability for more cognitively demanding tasks. For instance, it is of interest to extend the model for more complex domains like handling of mathematical questions. Would it be possible to automate the human assistance at least by less complicated mathematical inquiries, or is it still unachievable with currently existing NLP techniques and machine learning methods?

# Part II

# CLUSTERING-BASED TREND ANALYZER

# 5 FOUNDATIONS

As discussed in Chapter 1, predictive analytics is one of the potential Intelligent Process Automation (IPA) tasks due to its capability to forecast future events by using current and historical data and therethrough assist knowledge workers with, for instance, sales or investments. Modern machine learning algorithms allow an intelligent and fast analysis of large amounts of data. However, the evaluation of such algorithms and models remains one of the grave problems in this domain. This is mostly due to the lack of publicly available *benchmarks* for the evaluation of topic modeling and trend detection algorithms. Hence, in this part of the thesis, we are addressing this challenge and assembling the benchmark for detecting both *trends* and *downtrends* (i.e., topics that become less frequent overtime). To the best of our knowledge, the task of downtrend detection has not been well addressed before.

In this chapter, we introduce the concepts required by the second part of the thesis. We begin with the enlightenment to the topic modeling task and define the emerging trend in Section 5.1. Next, we examine the related work, existing approaches to topic modeling and trend detection, and their limitations in Section 5.2. In Chapter 6, we introduce our TRENDNERT benchmark for trend and downtrend detection and describe the method we employed for its assembly.

## 5.1 MOTIVATION AND CHALLENGES

Science changes and evolves rapidly: novel research areas emerge, while others fade away. Keeping pace with these changes is challenging. Therefore, recognizing and forecasting *emerging research trends* is of significant importance for researchers, academic publishers, as well as for funding agencies, stakeholders, and innovation companies.

Previously, the task of trend detection was mostly solved by domain experts who used specialized and cumbersome tools to investigate the data to get useful insights from it (e.g., through data visualization or statistics). However, manual analysis of large amounts of data could be time-consuming, and hiring domain experts is very costly. Furthermore, the overall increase of research data (i.e., Google Scholar, PubMed, DBLP) in the past decade makes the approach based on human domain experts less and less scalable. In contrast, automated approaches become a more desirable solution for the task of emerging trend identification (Kontostathis et al., 2004; Salatino, 2015). Due to a large number of electronic documents appearing on the web daily, several machine learning techniques were developed to find patterns of word occurrences in those documents using hierarchical probabilistic models. These approaches are called – topic models, and they allow the analysis of extensive textual collections and valuable insights from them. The main advantage of topic models is their ability to discover hidden patterns of word-use and connecting documents containing similar patterns. In other words, the idea of a topic model is that documents are a mixture of topics, where a topic is defined as a probability distribution over words (Alghamdi and Alfalqi, 2015).

Topics have a property of being able to evolve; thus, modeling topics without considering time may confuse the precise topic identification. Whereas modeling topics by considering time is called *topic evolution modeling*, and it can reveal important hidden information in the document collection, allowing recognizing topics with the appearance of time. This approach also provides the ability to check the evolution of topics during a given time window and examine whether a given topic is gaining interest and popularity over time (i.e., becoming an *emerging trend*) or if its development is stable. Considering the latter approach, we turn to the task of *emerging trend identification*.

How is an *emerging trend* defined? An emerging trend is a topic that is gaining interest and utility over time, and it has two attributes: *novelty* and *growth* (Small, Boyack, and Klavans, 2014; Tu and Seng, 2012). Rotolo, Hicks, and Martin (2015) explain the correlations of these two attributes in the following way: Up to the point of the emergence, a research topic is specified by a high level of novelty. At that time, it does not attract any considerable attention from the scientific community, and due to the limited impact, its growth is nearly flat. After some turning points (i.e., the appearance of significant scientific publications), the research topic starts to grow faster and may become a *trend* if the growth is fast; however, the level of a novelty decreases continuously once the emergence becomes clear. Then, at the final phase, the topic becomes well-established while its novelty levels out (He and Chen, 2018).

To detect trends in textual data, one employs computational analysis techniques and Natural Language Processing (NLP). In the subsequent section, we give an overview of the existing approaches for topic modeling and trend detection.

## 5.2 DETECTION OF EMERGING RESEARCH TRENDS

The task of trend detection is closely associated with the topic modeling task since, in the first instance, it detects the latent topics in the substantial collection of data. It secondarily employs techniques to investigate the evolution of the topic and whether it has the potential to become a trend.

Topic modeling has an extensive history of applications in the scientific domain, including but not limited to studies of temporal trends (Griffiths and Steyvers, 2004; Wang and McCallum, 2006) and investigation of the impact prediction (Yogatama et al., 2011). Below we give a general overview of the most significant methods used in this domain, explain their importance and, if any, the potential limitations.

### 5.2.1 *Methods for Topic Modeling*

In order to extract a topic from textual documents (i.e., research papers, blog posts), the precise definition of the model for topic representation is crucial.

LATENT SEMANTIC ANALYSIS or formerly called Latent Semantic Indexing is one of the fundamental methods in the topic modeling domain, proposed by Deerwester et al. in 1990. Its primary goal is to generate vector-based representations for documents and terms and then employ these representations to compute the similarity between documents (resp. terms) in order to select the most related ones. In this regard, Latent Semantic Analysis (LSA) takes a matrix of documents and terms and then decomposes it into a separate document-topic matrix and a topic-term matrix. Before turning to the central task of finding latent topics that capture the relationships among the words and documents, it also performs dimensionality reduction utilizing truncated Singular Value Decomposition. This technique is usually applied to reduce the sparseness, noisiness, and redundancy across dimensions in the document-topic matrix. Finally, using the resulting document and term vectors, one applies measures such as *cosine similarity* to evaluate the similarity of different documents, words, or queries.

One of the extensions to traditional LSA is probabilistic Latent Semantic Analysis (pLSA), that was introduced by Hofmann in 1999 to fix several shortcomings. The foremost advantage of the follow-up method was that it could successfully automate document indexing, which in turn improved the LSA in a probabilistic sense by using a generative model (Alghamdi and Alfalqi, 2015). Furthermore, the pLSA can differentiate between diverse contexts of word usage without utilizing dictionaries or a thesaurus. That affects better polysemy disambiguation and discloses typical similarities by grouping words that share a similar context. Among the works that employ pLSA, is the approach proposed by Gohr et al. (2009), where authors apply pLSA in a window that slides across the stream of documents, to analyze the evolution of topics. Also, Mei et al. (2008) uses the pLSA in order to create a network of topics.

**LATENT DIRICHLET ALLOCATION** was developed by Blei, Ng, and Jordan (2003) to overcome the limitations of both LSA and pLSA models. Nowadays, Latent Dirichlet Allocation (LDA) is expected to be one of the most utilized and robust probabilistic topic models. The fundamental concept of LDA is the assumption that every document is represented as a *mixture of topics*, where each topic is a discrete probability distribution that defines the probability of each word to appear in a given topic. These topic probabilities provide a compressed representation of a document. In other words, LDA models all of $d$ documents in the collection, as a mixture over $n$ latent topics, where each of the topics describes a multinomial distribution over a $w$ word in the vocabulary (Alghamdi and Alfalqi, 2015).

LDA fixes such weaknesses of the pLSA, like the incapacity to assign a probability to previously unseen documents (because pLSA learns the topic mixtures only for documents seen in the training phase), and the overfitting problem (i.e., the number of parameters in pLSA increases linearly with the size of the corpus) (Salatino, 2015).

An extension of the conventional LDA is the hierarchical LDA (Griffiths et al., 2004) where topics are grouped in hierarchies. Each node in the hierarchy is associated with a topic, and a topic is a distribution across words. Another comparable approach is the Relational Topic Model (Chang and Blei, 2010), which is a mixture of a topic model and a network model for groups of linked documents. The attribute of every document is its text (i.e., discrete observations taken from a fixed vocabulary), and the links between documents are hyperlinks or citations.

LDA is a prevalent method for discovering topics (Blei and Lafferty, 2006; Bolelli, Ertekin, and Giles, 2009; Bolelli et al., 2009; Hall, Jurafsky, and Manning, 2008; Wang and Blei, 2011; Wang and McCallum, 2006). However, it was explored, that to train and fine-tune a stable LDA model could be very challenging and time-consuming (Agrawal, Fu, and Menzies, 2018).

**CORRELATED TOPIC MODEL**: One of the main drawbacks of the LDA is its incapability of capturing dependencies among the topics (Alghamdi and Alfalqi, 2015; He et al., 2017). Therefore, Blei and Lafferty (2007) proposed the Correlated Topic Model as an extension of the LDA approach. In this model, the authors replaced the Dirichlet distribution with a logistic-normal distribution that models pairwise topic correlations with the Gaussian covariance matrix (He et al., 2017).

However, one of the shortcomings of this model is the computational cost. The number of parameters in the covariance matrix increases quadratic to the number of topics, and parameter estimation for the full-rank matrix can be inaccurate in high-dimensional space (He et al., 2017). Therefore, He et al. (2017)

proposed their method to overcome this problem. The authors introduced the model, which learns dense topic embeddings and captures topic correlations through the similarity between the topic vectors. According to He et al., their method allows efficient inference in the low-dimensional embedding space and significantly reduces previous cubic or quadratic time complexity to linear concerning the topic number.

### 5.2.2 *Topic Evolution of Scientific Publications*

Although the topic modeling task is essential, it has a significant offspring task, namely the *modeling of topic evolution*. Methods able to identify topics within the context of time are highly applicable in the scientific domain. They allow understanding of the topic lineage and reveal how research on one topic influences another. Generally, these methods could be classified according to the primary sources of information they employ: text, citations or key phrases.

TEXT-BASED: Extensive textual collections have dynamic co-occurrences of word patterns that change over time. Thus, models that track topics evolution over time should consider both word co-occurrence patterns and the timestamps Blei and Lafferty, 2006.

In 2006 Wang and McCallum proposed a Non-Markov Continuous Time model for the detection of topical trends in the collection of NeurIPS publications (the overall timestamp of 17 years was considered). The authors made the following assumption: If the pattern of the word co-occurrence exists for a short time, then the system creates a *narrow-time-distribution topic*. Whereas, for the long-term patterns system generated a *broad- time-distribution topic*. The principal objective of this approach is that it models topic evolution without discretizing the time, i.e., the state at time $t + t_1$ is independent of the state at time $t$ (Wang and McCallum, 2006).

Another work done by Blei and Lafferty (2006) proposed the Dynamic Topic Models. The authors assumed that the collection of documents is organized based on certain time spans, and the documents belonging to every time span are represented with a K-component model. Thereby, the topics are associated with time span $t$ and emerge from topics corresponding to span time $t - 1$. One of the main differences of this model compared to other existing approaches is that it utilizes Gaussian distribution for the topic parameters instead of the conventional Dirichlet distribution, and therefore, can capture the evolution over various time spans.

CITATION-BASED analysis, is the second popular direction that is considered to be effective for trend identification He et al., 2009; Le, Ho, and Nakamori, 2005; Shibata, Kajikawa, and Takeda, 2009; Shibata et al., 2008; Small, 2006. This method assumes that citations in their various forms (i.e., bibliographic citations, co-citation networks, citation graphs) indicate the meaningful relationship between topics, and uses citations to model the topic evolution in the scientific domain. Nie and Sun (2017) utilize this approach along with the Latent Dirichlet Allocation (LDA) and *k-means* clustering to identify research trends. The authors first use LDA to extract features and determine the optimal number of topics. Then, they use *k-means* to obtain thematic clusters, and finally, they compute *citation functions* to identify the changes of clusters over time.

Though the citation-based approach's main drawback is that a consistent collection of publications associated with a specific research topic is required for these techniques to detect the cluster and novelty of the particular topic He and Chen, 2018. Furthermore, there are also many trend detection scenarios (e.g., research news articles or research blog posts) in which citations are not readily available. That makes this approach not applicable to this kind of data.

KEYPHRASE-BASED: Another standard solution is based on the use of *keyphrase information* extracted from research papers. In this case, every keyword is representative of a single research topic. Systems like `Saffron` (Monaghan et al., 2010) as well as Saffron-based models use this approach. Saffron is a research toolkit that provides algorithms for keyphrase extraction, entity linking and taxonomy extraction. Asooja et al. (2016) utilize Saffron to extract the keywords from LREC proceedings, and then proposed trend forecast based on regression models as an extension to Saffron.

However, this method raises many questions, including the following: how to deal with the noisiness of keywords (i.e., not relevant keywords) and how to handle keywords that have their hierarchies (i.e., sub-areas)? Other drawbacks of this approach include the ambiguity of keywords (i.e., *"Java"* as an island and *"Java"* as a programming language) or necessity to deal with synonyms, which could be treated as different topics.

## 5.3 SUMMARY

An emerging trend detection algorithm aims to recognize topics that were earlier inappreciable but now gaining the importance and interest within the specific domain. Knowledge of emerging trends is especially essential for stakeholders, researchers, academic publishers, and funding organizations. Assume a business analyst in the biotech company who is interested in the analysis of technical articles for recent trends that could potentially impact the companies investments. A manual review of all the available information in a specific domain would be extremely time-consuming or even not feasible. However, this process could be solved through Intelligent Process Automation (IPA) algorithms. Such software would assist human experts who are tasked with identifying emerging trends through automatic analysis of extensive textual collections and visualization of occurrences and tendencies of a given topic.

# 6 BENCHMARK FOR SCIENTIFIC TREND DETECTION

This chapter covers work already published at international peer-reviewed conferences. The relevant publication is Moiseeva and Schütze, 2020. The research described in this chapter was carried out in its entirety by the author of the thesis. The other author of the publication acted as an advisor.

Computational analysis and modeling of the evolution of trends is a significant area of research because of its socio-economic impact. However, no large publicly available benchmark for trend detection currently exists, making a comparative evaluation of methods impossible. We remedy this situation by publishing the benchmark TRENDNERT, consisting of a set of gold trends (resp. downtrends) and document labels that is available as an unlimited download, and a large underlying document collection that can also be obtained for free. We propose Mean Average Precision (MAP) as an evaluation measure for trend detection and apply this measure in an investigation of several baselines.

## 6.1 MOTIVATION

What is an emerging research topic and how is it defined in the literature? At the time of emergence, a research topic often does not attract much recognition from the scientific society and is exemplified by just a few publications. He and Chen (2018) associate the appearance of a trend to some triggering event, e.g., the publication of an article in a high-impact journal. Later the research topic starts to grow faster and becomes a trend (He and Chen, 2018).

We adopt this as our key representation of *trend* in this paper: a trend is a research topic with a strongly increasing number of publications in a particular time interval. In opposite to trends, there are also *downtrends*, which refer to the topics that move lower in their demand or growth as they evolve. Therefore, we define a *downtrend* as the converse to a *trend*: a downtrend is a research topic that has a strongly decreasing number of publications in a particular time interval. Downtrends can be as important to detect as trends, e.g., for a funding agency planning its budget.

To detect both types of topics (i.e., trends and downtrends) in textual data, one employs *computational analysis* techniques and NLP. These methods enable the analysis of extensive textual collections and gain valuable insights into topical trendiness and evolution over time. Despite the overall importance of trend analysis, there is, to our knowledge, no large publicly available benchmark for trend detection. This makes a comparative evaluation of methods and algorithms impossible. Most of the previous work has used proprietary or moderate datasets, or evaluation measures were not directly bound to the objectives of trend detection (Gollapalli and Li, 2015).

We remedy this situation by publishing the benchmark TRENDNERT[1]. The benchmark consists of the following:

1. A set of gold trends compiled based on an extensive analysis of the meta-literature;
2. A set of labeled documents (based on more than one million scientific publications), where each document is assigned to a trend, a downtrend or the class of flat topics, and supported by the topic-title;
3. An evaluation script to compute Mean Average Precision (MAP), as the measure for the accuracy of trend detection;

We make the benchmark available as unlimited downloads[2]. The underlying research corpus can be obtained for free from Semantic Scholar[3] (Ammar et al., 2018).

### 6.1.1 *Outline and Contributions*

In this work, we address the challenge of the benchmark creation for (down)trend detection. This chapter is structured as follows: Section 6.2 introduces our model for corpus creation and its fundamental components. In Section 6.3, we present the crowdsourcing procedure for the benchmark labeling. Section 6.5 outlines the proposed baseline for trend detection, our experimental setup, and outcomes. Finally, we illustrate the details on the distributed benchmark in Section 6.6 and conclude in Section 6.7.

Our contributions within this work are as follows:

- TRENDNERT is the first publicly available benchmark for (down)-trend detection.
- TRENDNERT is based on a collection of more than one million documents. It is among the largest that has been used for trend detection and therefore offers a realistic setting for developing trend detection algorithms.
- TRENDNERT addresses the task of detecting both trends and downtrends. To the best of our knowledge, the task of downtrend detection has not been addressed before.

[1]The name is a concatenation of *trend* and its ananym *dnert*. Because it supports the evaluation of both trends and downtrends.
[2]https://doi.org/10.17605/OSF.IO/DHZCT
[3]`https://api.semanticscholar.org/corpus/`

## 6.2 CORPUS CREATION

In this section, we explain the methodology we used to create the TRENDNERT benchmark.

### 6.2.1 *Underlying Data*

We employ a corpus provided by Semantic Scholar.[4] It includes more than one million papers published mostly between 2000 and 2015 in about 5000 computer science journals and conference proceedings. Every record in the collection consists of a title, key phrases, abstract, full text, and metadata.

### 6.2.2 *Stratification*

The distribution of documents over time in the entire original dataset is skewed (see Figure 11). We discovered that clustering the entire collection as well as a random sample produces corrupt results because more weight is given to later years than to earlier years. Therefore, we first generated a *stratified sample* of the original document collection. To this end, we randomly pick 10, 000 documents for each year between 2000 and 2016 for an overall sample size of 160, 000.

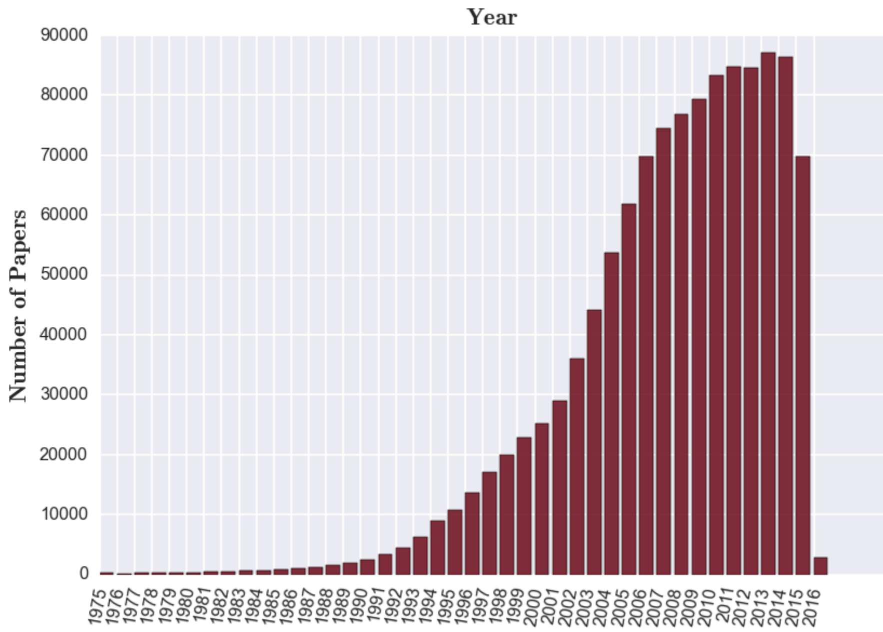


Figure 11: Overall distribution of papers in the entire dataset. Years 1975 to 2016.

[4] https://www.semanticscholar.org/

### 6.2.3 *Document representations & Clustering*

As we mentioned, this work's primary focus was the creation of a benchmark, not the development of new algorithms or models for trend detection. Therefore, for our experiments, we have selected an algorithm based on *k-means* clustering as a simple baseline method for trend detection.

In conventional document clustering, documents are usually represented as bag-of-words (BOW) feature vectors (Manning, Raghavan, and Schütze, 2008). Those, however, have a major weakness: they neglect the semantics of words (Le and Mikolov, 2014). Still, the recent work in representation learning domain and particularly *doc2vec* – the method proposed by Le and Mikolov (2014) – can provide representations to document clustering that overcome this weakness.

The *doc2vec*[5] algorithm is inspired by techniques for learning word vectors (Mikolov et al., 2013) and is capable of capturing semantic regularities in textual collections. This is an unsupervised approach for learning continuous distributed vector representations for text (i.e., paragraphs, documents). This approach maps documents into a vector space such that semantically related documents are assigned similar vector representations (e.g., an article about *"genomics"* is closer to an article about *"gene expression"* than to an article about *"fuzzy sets"*). Formally, each paragraph is mapped to the individual vector, represented by a column in matrix $D$, and every word is also mapped to the individual vector, represented by a column in matrix $W$. The paragraph vector and word vectors are concatenated to predict the next word in a context (Le and Mikolov, 2014). Other work has already successfully employed this type of document representations for topic modeling, combining them with both Latent Dirichlet Allocation (LDA) and clustering approaches (Curiskis et al., 2019; Dieng, Ruiz, and Blei, 2019; Moody, 2016; Xie and Xing, 2013).

In our work, we run *doc2vec* on the stratified sample of $160,000$ papers and represent each document as a length-normalized vector (i.e., document embedding). These vectors are then clustered into $k = 1000$ clusters using the scikit-learn[6] implementation of *k-means* (MacQueen, 1967) with default parameters. The combination of document representations with a clustering algorithm is conceptually simple and interpretable. Also, the comprehensive comparative evaluation of topic modeling methods utilizing document embeddings performed by Curiskis et al. (2019) showed that *doc2vec* feature representations with *k-means* clustering outperform several other methods[7] on three evaluation measures (Normalized Mutual Information, Adjusted Mutual Information, and Adjusted Rand Index).

---

[5] We use the implementation provided by Gensim: `https://radimrehurek.com/gensim/models/doc2vec.html`

[6] https://scikit-learn.org/stable/

[7] hierarchical clustering, k-medoids, NMF and LDA

We run 10 trials of stratification and clustering, resulting in 10 different clusterings. We do this to protect against the variability of clustering and because we do not want to rely on a single clustering for proposing (down)trends for the benchmark.

### 6.2.4 *Trend & Downtrend Estimation*

Recalling our definitions of trend and downtrend: A *trend* (resp. *downtrend*) is defined as a research topic that has a strongly increasing (resp. decreasing) number of publications in a particular time interval.

Linear trend estimation is a widely used technique in predictive (time-series) analysis that proves statements about tendencies in the data, by correlating the measurements to the times at which they occurred (Hess, Iyer, and Malm, 2001). Considering the definition of trend (resp. downtrend) and the main concept of linear trend estimation models, we utilize the *linear regression* to identify trend and downtrend candidates in resulting clustering sets.

Specifically, we count the number of documents per year for a cluster and then estimate the parameters of the best fitting line (i.e., slope). Clusters are then ranked according to the resulting slope and the $n = 150$ clusters with the largest (resp. smallest) slope are selected as *trend* (resp. *downtrend*) *candidates*. Thus, our definition of a single-clustering (down)trend candidate is a cluster with extreme ascending or descending slope. Figure 12 and Figure 13 give two examples of (down)trend candidates.

### 6.2.5 *(Down)trend Candidates Validation*

As detailed in Section 6.2.3, we run ten series of clustering, resulting in ten discrete clustering sets. We then estimated the *(down)trend candidates* according to the procedure described in Section 6.2.4. Consequently, for each of the ten clustering sets, we obtained 150 trend and 150 downtrend candidates.

Nonetheless, for the benchmark, among all the (down)trend candidates across the ten clustering sets, we want to sort out only those that are *consistent* overall sets. To obtain consistent (down)trend candidates, we apply a *Jaccard coefficient* metric, that is used for measuring the similarity and diversity of sample sets. We define an equivalence relation $\sim_R$: two candidates (i.e., clusters) are equivalent if their Jaccard coefficient is $\geqslant \tau_\sim$, where $\tau_\sim = 0.5$. Following this notion, we compute the equivalence of (down)trend candidates across the ten sets.

Finally, we define an equivalence class as an *equivalence class trend candidate* (resp. *equivalence class downtrend candidate*) if it contains trend (resp. downtrends) candidates in at least half of the clusterings. This

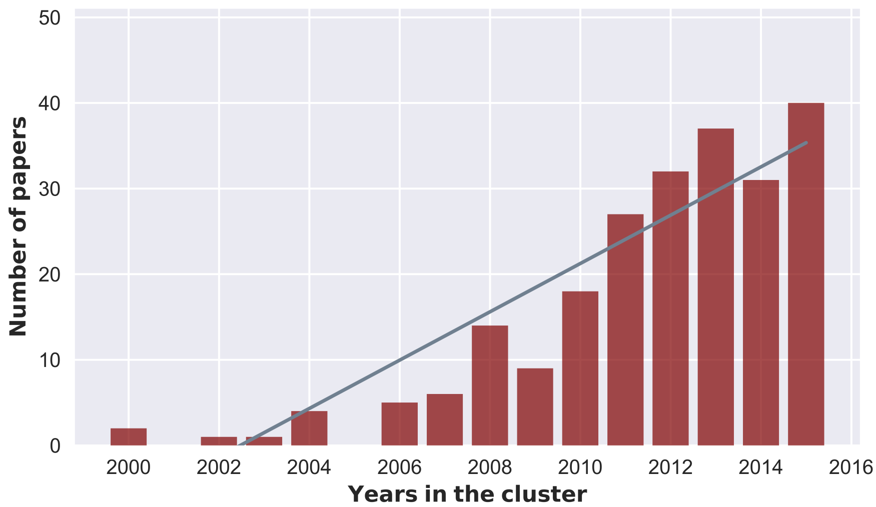


Figure 12: A trend candidate (positive slope); Topic: Sentiment and Emotion Analysis (using Social Media Channels)

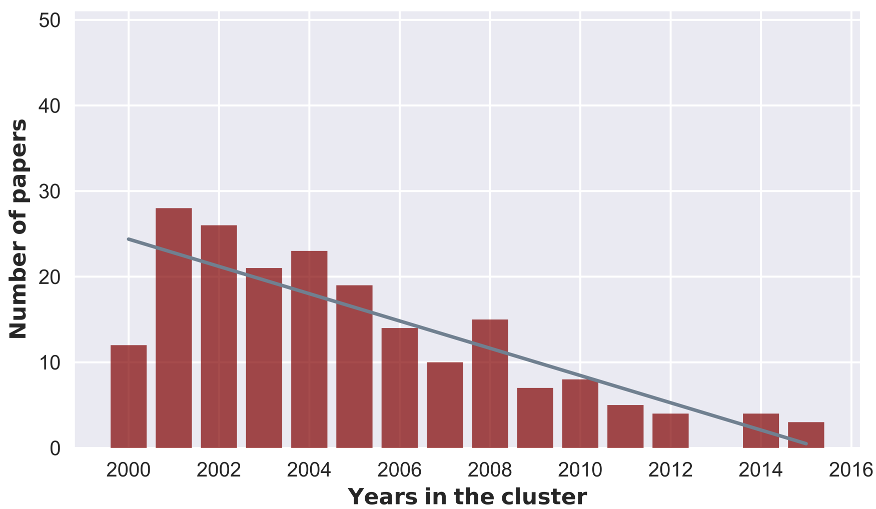


Figure 13: A downtrend candidate (negative slope); Topic: XML

procedure gave us in total 110 equivalence class trend candidates and 107 equivalence class downtrend candidates that we then annotate for our benchmark.[8]

## 6.3 BENCHMARK ANNOTATION

To annotate the equivalence class (down)trend candidates obtained from our model we used the *Figure-Eight*[9] platform. It provides an integrated quality-control mechanism and a training phase before the actual annotation (Kolhatkar, Zinsmeister, and Hirst, 2013), which minimizes the rate of poorly qualified annotators.

We design our annotation task as follows: A worker is shown a *description* of a particular (down)trend candidate along with the list of possible (down)trend *tags*. Descriptions are mixed and presented in random order. The temporal presentation order of candidates is also arbitrary. The worker is asked to pick from the list with (down)trend tags an item that matches the cluster (resp. candidate) best. Even if there are several items that are a possible match, the worker must choose only one. If there is no matching item, the worker can select the option *other*. Three different workers evaluated each cluster, and the final label is the majority label. If there is no majority label, we present the cluster repeatedly to the workers until there is a majority.

The *descriptors* of candidates are collected through the following procedure:

- First, we review the documents assigned to a candidate cluster.
- Then we extract keywords and titles of the papers attributed to the cluster. Semantic Scholar provides keywords and titles as a part of the metadata.
- Finally, we compute the top 15 most frequent keywords and randomly select 15 titles. These keywords and titles are the descriptor of the cluster candidate presented to crowd-workers.

We create the (down)trend *tags* based on keywords, titles, and metadata, such as publication venue. The cluster candidates were manually labeled by graduate employees (i.e., domain experts in the computer science domain) considering the abovementioned items.

### 6.3.1 *Inter-annotator Agreement*

To ensure the quality of obtained annotations, we computed the Inter Annotator Agreement (IAA) - a measure of how well two (or more)

[8]Note that the overall number of 217 refers to the number of the (down)trend candidates and not to the number of documents. Each candidate contains hundreds of documents.

[9]Formerly called CrowdFlower.

annotators can make the same decision for a particular category. To do that, we used the following metrics:

KRIPPENDORFF'S $\alpha$ is a reliability coefficient developed to measure the agreement among observers or measuring instruments drawing distinctions among typically unstructured phenomena, or assign computable values to them (Krippendorff, 2011). We choose Krippendorff's $\alpha$ as an inter-annotator agreement measure because it – unlike other specialized coefficients – is a generalization of several reliability criteria and applies to situations like ours where we have more than two annotators, and a given annotator only labels a subset of the data (i.e., we have to deal with missing values).

Krippendorff (2011) has proposed several metric variations each of which is associated with a specific set of weights. We use Krippendorff's $\alpha$, considering *any number of observers* and *missing data*. In this case, Krippendorff's $\alpha$ for the overall number of 217 annotated documents is $\alpha = 0.798$. According to Landis and Koch (1977), this value corresponds to a *substantial level* of agreement.

FIGURE-EIGHT AGREEMENT RATE: Figure-Eight[10] provides an agreement score $c$ for each annotated unit $u$, which is based on the majority vote of the trusted workers. The score has been proven to perform well compared to the classic metrics (Kolhatkar, Zinsmeister, and Hirst, 2013). We computed the average C for the 217 annotated documents. C is 0.799.

Since both Krippendorff's $\alpha$ and internal Figure Eight IAA are rather high, we consider the obtained annotations for the benchmark to be of good quality.

### 6.3.2 *Gold Trends*

We compiled a list of computer science trends for the years 2016 based on the analysis of the twelve survey publications (Ankerholz, 2016; Augustin, 2016; Brooks, 2016; Frot, 2016; Harriet, 2015; IEEE Computer Society, 2016; Markov, 2015; Meyerson and Mariette, 2016; Nessma, 2015; Reese, 2015; Rivera, 2016; Zaino, 2016). For this year we identified a total of 31 trends; see Table 20.

Since there is a variation as to the name of a specific trend, we created a unique name for each of them. Out of the 31 trends, 28 (90,3%) are instantiated as trends by our model that we confirmed in the crowdsourcing procedure[11]. We searched for the three missing trends using a variety of techniques (i.e., inspection of non-trends/ downtrends; random browsing of documents not assigned to trend

[10] Former CrowdFlower platform.

[11] The author verified this based on her judgment of equivalence of one of the 31 gold trend names in the literature with one of the trend tags.

candidates; and keyword searches). Two of the missing trends do not seem to occur in the collection at all: *"Human Augmentation"* and *"Food and Water Technology"*. The third missing trend, *"Cryptocurrency and Cryptography"*, does occur in the collection, but the occurrence rate is very small (ca. 0.01%). We do not consider these three trends in the rest of the paper.

| Computer Science Trend | Computer Science Trend |
|---|---|
| Autonomous Agents and Systems | Natural Language Processing |
| Autonomous Vehicles | Open Source |
| Big Data Analytics | Privacy |
| Bioinformatics and (Bio) Neuroscience | Quantum Computing |
| Biometrics and Personal Identification | Recommender Systems and Social Networks |
| Cloud Computing and Software as a Service | Reinforcement Learning |
| Cryptocurrency and Cryptography* | Renewable Energy |
| Cyber Security | Robotics |
| E-business | Semantic Web |
| Food and Water Technology* | Sentiment and Emotion Analysis |
| Game-based Learning | Smart Cities |
| Games and (Virtual) Augmented Reality | Supply Chains and RFIDs |
| Human Augmentation* | Technology for Climate Change |
| Machine/Deep Learning | Transportation and Energy |
| Medical Advances and DNA Computing | Wearables |
| Mobile Computing | |

Table 20: 31 areas identified as computer science trends for year 2016 in the media. 28 of these trends are instantiated by our model as well, and thus covered in our benchmark. Topics marked with asterisk ($*$) were not found in our underlying data collection.

In summary, based on our analysis of meta-literature, we identified 28 gold trends that also occur in our collection. We will use them as the *gold standard* that *trend detection algorithms should aim to discover*.

## 6.4 EVALUATION MEASURE FOR TREND DETECTION

Whether something is a trend is a graded concept. For this reason, we adopt an evaluation measure based on a ranking of candidates, specifically Mean Average Precision (MAP). The score denotes the average of the precision values of ranks of correctly identified and non-redundant trends.

We approach trend detection as computing a ranked list of sets of documents, where each set is a trend candidate. We refer to documents as *trendy*, *downtrendy* and *flat* depending on which class they are assigned to in our benchmark. We consider a trend candidate

$c$ as a correct recognition of gold trend $t$ if it satisfies the following requirements:

- $c$ is trendy;
- $t$ is the largest trend in $c$;
- $|t \cap c|/|c| \geqslant \rho$, where $\rho$ is a coverage parameter;
- $t$ was not earlier recognized in the list.

These criteria give us a *true positive* (i.e., recognition of a gold trend) or *false positive* (otherwise) for every position of the ranked list, and a precision score for each trend. Precision for a trend that was not observed is 0. We then compute MAP; see Table 21 for an example.

| Gold Trends | Gold Downtrends |
|---|---|
| Cloud Computing | Compilers |
| Bioinformatics | Petri Nets |
| Sentiment Analysis | Fuzzy Sets |
| Privacy | Routing Protocols |

| | *Trend Candidate* | $\|t \cap c\|/\|c\|$ | *tp/fp* | P |
|---|---|---|---|---|
| $c_1$ | Cloud Computing | 0.60 | tp | 1.00 |
| $c_2$ | Privacy | 0.20 | fp | 0.50 |
| $c_3$ | Cloud Computing | 0.70 | fp | 0.33 |
| $c_4$ | Bioinformatics | 0.55 | tp | 0.50 |
| $c_5$ | Sentiment Analysis | 0.95 | tp | 0.60 |
| $c_6$ | Sentiment Analysis | 0.59 | fp | 0.50 |
| $c_7$ | Bioinformatics | 0.41 | fp | 0.43 |
| $c_8$ | Compilers | 0.60 | fp | 0.38 |
| $c_9$ | Petri Nets | 0.80 | fp | 0.33 |
| $c_{10}$ | Privacy | 0.33 | fp | 0.30 |

Table 21: Example of proposed MAP evaluation (with $\rho = 0.5$): MAP is the average of the precision values: 1.0 (Cloud Computing), 0.5 (Bioinformatics), 0.6 (Sentiment Analysis), and 0.0 (Privacy), i.e., MAP = 2.1/4 = 0.525. True positive: *tp*, False positive: *fp*, Precision: *P*.

## 6.5 TREND DETECTION BASELINES

In this section, we investigate the impact of four different configuration choices on the performance of trend detection by way of ablation.

### 6.5.1 *Configurations*

ABSTRACTS VS. FULL TEXTS: The underlying data collection contains both abstracts and full texts[12]. Our initial expectation was that the full text of a document comprises more information than the abstract, and it should be a more reliable basis for trend detection. This is one of the configurations we test in the ablation. We employ our proposed method on full text (solely document texts without abstracts) and abstract (only abstract texts) collection separately and observe the results.

STRATIFICATION: Due to the uneven distribution of documents over the years, the last years (with increased volume of publications) may get too much impact on results compared to earlier years. To investigate this, we conduct an ablation experiment where we compare: (i) randomly sampled 160k documents from the entire collection and (ii) stratified sampling. In stratified sampling, we select 10k documents for each of the years 2000 – 2015, resulting in a sampled collection of 160k.

LENGTH $L_t$ OF INTERVAL: Clusters are ranked by *trendiness*, and the resulting ranked list is evaluated. To measure the trendiness or growth of topics over time we fit a line to an interval $\{(i, n_i\}|i_0 \leqslant i < i_0 + L_t\}$ by linear regression, where $L_t$ is the length of the interval, $i$ is one of the 16 years (2000, ..., 2015) and $n_i$ is the number of documents that were assigned to cluster $c$ in that year. As a simple default baseline, we apply the regression to a half of an entire interval, i.e., $L_t = 8$. There are nine such intervals in our 16-year period. As the final measure of growth for the cluster, we take the maximal of the nine individual slopes. To determine how much this configuration choice affects our results, we also test a linear regression over the entire time span, i.e., $L_t = 16$. In this case, there is a single interval.

CLUSTERING METHOD: We consider *Gaussian Mixture Models* (GMM) and *k-means*. We use GMM with *spherical* type of covariance, where each component has the same variance. For both, we proceed as described in Section 6.2.3.

### 6.5.2 *Experimental Setup and Results*

We conducted experiments on the Semantic Scholar corpus and evaluated a ranked list of trend candidates against the benchmark considering the configurations mentioned above. We adopted Mean Average Precision (MAP) as the principal evaluation measure. As a secondary evaluation measure, we computed recall at 50 (R@50), which estimates the percentage of gold trends found in the list of 50 highest-ranked trend candidates. We found that the configuration (0) in Ta-

[12]Semantic Scholar provided the underlying dataset at the beginning of our work in 2016.

ble 22 works best. We then conducted ablation experiments to discover the importance of the configuration choices.

| | **(0)** | **(1)** | **(2)** | **(3)** | **(4)** |
|---|---|---|---|---|---|
| document part | A | **F** | A | A | A |
| stratification | yes | yes | **no** | yes | yes |
| $L_t$ | 8 | 8 | 8 | **16** | 8 |
| clustering | $GMM^{sph}$ | $GMM^{sph}$ | $GMM^{sph}$ | $GMM^{sph}$ | **k$\mu$** |
| **MAP** | .36 (.03) | .07 (.19) | .30 (.03) | .32 (.04) | .34 (.21) |
| **R@50 avg** | .50 (.012) | .25 (.018) | .50 (.016) | .46 (.018) | .53(.014) |
| **R@50 max** | .61 | .37 | .53 | .61 | .61 |

Table 22: Ablation results. Standard deviations are in parentheses. *GMM* clustering performed with the spherical (sph.) type of covariance. *K-means* clustering is denoted as k$\mu$ in the ablation table.

COMPARING (0) AND (1), [13] we observed that abstracts (A) are a more reliable representation for our trend detection baseline than full texts (F). The possible explanation for that observation is that an abstract is a *summary* of a scientific paper that covers only the central points and is semantically very concise and rich. In opposite, the full text contains numerous parts that are secondary (i.e., future work, related work) and, thus, may skew the document's overall meaning.

COMPARING (0) AND (2), we recognized that stratification ($yes$) improves results compared to the setting with non-stratified ($no$), randomly sampled data. We would expect the effect of stratification to be even more substantial for collections in which the distribution of documents over time is more skewed than in our.

COMPARING (0) AND (3), the length of the interval is a vital configuration choice. As we assumed, the 16 year interval is rather long, and 8 year interval could be a better choice, primarily if one aims to find *short-term* trends.

COMPARING (0) AND (4), we recognized that topics we obtain from k-means have a similar nature to those from GMM. However, GMM that take variance and covariance into account and estimate a soft assignment, still perform slightly better.

## 6.6 BENCHMARK DESCRIPTION

Below we report the contents of the TRENDNERT benchmark[14].

1. Two records containing information on documents. One file with the documents assigned to (down)trends, and the other file with documents assigned to *flat* topics;

[13] Numbers (0), (1), (2), (3), (4) are the configuration choices in the ablation table.
[14] https://doi.org/10.17605/OSF.IO/DHZCT

2. A script to produce document hash codes;
3. A file to map every hash code to internal IDs in the benchmark;
4. A script to compute MAP (default setting is $\rho = .25$);
5. `README.md` a file with guidance notes;

Every document in the benchmark contains the following information:

- Paper ID: internal ID of documents assigned to each document in the original Semantic Scholar collection;
- Cluster ID: unique ID of meta-cluster where the document was observed;
- Label/Tag: Name of the gold (down)trend assigned by crowd-workers (e.g., *Cyber Security*, *Fuzzy Sets and Systems*);
- Type of (down)trend candidate: trend (T), downtrend (D) or flat topic (F);
- Hash ID[15] of each paper from the original Semantic Scholar collection;

## 6.7 SUMMARY

Emerging trend detection is a promising task for Intelligent Process Automation (IPA) that allows companies and funding agencies to automatically analyze extensive textual collections in order to detect emerging fields and forecast the future employing machine learning algorithms. Thus, the availability of publicly available benchmarks for trend detection is of significant importance, as only thereby can one evaluate the performance and robustness of the techniques employed.

Within this work, we release TRENDNERT – the *first publicly available* benchmark for (down)trend detection, that offers a realistic setting for developing trend detection algorithms. TRENDNERT also supports *downtrend detection* – a significant problem that was not addressed before. We also present several experimental findings on trend detection. First, *stratification* improves trend detection if the distribution of documents is skewed over time. Third, *abstract-based* trend detection performs better than full text if straightforward models are used.

## 6.8 FUTURE WORK

There are possible directions for future work:

- The presented approach to estimate the trendiness of the topic by means of linear regression is straightforward and can be

[15] MD5-hash of: First Author + Title + Year

replaced by more sophisticated versions like Kendall's $\tau$ correlation coefficient or the Least Squares Regression Smoother (LOESS) (Cleveland, 1979).

- It also would be challenging to replace the *k-means* clustering algorithm, that we adopted within this work, with the LDA model while retaining the document representations as it was done in Dieng, Ruiz, and Blei, 2019. The case to investigate would be then, whether the topic modeling outperforms the simple clustering algorithm in this particular setting? And would the resulting system benefit from the topic modeling algorithm?
- Furthermore, it would be of interest to conduct more research on the following: What kind of document representation can make effective use of the information in full texts? Recently proposed contextualised word embeddings could be a possible direction for these experiments.

# 7 DISCUSSION AND FUTURE WORK

As we observed in the two parts of this thesis, Intelligent Process Automation (IPA) is a challenging research area due to its multifacetedness and appliance in a real-world scenario. Its main objective is to automatize time-consuming and routine tasks and free up the knowledge worker for more cognitively demanding tasks. However, this task addresses many difficulties, such as a lack of resources for both training and evaluation of algorithms, as well as an insufficient performance of machine learning techniques when applied to real-world data and practical problems. In this thesis, we investigated two IPA applications within the Natural Language Processing (NLP) domain – *conversational agents* and *predictive analytics* – as well as addressed some of the most common issues.

- In the first part of the thesis, we have addressed the challenge of implementing a robust conversational system for IPA purposes within the practical e-learning domain, considering the conditions of missing training data. The primary purpose of the resulting system is to perform the onboarding process between the tutors and students. This onboarding reduces repetitive and time-consuming activities while allowing tutors to focus on solely mathematical questions, which is the core task with the most impact on students. Besides that, by interacting with users, the system augments the resources with structured and labeled training data that we used to implement learnable dialogue components.

  The realization of such a system was associated with several challenges. Among others were missing structured data and ambiguous user-generated content that requires a precise language understanding. Also, the system that simultaneously communicates with a large number of users has to maintain additional meta policies such as fallback and check-up policies to hold the conversation intelligently.

- Moreover, we have shown the rule-based dialogue system's potential extension using transfer learning. We utilized a small dataset of structured dialogues obtained in a trial-run of the rule-based system and applied the current state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) architecture to solve the domain-specific tasks. Specifically, we retrained two of three components in the conversational system – Named Entity Recognition (NER) and Dialogue Manager (DM) (i.e., NAP task) – and confirmed the applicability of such algorithms in a real-world, low-resource scenario by showing the reasonably high performance of resulting models.

- Last but not least, in the second part of the thesis, we addressed the issue of missing publicly available benchmarks for the evaluation of machine learning algorithms for the trend detection task. To the best of our knowledge, we released the first open benchmark for this purpose, which offers a realistic setting for developing trend detection algorithms. Besides that, this benchmark supports downtrend detection – a significant problem that was not sufficiently addressed before. We additionally proposed the evaluation measure and presented several experimental findings on trend detection.

The ultimate objective of Intelligent Process Automation (IPA) should be a creation of robust algorithms in low-resource and domain-specific conditions. This is still a challenging task; however, some of the the experiments presented in this thesis revealed that this goal could be potentially achieved by means of Transfer Learning (TL) techniques.

# BIBLIOGRAPHY


Agrawal, Amritanshu, Wei Fu, and Tim Menzies (2018). „What is wrong with topic modeling? And how to fix it using search-based software engineering." In: *Information and Software Technology* 98, pp. 74–88 (cit. on p. 66).

Alan, M (1950). „Turing." In: *Computing machinery and intelligence. Mind* 59.236, pp. 433–460 (cit. on p. 7).

Alghamdi, Rubayyi and Khalid Alfalqi (2015). „A survey of topic modeling in text mining." In: *Int. J. Adv. Comput. Sci. Appl.(IJACSA)* 6.1 (cit. on pp. 64–66).

Ammar, Waleed, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. (2018). „Construction of the literature graph in semantic scholar." In: *arXiv preprint arXiv: 1805. 02262* (cit. on p. 72).

Ankerholz, Amber (2016). *2016 Future of Open Source Survey Says Open Source Is The Modern Architecture.* `https://www.linux.com/news/2016-future-open-source-survey-says-open-source-modern-architecture` (cit. on p. 78).

Asooja, Kartik, Georgeta Bordea, Gabriela Vulcu, and Paul Buitelaar (2016). „Forecasting emerging trends from scientific literature." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 417–420 (cit. on p. 68).

Augustin, Jason (2016). *Emergin Science and Technology Trends: A Synthesis of Leading Forecast*. `http://www.defenseinnovationmarketplace.mil/resources/2016_SciTechReport_16June2016.pdf` (cit. on p. 78).

Blei, David M and John D Lafferty (2006). „Dynamic topic models." In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 113–120 (cit. on pp. 66, 67).

Blei, David M, John D Lafferty, et al. (2007). „A correlated topic model of science." In: *The Annals of Applied Statistics* 1.1, pp. 17–35 (cit. on p. 66).

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). „Latent dirichlet allocation." In: *Journal of machine Learning research* 3.Jan, pp. 993–1022 (cit. on p. 66).

Bloem, Peter (2019). *Transformers from Scratch*. `http://www.peterbloem.nl/blog/transformers` (cit. on p. 45).

Bocklisch, Tom, Joey Faulkner, Nick Pawlowski, and Alan Nichol (2017). „Rasa: Open source language understanding and dialogue management." In: *arXiv preprint arXiv:1712.05181* (cit. on p. 15).

Bolelli, Levent, Şeyda Ertekin, and C Giles (2009). „Topic and trend detection in text collections using latent dirichlet allocation." In: *Advances in Information Retrieval*, pp. 776–780 (cit. on p. 66).

Bolelli, Levent, Seyda Ertekin, Ding Zhou, and C Lee Giles (2009). „Finding topic trends in digital libraries.“ In: *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*. ACM, pp. 69–72 (cit. on p. 66).

Bordes, Antoine, Y-Lan Boureau, and Jason Weston (2016). „Learning end-to-end goal-oriented dialog.“ In: *arXiv preprint arXiv:1605.07683* (cit. on pp. 16, 17).

Brooks, Chuck (2016). *7 Top Tech Trends Impacting Innovators in 2016.* `http://innovationexcellence.com/blog/2015/12/26/7-top-tech-trends-impacting-innovators-in-2016/` (cit. on p. 78).

Burtsev, Mikhail, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Eugene Botvinovsky, Nickolay Bushkov, Olga Gureenkova, Andrey Kamenev, Vasily Konovalov, et al. (2018). „DeepPavlov: An Open Source Library for Conversational AI.“ In: (cit. on p. 15).

Chang, Jonathan, David M Blei, et al. (2010). „Hierarchical relational models for document networks.“ In: *The Annals of Applied Statistics* 4.1, pp. 124–150 (cit. on p. 66).

Chen, Hongshen, Xiaorui Liu, Dawei Yin, and Jiliang Tang (2017). „A survey on dialogue systems: Recent advances and new frontiers.“ In: *Acm Sigkdd Explorations Newsletter* 19.2, pp. 25–35 (cit. on pp. 14–16).

Chen, Lingzhen, Alessandro Moschitti, Giuseppe Castellucci, Andrea Favalli, and Raniero Romagnoli (2018). „Transfer Learning for Industrial Applications of Named Entity Recognition.“ In: *NL4AI@ AI* IA*, pp. 129–140 (cit. on p. 43).

Cleveland, William S (1979). „Robust locally weighted regression and smoothing scatterplots.“ In: *Journal of the American statistical association* 74.368, pp. 829–836 (cit. on p. 84).

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (2011). „Natural language processing (almost) from scratch.“ In: *Journal of machine learning research* 12.Aug, pp. 2493–2537 (cit. on p. 16).

Cuayáhuitl, Heriberto, Simon Keizer, and Oliver Lemon (2015). „Strategic dialogue management via deep reinforcement learning.“ In: *arXiv preprint arXiv:1511.08099* (cit. on p. 14).

Cui, Lei, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou (2017). „Superagent: A customer service chatbot for e-commerce websites.“ In: *Proceedings of ACL 2017, System Demonstrations*, pp. 97–102 (cit. on p. 8).

Curiskis, Stephan A, Barry Drake, Thomas R Osborn, and Paul J Kennedy (2019). „An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit.“ In: *Information Processing & Management* (cit. on p. 74).

Dai, Zihang, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov (2019). „Transformer-xl: Attentive language models beyond a fixed-length context.“ In: *arXiv preprint arXiv:1901.02860* (cit. on pp. 43, 44).

Daniel, Ben (2015). „Big Data and analytics in higher education: Opportunities and challenges.“ In: *British journal of educational technology* 46.5, pp. 904–920 (cit. on p. 22).

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman (1990). „Indexing by latent semantic analysis.“ In: *Journal of the American society for information science* 41.6, pp. 391–407 (cit. on p. 65).

Deoras, Anoop, Kaisheng Yao, Xiaodong He, Li Deng, Geoffrey Gerson Zweig, Ruhi Sarikaya, Dong Yu, Mei-Yuh Hwang, and Gregoire Mesnil (2015). *Assignment of semantic labels to a sequence of words using neural network architectures*. US Patent App. 14/016,186 (cit. on p. 13).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). „Bert: Pre-training of deep bidirectional transformers for language understanding.“ In: *arXiv preprint arXiv:1810.04805* (cit. on pp. 42–45).

Dhingra, Bhuwan, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng (2016). „Towards end-to-end reinforcement learning of dialogue agents for information access.“ In: *arXiv preprint arXiv:1609.00777* (cit. on p. 16).

Dieng, Adji B, Francisco JR Ruiz, and David M Blei (2019). „Topic Modeling in Embedding Spaces.“ In: *arXiv preprint arXiv:1907.04907* (cit. on pp. 74, 84).

Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell (2014). „Decaf: A deep convolutional activation feature for generic visual recognition.“ In: *International conference on machine learning*, pp. 647–655 (cit. on p. 43).

Eric, Mihail and Christopher D Manning (2017). „Key-value retrieval networks for task-oriented dialogue.“ In: *arXiv preprint arXiv: 1705. 05414* (cit. on p. 16).

Fang, Hao, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A Smith, and Mari Ostendorf (2018). „Sounding board: A user-centric and content-driven social chatbot.“ In: *arXiv preprint arXiv:1804.10202* (cit. on p. 10).

Friedrich, Fabian, Jan Mendling, and Frank Puhlmann (2011). „Process model generation from natural language text.“ In: *International Conference on Advanced Information Systems Engineering*, pp. 482 –496 (cit. on p. 2).

Frot, Mathilde (2016). *5 Trends in Computer Science Research.* `https://www.topuniversities.com/courses/computer-science-information-systems/5-trends-computer-science-research` (cit. on p. 78).

Glasmachers, Tobias (2017). „Limits of end-to-end learning.“ In: *arXiv preprint arXiv:1704.08305* (cit. on pp. 16, 17).

Goddeau, David, Helen Meng, Joseph Polifroni, Stephanie Seneff, and Senis Busayapongchai (1996). „A form-based dialogue manager for spoken language applications.“ In: *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*. Vol. 2. IEEE, pp. 701–704 (cit. on p. 14).

Gohr, Andre, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou (2009). „Topic evolution in a stream of documents." In: *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, pp. 859–870 (cit. on p. 65).

Gollapalli, Sujatha Das and Xiaoli Li (2015). „EMNLP versus ACL: Analyzing NLP research over time." In: *EMNLP*, pp. 2002–2006 (cit. on p. 72).

Griffiths, Thomas L and Mark Steyvers (2004). „Finding scientific topics." In: *Proceedings of the National academy of Sciences* 101.suppl 1, pp. 5228–5235 (cit. on p. 65).

Griffiths, Thomas L, Michael I Jordan, Joshua B Tenenbaum, and David M Blei (2004). „Hierarchical topic models and the nested Chinese restaurant process." In: *Advances in neural information processing systems*, pp. 17–24 (cit. on p. 66).

Hall, David, Daniel Jurafsky, and Christopher D Manning (2008). „Studying the history of ideas using topic models." In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 363–371 (cit. on p. 66).

Harriet, Taylor (2015). *Privacy will hit tipping point in 2016*. `http://www.cnbc.com/2015/11/09/privacy-will-hit-tipping-point-in-2016.html` (cit. on p. 78).

He, Jiangen and Chaomei Chen (2018). „Predictive Effects of Novelty Measured by Temporal Embeddings on the Growth of Scientific Literature." In: *Frontiers in Research Metrics and Analytics* 3, p. 9 (cit. on pp. 64, 68, 71).

He, Junxian, Zhiting Hu, Taylor Berg-Kirkpatrick, Ying Huang, and Eric P Xing (2017). „Efficient correlated topic modeling with topic embedding." In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 225–233 (cit. on pp. 66, 67).

He, Qi, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles (2009). „Detecting topic evolution in scientific literature: How can citations help?" In: *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pp. 957–966 (cit. on p. 68).

Henderson, Matthew, Blaise Thomson, and Jason D Williams (2014). „The second dialog state tracking challenge." In: *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 263–272 (cit. on p. 15).

Henderson, Matthew, Blaise Thomson, and Steve Young (2013). „Deep neural network approach for the dialog state tracking challenge." In: *Proceedings of the SIGDIAL 2013 Conference*, pp. 467–471 (cit. on p. 14).

Hess, Ann, Hari Iyer, and William Malm (2001). „Linear trend analysis: a comparison of methods." In: *Atmospheric Environment* 35.30. Visibility, Aerosol and Atmospheric Optics, pp. 5211 –5222. ISSN: 1352-2310. DOI: `https://doi.org/10.1016/S1352-2310(01)00342-9`. URL: `http://www.sciencedirect.com/science/article/pii/S1352231001003429` (cit. on p. 75).

Hofmann, Thomas (1999). „Probabilistic latent semantic analysis." In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 289–296 (cit. on p. 65).

Honnibal, Matthew and Ines Montani (2017). „spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing." In: *To appear* 7 (cit. on p. 15).

Howard, Jeremy and Sebastian Ruder (2018). „Universal language model fine-tuning for text classification." In: *arXiv preprint arXiv:1801.06146* (cit. on p. 44).

IEEE Computer Society (2016). *Top 9 Computing Technology Trends for 2016*. `https://www.scientificcomputing.com/news/2016/01/top-9-computing-technology-trends-2016` (cit. on p. 78).

Ivančić, Lucija, Dalia Suša Vugec, and Vesna Bosilj Vukšić (2019). „Robotic Process Automation: Systematic Literature Review." In: *International Conference on Business Process Management*. Springer, pp. 280–295 (cit. on pp. 1, 2).

Jain, Mohit, Pratyush Kumar, Ramachandra Kota, and Shwetak N Patel (2018). „Evaluating and informing the design of chatbots." In: *Proceedings of the 2018 Designing Interactive Systems Conference*. ACM, pp. 895–906 (cit. on p. 8).

Khanpour, Hamed, Nishitha Guntakandla, and Rodney Nielsen (2016). „Dialogue act classification in domain-independent conversations using a deep recurrent neural network." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2012–2021 (cit. on p. 13).

Kim, Young-Bum, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong (2015). „New transfer learning techniques for disparate label sets." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 473–482 (cit. on p. 43).

Kolhatkar, Varada, Heike Zinsmeister, and Graeme Hirst (2013). „Annotating anaphoric shell nouns with their antecedents." In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 112–121 (cit. on pp. 77, 78).

Kontostathis, April, Leon M Galitsky, William M Pottenger, Soma Roy, and Daniel J Phelps (2004). „A survey of emerging trend detection in textual data mining." In: *Survey of text mining*. Springer, pp. 185–224 (cit. on p. 63).

Krippendorff, Klaus (2011). „Computing Krippendorff's alpha- reliability." In: *Annenberg School for Communication (ASC) Departmental Papers* 43 (cit. on p. 78).

Kurata, Gakuto, Bing Xiang, Bowen Zhou, and Mo Yu (2016). „Leveraging sentence-level information with encoder lstm for semantic slot filling." In: *arXiv preprint arXiv:1601.01530* (cit. on p. 13).

Landis, J Richard and Gary G Koch (1977). „The measurement of observer agreement for categorical data." In: *biometrics*, pp. 159–174 (cit. on p. 78).

Le, Minh-Hoang, Tu-Bao Ho, and Yoshiteru Nakamori (2005). „Detecting emerging trends from scientific corpora." In: *International Journal of Knowledge and Systems Sciences* 2.2, pp. 53–59 (cit. on p. 68).

Le, Quoc V and Tomas Mikolov (2014). „Distributed Representations of Sentences and Documents." In: *ICML*. Vol. 14, pp. 1188–1196 (cit. on p. 74).

Lee, Ji Young and Franck Dernoncourt (2016). „Sequential short-text classification with recurrent and convolutional neural networks." In: *arXiv preprint arXiv:1603.03827* (cit. on p. 13).

Leopold, Henrik, Han van der Aa, and Hajo A Reijers (2018). „Identifying candidate tasks for robotic process automation in textual process descriptions." In: *Enterprise, Business-Process and Information Systems Modeling*. Springer, pp. 67–81 (cit. on p. 2).

Levenshtein, Vladimir I (1966). „Binary codes capable of correcting deletions, insertions, and reversals." In: *Soviet physics doklady*. 8, pp. 707–710 (cit. on p. 25).

Liao, Vera, Masud Hussain, Praveen Chandar, Matthew Davis, Yasaman Khazaeni, Marco Patricio Crasso, Dakuo Wang, Michael Muller, N Sadat Shami, Werner Geyer, et al. (2018). „All Work and No Play?" In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, p. 3 (cit. on p. 8).

Lowe, Ryan Thomas, Nissan Pow, Iulian Vlad Serban, Laurent Charlin, Chia-Wei Liu, and Joelle Pineau (2017). „Training end-to-end dialogue systems with the ubuntu dialogue corpus." In: *Dialogue & Discourse* 8.1, pp. 31–65 (cit. on p. 22).

Luger, Ewa and Abigail Sellen (2016). „Like having a really bad PA: the gulf between user expectation and experience of conversational agents." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 5286–5297 (cit. on p. 8).

MacQueen, James (1967). „Some methods for classification and analysis of multivariate observations." In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. Oakland, CA, USA., pp. 281–297 (cit. on p. 74).

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Web publication at informationretrieval.org. Cambridge University Press (cit. on p. 74).

Markov, Igor (2015). *13 Of 2015's Hottest Topics In Computer Science Research*. `https://www.forbes.com/sites/quora/2015/04/22/13-of-2015s-hottest-topics-in-computer-science-research/#fb0d46b1e88d` (cit. on p. 78).

Martin, James H and Daniel Jurafsky (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River.

Mei, Qiaozhu, Deng Cai, Duo Zhang, and ChengXiang Zhai (2008). „Topic modeling with network regularization." In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 101–110 (cit. on p. 65).

Meyerson, Bernard and DiChristina Mariette (2016). *These are the top 10 emerging technologies of 2016.* `http://www3.weforum.org/docs/GAC16_Top10_Emerging_Technologies_2016_report.pdf` (cit. on p. 78).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). „Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781* (cit. on p. 74).

Miller, Alexander, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston (2016). „Key-value memory networks for directly reading documents." In: (cit. on p. 16).

Mohanty, Soumendra and Sachin Vyas (2018). „How to Compete in the Age of Artificial Intelligence." In: (cit. on pp. III, V, 1, 2).

Moiseeva, Alena and Hinrich Schütze (2020). „TRENDNERT: A Benchmark for Trend and Downtrend Detection in a Scientific Domain." In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence* (cit. on p. 71).

Moiseeva, Alena, Dietrich Trautmann, Michael Heimann, and Hinrich Schütze (2020). „Multipurpose Intelligent Process Automation via Conversational Assistant." In: *arXiv preprint arXiv:2001.02284* (cit. on pp. 21, 41).

Monaghan, Fergal, Georgeta Bordea, Krystian Samp, and Paul Buitelaar (2010). „Exploring your research: Sprinkling some saffron on semantic web dog food." In: *Semantic Web Challenge at the International Semantic Web Conference*. Vol. 117. Citeseer, pp. 420–435 (cit. on p. 68).

Monostori, László, András Márkus, Hendrik Van Brussel, and E Westkämpfer (1996). „Machine learning approaches to manufacturing." In: *CIRP annals* 45.2, pp. 675–712 (cit. on p. 15).

Moody, Christopher E (2016). „Mixing dirichlet topic models and word embeddings to make lda2vec." In: *arXiv preprint arXiv:1605.02019* (cit. on p. 74).

Mou, Lili, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin (2016). „How transferable are neural networks in nlp applications?" In: *arXiv preprint arXiv:1603.06111* (cit. on p. 43).

Mrkšić, Nikola, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young (2016). „Neural belief tracker: Data-driven dialogue state tracking." In: *arXiv preprint arXiv:1606.03777* (cit. on p. 14).

Nessma, Joussef (2015). *Top 10 Hottest Research Topics in Computer Science.* `http://www.pouted.com/top-10-hottest-research-topics-in-computer-science/` (cit. on p. 78).

Nie, Binling and Shouqian Sun (2017). „Using text mining techniques to identify research trends: A case study of design research." In: *Applied Sciences* 7.4, p. 401 (cit. on p. 68).

Pan, Sinno Jialin and Qiang Yang (2009). „A survey on transfer learning." In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359 (cit. on p. 41).

Qu, Lizhen, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, and Timothy Baldwin (2016). „Named entity recognition for novel types by transfer learning." In: *arXiv preprint arXiv:1610.09914* (cit. on p. 43).

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). „Language models are unsupervised multitask learners." In: *OpenAI Blog* 1.8 (cit. on p. 44).

Reese, Hope (2015). *7 trends for artificial intelligence in 2016: 'Like 2015 on steroids'*. `http://www.techrepublic.com/article/7-trends-for-artificial-intelligence-in-2016-like-2015-on-steroids/` (cit. on p. 78).

Rivera, Maricel (2016). *Is Digital Game-Based Learning The Future Of Learning?* `https://elearningindustry.com/digital-game-based-learning-future` (cit. on p. 78).

Rotolo, Daniele, Diana Hicks, and Ben R Martin (2015). „What is an emerging technology?" In: *Research Policy* 44.10, pp. 1827–1843 (cit. on p. 64).

Salatino, Angelo (2015). „Early detection and forecasting of research trends." In: (cit. on pp. 63, 66).

Serban, Iulian V, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau (2016). „Building end-to-end dialogue systems using generative hierarchical neural network models." In: *Thirtieth AAAI Conference on Artificial Intelligence* (cit. on p. 11).

Sharif Razavian, Ali, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson (2014). „CNN features off-the-shelf: an astounding baseline for recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813 (cit. on p. 43).

Shibata, Naoki, Yuya Kajikawa, and Yoshiyuki Takeda (2009). „Comparative study on methods of detecting research fronts using different types of citation." In: *Journal of the American Society for information Science and Technology* 60.3, pp. 571–580 (cit. on p. 68).

Shibata, Naoki, Yuya Kajikawa, Yoshiyuki Takeda, and Katsumori Matsushima (2008). „Detecting emerging research fronts based on topological measures in citation networks of scientific publications." In: *Technovation* 28.11, pp. 758–775 (cit. on p. 68).

Small, Henry (2006). „Tracking and predicting growth areas in science." In: *Scientometrics* 68.3, pp. 595–610 (cit. on p. 68).

Small, Henry, Kevin W Boyack, and Richard Klavans (2014). „Identifying emerging topics in science and technology." In: *Research Policy* 43.8, pp. 1450–1467 (cit. on p. 64).

Su, Pei-Hao, Nikola Mrkšić, Iñigo Casanueva, and Ivan Vulić (2018). „Deep Learning for Conversational AI." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 27–32 (cit. on p. 12).

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). „Sequence to sequence learning with neural networks." In: *Advances in neural information processing systems*, pp. 3104–3112 (cit. on pp. 16, 17).

Trautmann, Dietrich, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych (2019). „Robust Argument Unit Recognition and Classification." In: *arXiv preprint arXiv:1904.09688* (cit. on p. 41).

Tu, Yi-Ning and Jia-Lang Seng (2012). „Indices of novelty for emerging topic detection." In: *Information processing & management* 48.2, pp. 303–325 (cit. on p. 64).

Turing, Alan (1949). *London Times letter to the editor* (cit. on p. 7).

Ultes, Stefan, Lina Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, et al. (2017). „Pydial: A multi-domain statistical dialogue system toolkit." In: *Proceedings of ACL 2017, System Demonstrations*, pp. 73–78 (cit. on p. 15).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). „Attention is all you need." In: *Advances in neural information processing systems*, pp. 5998–6008 (cit. on p. 45).

Vincent, Pascal, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol (2008). „Extracting and composing robust features with denoising autoencoders." In: *Proceedings of the 25th international conference on Machine learning*. ACM, pp. 1096–1103 (cit. on p. 44).

Vodolán, Miroslav, Rudolf Kadlec, and Jan Kleindienst (2015). „Hybrid dialog state tracker." In: *arXiv preprint arXiv:1510.03710* (cit. on p. 14).

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman (2018). „Glue: A multi-task benchmark and analysis platform for natural language understanding." In: *arXiv preprint arXiv:1804.07461* (cit. on p. 44).

Wang, Chong and David M Blei (2011). „Collaborative topic modeling for recommending scientific articles." In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 448–456 (cit. on p. 66).

Wang, Xuerui and Andrew McCallum (2006). „Topics over time: a non-Markov continuous-time model of topical trends." In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 424–433 (cit. on pp. 65–67).

Wang, Zhuoran and Oliver Lemon (2013). „A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information." In: *Proceedings of the SIGDIAL 2013 Conference*, pp. 423–432 (cit. on p. 14).

Weizenbaum, Joseph et al. (1966). „ELIZA—a computer program for the study of natural language communication between man and machine." In: *Communications of the ACM* 9.1, pp. 36–45 (cit. on p. 7).

Wen, Tsung-Hsien, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young (2015). „Semantically conditioned lstm-based natural language generation for spoken dialogue systems." In: *arXiv preprint arXiv:1508.01745* (cit. on p. 14).

Wen, Tsung-Hsien, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young (2016). „A network-based end-to-end trainable task-oriented dialogue system." In: *arXiv preprint arXiv:1604.04562* (cit. on pp. 11, 16, 17, 22).

Werner, Alexander, Dietrich Trautmann, Dongheui Lee, and Roberto Lampariello (2015). „Generalization of optimal motion trajectories for bipedal walking." In: pp. 1571–1577 (cit. on p. 41).

Williams, Jason D, Kavosh Asadi, and Geoffrey Zweig (2017). „Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning." In: *arXiv preprint arXiv:1702.03274* (cit. on p. 17).

Williams, Jason, Antoine Raux, and Matthew Henderson (2016). „The dialog state tracking challenge series: A review." In: *Dialogue & Discourse* 7.3, pp. 4–33 (cit. on p. 12).

Wu, Chien-Sheng, Richard Socher, and Caiming Xiong (2019). „Global-to-local Memory Pointer Networks for Task-Oriented Dialogue." In: *arXiv preprint arXiv:1901.04713* (cit. on pp. 15–17).

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. (2016). „Google's neural machine translation system: Bridging the gap between human and machine translation." In: *arXiv preprint arXiv:1609.08144* (cit. on p. 45).

Xie, Pengtao and Eric P Xing (2013). „Integrating Document Clustering and Topic Modeling." In: *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence* (cit. on p. 74).

Yan, Zhao, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li (2017). „Building task-oriented dialogue systems for online shopping." In: *Thirty-First AAAI Conference on Artificial Intelligence* (cit. on p. 14).

Yogatama, Dani, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R Routledge, and Noah A Smith (2011). „Predicting a scientific community's response to an article." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 594–604 (cit. on p. 65).

Young, Steve, Milica Gašić, Blaise Thomson, and Jason D Williams (2013). „Pomdp-based statistical spoken dialog systems: A review." In: *Proceedings of the IEEE* 101.5, pp. 1160–1179 (cit. on p. 17).

Zaino, Jennifer (2016). *2016 Trends for Semantic Web and Semantic Technologies.* `http://www.dataversity.net/2017-predictions-semantic-web-semantic-technologies/` (cit. on p. 78).

Zhou, Hao, Minlie Huang, et al. (2016). „Context-aware natural language generation for spoken dialogue systems." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2032–2041 (cit. on pp. 14, 15).

Zhu, Yukun, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). „Aligning books and movies: Towards story-like visual explanations by watching movies and reading books." In: *Proceedings of the IEEE international conference on computer vision*, pp. 19–27 (cit. on p. 44).