

**CookBERT – Adapting BERT for the Cooking Domain**

Bachelor thesis in Media Informatics at the Institute for Language, Literature and Cultural Studies (I:IMSK)

Submitted by: Pascal Strobel

Address: Gluckstraße 3, 93053 Regensburg

E-Mail (university): pascal.strobel@stud.uni-regensburg.de

E-Mail (private): paschistrobel@web.de

Matriculation number: 2106133

First reviewer: Prof. Dr. Udo Kruschwitz

Second reviewer: PD Dr. David Elsweiler

Supervisor: Alexander Frummet (M. Sc.)

Current semester: 7

Submitted on: 30.2.2016

Contents

[1 Introduction 8](#_Toc96413059)

[2 Background and Related Work 12](#_Toc96413060)

[2.1 BERT 12](#_Toc96413061)

[2.1.1 Pretraining Contextual Representations 12](#_Toc96413062)

[2.1.2 Pretraining Objectives 13](#_Toc96413063)

[2.1.3 BERT workflow 14](#_Toc96413064)

[2.2 BERT and Conversational Agents 14](#_Toc96413065)

[2.3 Adapting BERT for Different Domains 15](#_Toc96413066)

[2.4 BERT for the Cooking Domain 16](#_Toc96413067)

[2.5 Summary and Key Differentiators 16](#_Toc96413068)

[3 Methodology 16](#_Toc96413069)

[3.1 Data for DAPT 16](#_Toc96413070)

[3.2 Analyzing Domain Similarity 18](#_Toc96413071)

[3.3 Domain Vocabulary Insertion 19](#_Toc96413072)

[3.4 DAPT 21](#_Toc96413073)

[3.5 Implementation Details 21](#_Toc96413074)

[3.6 Finetuning 22](#_Toc96413075)

[3.6.1 Intent Classification 22](#_Toc96413076)

[3.6.2 Named-entity recognition 23](#_Toc96413077)

[3.6.3 Question Answering 23](#_Toc96413078)

[4 Evaluation 23](#_Toc96413079)

[4.1 Multi-class classification 23](#_Toc96413080)

[4.2 Named-entity recognition 23](#_Toc96413081)

[4.3 Question Answering 23](#_Toc96413082)

[5 Discussion 23](#_Toc96413083)

[6 Limitations 23](#_Toc96413084)

[7 Conclusion 24](#_Toc96413085)

[Bibliography 25](#_Toc96413086)

[Anhang A: Bausteine wissenschaftlicher Arbeiten 30](#_Toc96413087)

[A1 Theoretische Arbeit 30](#_Toc96413088)

[A2 Konstruktive Arbeit 30](#_Toc96413089)

[A3 Empirische Arbeit 30](#_Toc96413090)

[Erklärung zur Urheberschaft 32](#_Toc96413091)

Abbildungsverzeichnis (optional, in der Regel nicht notwendig)

[Abbildung 1: Blumen (Quelle, Jahr, Seitenzahl) **Fehler! Textmarke nicht definiert.**](#_Toc359834290)

Tabellenverzeichnis (optional, in der Regel nicht notwendig)

[Tabelle 1: Empfohlener Textumfang **Fehler! Textmarke nicht definiert.**](#_Toc359834281)

**Zusammenfassung**

Abstract

* Recent Fortschritt in NLP hat sich auch auf CA ausgewirkt, welche zunehmend ubiquitous werden und in vielen Bereichen des Lebens zu finden sind
* Früher regelbasiert, heute neuronale Netzte
* Eines davon ist BERT

# Introduction

Conversational agents (CAs) like Amazon’s Alexa or Apple’s Siri become more and more pervasive and are applied in a broad range of contexts, including health (Ni et al., 2017; Xu et al., 2019), elderly care (Bickmore et al., 2005), education (Graesser et al., 1999; Winkler et al., 2020), customer service (Cui et al., 2017) and home cooking (Angara et al., 2017; Chu, 2021). Although there are various types of conversational agents, which are titled and categorized very inconsistently in literature and media, they all provide an alternative to traditional methods for humans to seek for information by making the search process more conversational, mainly via written or spoken natural language (McTear, 2020, pp. 12–13). Users benefit from this more natural interaction as it promises an increased ease of use and speed of user requests as well as a convenient usage (Brandtzaeg & Følstad, 2017). While early approaches to create CAs were mainly based on handcrafted rules (e. g. Weizenbaum’s ELIZA (1966)), this has shifted in recent years towards the utilization of large-scale pretrained language models which can gain a superb grasp of human language.

One of the most promising models in recent development is Bidirectional Encoder Representations from Transformers (BERT) - a huge neural network proposed by the Google AI team (Devlin et al., 2018) which is pretrained on 3.3 billion words from BooksCorpus (Zhu et al., 2015) and English Wikipedia. It builds upon previous approaches on pretraining contextual representations (Dai & Le, 2015; Howard & Ruder, 2018; Peters et al., 2018; Radford et al., 2018), but what really sets it apart is that it’s “the first deeply bidirectional, unsupervised language representation, pretrained using only a plain text corpus.” This bidirectionality, combined with the self-attention mechanism, provides a better grasp of word meanings and context, which is reflected in achieving state-of-the-art performance on eleven natural language processing (NLP) tasks (Devlin et al., 2018). BERT’s outstanding performance and its open sourcing ensured that it was subsequently integrated into CA pipelines where it also achieves promising performance for a variety of tasks (Chao & Lane, 2019; Chen et al., 2019; Vakulenko et al., 2021; Voskarides et al., 2020).

As mentioned before, many CAs are applied in a specific context or domain and thus have to deal with domain specific data. For example, a conversational cooking assistant will mostly encounter cooking-related information needs like questions about the preparation or the quantity of ingredients (Frummet et al., 2021) but this is probably not the case for a customer-service chatbot for an e-commerce website. However, one of BERT’s limitations is the lack of domain specific knowledge, since pretraining was only performed on text data of the general domain, which can in turn lead to performance loss on the downstream tasks it is applied for (Gururangan et al., 2020; Lee et al., 2020).

Proceeding from this, the goal of this bachelor thesis is (overcome this limitation) the adaptation of BERT for one particular domain, i.e., the domain of cooking, in order to provide a sophisticated model that can be utilized in conversational agents for this domain. The cooking domain was chosen because it is considered a pertinent context for CAs. Firstly, cooking provides situations where traditional search is rather inconvenient, as users are multi-tasking, and their hands are occupied. Moreover, it has been argued in the past, that aiding in the kitchen (e. g. via recipe recommendations) could potentially lower the barriers to healthier cooking and thus improve the nutrition of people (Elsweiler et al., 2015; Elsweiler et al., 2017; Freyne & Berkovsky, 2010). There also seems to be an strong demand for CAs in the kitchen, e. g. Google Home Devices were used for more than 16 million recipes during 2018th Christmas season, passing one million on Christmas day alone (Huffman, 2019).

Adapting BERT for the cooking domain is hoped to increase its natural language understanding for this domain and with it the performance on downstream tasks relevant for kitchen conversational agents, which is an important step towards building a truly conversational system. The concrete contributions of this thesis include:

* the introduction of CookBERT, a domain adapted BERT model for the cooking domain,
* the evaluation of CookBERT on three conversational agent relevant tasks, as well as a comparison to the similar FoodBERT and standard BERT model,
* additional evidence that further pretraining is a viable strategy to obtain domain specific language representation models in a fast and cheap way

In doing so, the thesis adds to different fields in the literature

* In doing so, the thesis adds to different fields in literature where Language representation models .
* Moreover, the results are insightful for the future development of conversational agents (not only for the kitchen) that have pretrained language representations integrated in their pipeline, as they show that the adaption for the desired domain that the assistant is going to be applied, can be useful to increase natural understanding, which is the heart of such applications. Wenn input nicht richtig verstanden werden kann, kann das system logischerweise auch das Informationsbedürfniss nicht korrekt befriedigt werden
* Even though the

# Related Work

To set the context for this thesis and motivate the research question as well the methodological decisions taken, this chapter covers the background and related work from research contributions across diverse fields of computer and information science, ranging from conversational agents to word embeddings to the recently popular transformer neural networks. The chapter is arranged as follows:

* Section 2.1 enthält den Background zu BERT und seine core concepts, die zum Verständnis diese Arbeit beitragen
* Section

## Conversational Agents

* Was sind Conversational agents
* Welche Arten gibt es/ wie kann man sie einteilen

## BERT Prerequisites

Vor encoder-decoder architecture noch word embeddings reinbringen

### Encoder-Decoder Architecture



Figure 1: Encoder-decoder architecture (taken from Zhang et al. (2021))

* State durch context vector austauschen

Encoder-decoder (Sutskever et al., 2014) is a specific neural network architecture that was proposed to tackle sequence-to-sequence problems. The power of this architecture lies in its ability to map sequences of variable-length to each other, which was previously not possible with the existing neural network architectures. Since human language can be viewed as a sequence of words, the encoder-decoder architecture is very well suited for this and is used, for example, in text summarization (Nallapati et al., 2016), machine translation (Wu et al., 2016), speech recognition (Bahdanau et al., 2016) or video captioning (Venugopalan et al., 2015).



Figure 2: Machine translation illustrated as a sequence-to-sequence learning problem with a RNN encoder and a RNN decoder (taken from Zhang et al. (2021))

* Hidden states in die encoder states eintragen (also auf die Pfeile zwischen den Encoder blöcken) + context vector kennzeichnen

The architecture consists of two major components, illustrated in figure …. The first one is the encoder, which processes every item of the variable-length input sequence and captures it into a single, fixed dimensional representation vector, also known as context vector, which acts as the final hidden state of the encoder. This context vector is subsequently fed into the second component, the decoder, which then generates a variable-length output sequence. As the Encoder and decoder blocks are typically implemented with a recurrent neural network (RNN) architecture, especially LSTM (Hochreiter & Schmidhuber, 1997), the input processing of the encoder and the output generation of the decoder is done step by step in an auto-regressive manner, meaning that the they use information from previous steps to output the hidden state and predicted word, respectively (see Bild 2). While the default encoder-decoder architecture works fine for short input sequences, it struggles with longer ones, because it’s difficult for the encoder to compress all the contextual information of the long sequence into a single fixed size vector, which thus motivates optimization by means of “attention".

### Attention



Figure 3: Attention visualized in practical use with machine translation. X-axis and y-axis correspond to the words in the source sentence (English) and generated translation (French), respectively. Pixels show indicate the focus of attention in grayscale (Taken from Bahdanau et al. (2014)).

Attention was introduced and refined by Bahdanau et al. (2014) and Luong et al. (2015), respectively. It is a technique that allows sequence-to-sequence models to better deal with long input sequences, as it enables the network to focus only on certain parts of the input sequence as needed. Thus, the model can keep track of all inputs that are believed to be crucial for determining the output. Figure 3 illustrates the impact of attention in practical use with machine translation. To correctly translate the English sequence “European Economic Area” into French, the order of the words needs to be reversed. By paying attention to the respective proper input words, the model is able to generate the desired output.

In order to integrate attention into an encoder-decoder model, two aspects need to be changed. On the one hand, the encoder not only passes its last hidden state (the context vector) to the decoder, but all of its hidden states that were output when processing the input sequence step by step. Note that each of these hidden states is specifically associated with a particular word of the input sequence, namely the word that was being processed at the time. The decoder, on the other hand, assigns a score to these handed over hidden states and multiplies it by its softmaxed score to boost the hidden states with high, and tone down the hidden states with low scores. The scoring is done for each step of the decoder. (Alammar, 2018b)

A more recent form of attention is self-attention. Self-attention has been proposed in several papers, in which it is also sometimes referred to as intra-attention (Cheng et al., 2016; Lin et al., 2017; Parikh et al., 2016; Paulus et al., 2017; Vaswani et al., 2017). It differs from the standard attention mechanism in that it applies attention within the same sequence, rather than across two different sequences. When processing each word of the input sequence, attention is paid to other input words that are assumed to be relevant for the “understanding” of the current word. Roughly speaking, self-attention is a mechanism to enrich the currently processed word with contextual information from its environment, which is particularly useful when facing disambiguation or for the resolution of coreferences and pronouns. (citation)

A deeper insight into the inner workings of self-attention and also multi-head self-attention in the context of the famous Transformer architecture is given in the next section.

### Transformers



Figure 4: Transformer architecture (Taken from Vaswani et al. (2017, p. 3))

The Transformer was proposed in the well-known “Attention is All You Need” paper by Vaswani et al. (2017). It is a special network architecture, namely the first to be solely based on the attention mechanism, not combining it with recurrence nor convolution. Besides the fact that the Transformer achieves superior performance in machine translation, it is above all the good parallelizability and the associated significant speed boost when training deep learning models that makes it stand out from previous approaches.

Architecture

Overall, the Transformer follows the encoder-decoder architecture, as it consists of an encoding and decoding component, shown on the left and right side in figure 4, respectively. The encoding component is a stack of six encoders, and the decoding component a stack of six decoders. Each of the encoders can in turn be broken down into a multi-head self-attention sub-layer (detailed explanation follows below) and a simple fully connected feed-forward (no recurrence) sub-layer. Apart from an extra multi-head self-attention sub-layer, the decoders are built the same. The additional layer of every decoder each accept the output of the last encoder of the encoder stack and use it to help the decoder focus on appropriate places in the input sequence. Since the architecture does not rely on recurrence nor convolutions, it adds positional encodings that encode the necessary information about the position of the words and distance to other words in the input sequence to the encoder and decoder inputs. As can be seen in figure 4, there are also shortcut connections, so-called skip connections, for each sublayer in the encoders and decoders to the next normalization layer. These skip connections serve for both forward and backward passes and are mechanisms to avoid the problem of vanishing and exploding gradients. We’ve seen similar workarounds in LSTM (section 2.2.4) by preventing the repeated flow of gradients through non-linear activation functions such as sigmoid and tanh – effects of inserting domain vocabulary seite 30

The normalization is applied to mitigate the problem known as covariate shift, which refers to the distributions of the training and test sets being different. This disparity is reduced by fixing the mean and the variance of the summed inputs of the layer.- effects of inserting domain vocabulary – seite 30 (Alammar, 2018a; Rush, 2018; Vaswani et al., 2017)

****Self-Attention****

The self-attention mechanism of the Transformer architecture, which is more precisely referred to as *scaled dot-product attention* by Vaswani et al. (2017), consists of six steps and can be illustrated with the abstracted concept of *query*, *key,* and *value* vectors. In step one, these three vectors are initially created by multiplying the embedding vector by three weight matrices that were learned during the training of the network (Note that only the first encoder starts with the original embeddings, all other encoders start with the output of the preceding encoder). In step two, the scaled dot-product attention score is calculated for every word of the input sequence by taking the dot product between the query vector of the currently processed word and the respective key vectors of the other words. The next two steps include the division of the calculated score by a fixed number (typically the square root of the key vector dimension) and feeding it through a softmax function to get more stable gradients and normalize the score, respectively. The resulting softmax score that is now assigned to each word in the input can be seen as the weight that each input word has on grasping the actual meaning of the currently processed word. In the fifth step, each value vector is multiplied by its softmax score and then all value vectors are summed to produce the final output of the scaled dot-product attention layer, which is the contextualized embedding for the currently processed word. Note that in the actual implementation, matrices are used for calculation, since they enable faster processing. This means that all embeddings are packed into a single input matrix, with each row corresponding to a word of the input sequence. After calculating the key, value and query matrices, the output of the scaled dot-product attention layer can be calculated with a shortened equation:

where , and denote the query, key, and value matrix, respectively, and denotes the key dimension. The general flow of information is visualized again in figure 5. (Alammar, 2018a; Rush, 2018; Vaswani et al., 2017)

Multi-head self-attention



Figure 5: Comparison of scaled dot-product attention (left) and multi-head scaled dot-product attention (right). (Taken from Vaswani et al. (2017, p. 4))

Multi-head self-attention improves the scaled dot-product attention mechanism, as it runs the mechanism multiple times in parallel, each with different query/key/value weight matrices that were learned during training the network. In case of the Transformer which uses eight attention heads, this results in eight output matrices. To ensure that the upcoming feed-forward layer receives its desired input, namely a single matrix, the eight scaled dot-product attentions are concatenated and linearly transformed. Vaswani et al. (2017) claim that this multi-head approach allows to “jointly attend to information from different representation subspaces at different positions” (p. 4) and thus exceeds the performance of single-head attention. The general information flow of the multi-head approach compared to a single-head scaled dot-product attention layer is visualized again in figure 5. (Alammar, 2018a; Rush, 2018; Vaswani et al., 2017)

## BERT

Blabla bla

Transformer

BERT ist insofern kein traditional Language Model, da es nicht basierend auf den Vorherigen Kontext/ Wörtern die Wahrscheinlichkeit für das nächste Wort in der Sequenz bestimmen kann.

## Cooking datasets

Despite the fact that cooking has recently received some attention for NLP research, the number of sophisticated datasets in this domain is rather small. This shrinks even more when only considering datasets that are somehow relevant for conversational AI/agents and thus could be used to train and test CookBERT. Available datasets that meet this criterion and are therefore utilized in this thesis are presented in the next sections.

### RecipeNLG

RecipeNLG (Bień et al., 2020) is a cooking recipe dataset for semi-structured text generation. It contains over 2.2 million distinct recipes and is assumed to be the largest publicly available dataset for the cooking domain. RecipeNLG builds upon the preceding Recipe1M+ dataset (Marin et al., 2019) and extends it with over one million cleaned, deduplicated recipes scraped from multiple cooking websites. Each entry of the dataset contains the following information: the title of the recipe, a list of ingredients and quantities, a list of instructions, the link to the recipe, information about its source (gathered or originating from Recipe1M+ dataset) and a list of automatically extracted food entities. Bień et al. (2020) also trained two GPT-2 language models on their and the Recipe1M+ dataset, respectively, in order to compare their ability to generate recipes only based on food entities. They found that the model trained on RecipeNLG both made fewer linguistic errors and performed better for all translation metrics than the model trained on Recipe1M+.

### Cookversational search



Figure 6: Frequencies of level 1 information needs (Taken from Frummet et al. (2021))

Cookversational search is the resulting dataset of the work Frummet et al. (2021), in which the information needs that arise during cooking were examined (see section …).

* Der Datensatz ist für die Aufgabe der Text-classification gedacht
* Er enthält insgesamt 2675 user utterances, für die das zugrundeliegende Informationsbedürfnis klassifiziert werden soll. 🡪 kuratierter datensatz
* Die utterances gibt es in Deutsch (Originalsprache) und auf englisch (automatically translated with DeepL)
* Für jede utterance the dataset contains a lot of information, including 5 different information need levels of information need for each utterance
* Im Paper wurden zudem noch unterschiedliche Machine Learning Ansätze und deren Performance bei der Klassifikation der Informationsbedürfnisse auf level 1 Ebene (enthält 11 Klassen) und für unterschiedliche Conditionen (utterance without context, utterance with 1 prev turn, utterance with all prev turns) verglichen.
* They found that the GermanBERT model (cite) performed best among the other approaches
* When looking at the conditions, where the utterance was prepended with 1 or all previous utterances, no significant differences between the models performance was found.
* During the coding process, they found context to be an important factor for identifying information needs and thus tested for the three conditions
* They applied multiple baseline () and BERT-based models (GermanBERT and two multilingual BERT models)
* Similar to our results, Ren et al. [68] and Aliannejadi et al. [5] showed that including more context, i.e. conversational history, improves the classification performance of the current turn.
* Including the context in the form of previous turns, however, significantly improved results in competence-oriented needs Cooking technique and Preparation.
* Beides Seite 21
* Auch wenn es beim best-performendsten Model (GermanBERT, welches dann für die 3 conditions applied wurde) overall keine signifikanten unterschiede bei der Performance für die 3 conditions gibt, kann das hinzufügen von kontext in form von previous turns trotzdem für bestimmte information needs signifikant die Performance verbessern. Und wird deshalb als sinnvoll erachtet
* Das aber alles eher schon vorher in related work packen und nicht hier beim datensatz

### DoQA

* Dataset mit 2437 dialogen und 10.917 QA pairs zu den drei domänen Cooking, Travelling, Movies, wobei Cooking mit insgesamt 7320 samples den größen Anteil hat.
* The dialogues were collected from three Stack Exchange sites using the wizard of oz method with crowdsourcing.
* “Compared to previous work, DoQA comprises well-defined information needs, leading to more coherent and natural conversations with less factoid questions.”
* “DoQA makes the following contributions. Firstly, contrary to made-up reading comprehension tasks, DoQA reflects real user needs, as defined by a topic in an existing FAQ. Good results on DoQA are of practical interest, as they would show that effective conversational QA interfaces to FAQs can be built. Secondly, for the same reason, the conversations in DoQA are more coherent, natural and contain less factoids than other datasets, as shown by our analysis. Thirdly, the IR scenario and the multiple domains make DoQA more challenging and realistic. Table 1 summarizes the characteristics of DoQA.”
* Experiment:
  + Haben nur die extractive information genutzt (abstractive nicht)
  + Haben zwei Models getestet : BERT normal und BERT + HAE
  + Haben beide models für 4 settings getestet (native, zero-shot, transfer, transfer-all)
  + BERT+HAE outperformed BERT normal in allen settings deutlich, showing that doqa is indeed a conversational dataset, where questions and asnwer history needs to be modelled

### FoodBase

## Summary and Key Differentiators

# Methodology

## Preparing the Data for DAPT

* Das und nächste Sektion unter einem Punkt “Prerequisites for DAPT” zusammenfassen
* Für das DAPT wird der RecipeNLG Datensatz (siehe Section in related work verwendet)
* Da für das Pretraining natürliche Sprachdaten/ Textdaten und nicht nur einzelne kontextfreie Wörter benötigt werden, werden vor allem die Rezeptinstruktionen verwendet.
* Um DAPT zu ermöglichen wurden die Daten, die aktuell noch in einzelne Instruktionen aufgeteilt waren zusammengefügt.
* jedes Rezept wird als eigenes, unabhängiges Dokument betrachtet und daher alle Rezeptinstruktionen zusammengefügt und jeweils in eine Zeile einer Textdatei geschrieben (was später furs Pretraining sinnvoll ist, da Zeile für Zeile angeguckt wird)

In order to adapt BERT for the cooking domain, the RecipeNLG dataset (Bień et al., 2020) was utilized.

* Mit über 2.2 Millionen unqie recipes is it assumed to be the largest publicly available dataset in the domain, und damit knapp 2\* so groß wie der vorgänger, Recipe1M+
* RecipeNLG is an expension of Recipe1M+, which Pellegrini et al. (2021) utilized to create their FoodBERT.
* RecipeNLG enthält gesäuberte, deduplizierte Rezepte
* The dataset contains
* Title: Rezepttitel
* Ingredients: Zutaten mit mengenangaben
* Directions: List of Instruktionen 🡪 das habe ich verwendet
* Link: link zum Rezept
* Source: Gathered (74%) oder von Recipes1M (26%)
* NER: named food entities; extracted mit einem NER
* However, only the instructions were of interest for the unsupervised pretraining.
* The im imperative formulierten instructions liegen als liste von einzelnen Anweisungen vor, ein Beispiel ist in Abb. Gegeben.
* The quality and influence of the instruction characteristics are discussed in section 4.4 (limitation section)
* Overall statistics of the instruction data can be found in table … (Anzahl Rezepte, durchschnittliche Instructions, Anzahl Wörter gesamt, Durchschnittliche Anzahl pro Rezept)

Mit 2.2 mio unique recipes ist es mehr als doppelt so groß wie recipe1M+

* Wie in section … schon erklärt, sind führen generell mehr Daten bei DAPT auch zu besserer Performance (wie eigentlich immer in machine learning). Deshalb wird ein möglichst großer Korpus für DAPT ausgewählt.
* FoodBERT verwendet den Recipe1M Datensatz bzw. Die Instruktionen davon
* In meinem Ansatz wird ein noch größerer Korpus verwendet
* Datenmengen bei DAPT in der Literatur
* FinBERT: TRC2-financial, 46.143 documents, 29 million words, 400k sentences
* HateBERT: RAL-E, 1.478.348 messages, 43.379.350 million tokens,
* BioBERT: PubMed Abstracts = 4.5B words + PMC Full-text articles 13.5B words
* CSBERT: 40.505.050 dialogues and 317.093.459 turns coming from all available customer service intents
* TOD BERT: 100.707 Dialoge, 1.388.152 Utterances von verschiedenen task-oriented dialogue Datensätzen
* MenuNER: YELP Dataset (only reviews from the restaurant category), 15.000.000 sentences
* Pretraining BERT on domain resources for short answer grading: Textbooks und QAs mit 1.1m, 0.6m und 1.3m Wörtern
* Außerdem gibt es Paper, die sagen, dass mehr pretraining Data (When do you need billions of words of pretraining data) bzw. Mehr DAPT data (sinnvoll ist)

## Analyzing Domain Similarity



Before the actual DAPT, the similarity of the target domain (cooking) and BERT’s pretraining domain was analyzed. The approach for the analysis is adopted from Gururangan et al. (2020) and quantifies the domain similarity based on the vocabulary overlap of the pretraining corpora. Therefore, RecipeNLG, Recipe1M+ and the WikiBook from CookBERT, FoodBERT and the standard BERT respectively were used for corpus from BERT pretraining were used for the analysis. As BERT’s original pretraining data is not publicly distributed, a Wikipedia dump (Merity et al., 2016)(515MB) and randomly sampled books from the “Homemade BookCorpus” (Kobayashi, 2018)(444MB) were used to reconstruct a similar corpus. From RecipeNLG and Recipe1M+, the recipe instructions were used as corpus data (1GB and 619MB respectively). For each corpus, the vocabulary, consisting of unigrams (after lowercasing and removal of stopwords and punctuation) was then created for each of the three corpora.

The vocabulary overlap between the corpora was then determined based on the 10000 most frequent unigrams of each domain and is illustrated in Fig 4. It shows a strong overlap between Recipe1M+ and RecipeNLG, which is not surprising given the fact that both corpora are from the cooking domain and Recipe1M+ is a subset of RecipeNLG. In contrast, the overlap between the WikiBooks corpus and the two cooking corpora is quite small, emphasizing the data shift between the cooking domain and the general text domain. Furthermore, this simple analysis indicates the degree of benefit to be expected by adapting BERT for the cooking domain, as the potential for DAPT is higher, the more dissimilar the domains (Gururangan et al., 2020, p. 3).

* Unbedingt noch „A survey on Transfer Learning“ angucken!!

## Domain Vocabulary Insertion

|  |  |
| --- | --- |
| **Word** | **Tokenized representation** |
| baguette | bag ##uet ##te |
| cranberry | cr ##an ##berry |
| caramelized | cara ##mel ##ized |
| zucchini | zu ##cchi ##ni |
| preheat | pre ##hea ##t |
| tortilla | tor ##till ##a |
| eggplant | egg ##pl ##ant |

The influence of out-of-vocabulary words was proven to have negative influence on NLP models (<https://aclanthology.org/P11-2071.pdf,> <https://arxiv.org/pdf/1802.02614.pdf>). Even though BERT deals quite well with OOV words by splitting them up into smaller subtokens, kann die insertion of domain specific vocabulary as an adaption strategy auch hier zu einer besseren Performance führen (SciBERT und ExBERT).

Auch für die Kochdomäne scheint dieser Schritt sinnvoll zu sein, da viele gängige kochspezifische Vokbalen nicht im BERT Vokabular enthalten sind und dementsprechen in “unrepresentative” Subtokens zerlegt werden, as shown in figure

Um das Vokabular von CookBERT zu erweitern, wurden alle Wörter aus dem in Sektion 3.2 erstellten RecipeNLG Vokabular, die mindestens 1000 mal vorkamen und noch nicht Teil des BERT Vokabulars sind, hinzugefügt. Da die weights für neu eingefügten Wörter neu initialisiert werden, wurde darauf geachtet, dass sie einigermaßen häufig im Corpus vorkommen, um gute Repräsentationen lernen zu können.

Insgesamt wurden 1229 kochspezifische Wörter zum Bereits existierenden Vokabular hinzugefügt, was zu einer neuen Gesamtgröße von 30000 Vokabeln führt.

* Alle Wörter mit einer Häufigkeit von mindestens 1000, und die noch nicht im BERT vocab enthalten waren 🡪 Das hat den grund, dass die Wörter häufig genug vorkommen, um gute Repräsentationen zu erlernen, nachdem die Gewichtungen für diese quasi von 0 initialisiert werden müssen. Kommen sie zu selten vor, dann kann dies evtl. nicht richtig erlernt werden
* Wie in der vorherigen Section schon erwähnt wurden jedes Rezept erst tokenisiert, lowercased, und stop-words und punctuation wurde entfernt.
* Insgesamt wurden 1229 Wörter zum bestehenden Vokabular hinzugefügt
* Irgendwo noch mit reinbringen, wie FoodBERT das gemacht hat: die hatten eien Liste mit Zutaten und haben dann alle, die öfter als 10 mal vorkamen, zum Vokabular hinzugefügt. 🡪 Allerdings: es gibt auch viele Wörter, die nicht zutaten sind, die im Kochjargon vorkommen (skillet, …), diese werden aber vernachlässigt
* Fügen alle Zutaten hinzu, wenn diese mind. 10 mal vorkommen. Allerdings nehme ich an, dass dies nicht ausreichend ist, da die repräsentationen komplett neu initialisiert wurden und erst trainiert werden müssen, wobei 10 beipsiele nicht ausreichen
* With the insertion of our custom legal vocabulary, the tokenization provides different segmentation of words affecting the input sequence for BERT and we expect to slightly improve the performance in the same way SciBERT did by creating their own SciVocab (subsection 3.2.2).
* Effects of inserting domain vocabulary (S. 52)

## DAPT

* BERT kommt in mehreren Ausführungen (BERT large, BERT base, BERT base cased, BERT base uncased, …)
* BERTBASE: (L=12, H=768, A=12, Total parameters = 110M)
* BERTLARGE (L=24, H = 1024, A=16, Total Parameters = 340M)
* L: number of layers, H = hidden size, A = number of atterntion heads
* Als ausgangspunkt wurde der BERT base uncased checkpoint verwendetDAPT: **BERTBASE\_UNCASED, da**
* Cased model würde zwischen Bread und bread unterscheiden, obwohl beide dasselbe Konzept sind. Während in anderen Sprachen die groß und kleinschreibung eine wichtigere Rolle spielt (z. b. German) ist das im englischen moistens nicht wirklich der Fall, weswegen uncased verwendet wurde
* BERTbase, da obwohl BERTlarge yields better results, aufgrund der erhöhten Komplexität ein deutlich höherer Ressourcenaufwand
* If your task has a large domain-specific corpus available (e.g., "movie reviews" or "scientific papers"), it will likely be beneficial to run additional steps of pre-training on your corpus, starting from the BERT checkpoint. (https://github.com/google-research/bert#pre-training-tips-and-caveats)
* The learning rate we used in the paper was 1e-4. However, if you are doing additional steps of pre-training starting from an existing BERT checkpoint, you should use a smaller learning rate (e.g., 2e-5).
* Um DAPT durchzuführen, wurde das Model weiter auf den MLM tasks trainiert. NSP wurde nicht verwendet, da nicht hilfreich (siehe RoBERTa und CamemBERT)
* Result of DAPT 🡪 Beispielsätze zur Demonstration einfügen

## Implementation Details

* Evtl als unterpunkt zu 3.4 packen oder ans ende der Methodology als eigenen Puntk, in dem dann die Implementation details sowohl zum DAPT als auch furs Finetuning stehen.
* Learning rate, epochen, Model startpunkt, …
* Dauer des Learning vorgangs
* Verwendete Library: Huggingface
* Verwendete Umgebung: Google Colab
* GPU: P100 GPU von Google Colab+

## Finetuning

### Intent Classification

* Multi-class classification problem
* Siehe paper von Frummet für Vorgehen (an dem orientiere ich mich eben)
* Datensatz von Frummet
* Alles so wie Frummet gemacht
* 85% train, 15% test
* No resampling
* No stopword removal
* Stratified sampling for 10 fold cross validation
* To avoid catastrophic forgetting: lower learning rate of 2e-5
* Training for 4 epochs, dropout probability of 10%, batch\_size 32
* Early stopping was included
* Wegen computing limitations wurde eine maximale Sequ. Length von 256 verwendet. D.h. wenn mehrere Turns mit angehänt wurden, wurden nur die letzten 256 tokens verwendet.
* Auswertung mit drei contexten:
  + 1. No context
    2. 1 prev turn
    3. All prev turns
* Anders als Frummet gemacht:
* Frummet hat 11 binary classifiers mit jeweils einem classificationHead der Dimension 768,2. Ich habe nur einen classifier mit classification Head mit dimension 768,11.
* Class weights were adjusted by FARMs datasilo 🡪 evlt auch machen, siehe <https://discuss.huggingface.co/t/class-weights-for-bertforsequenceclassification/1674/7>

### Named Entity recognition

### Question Answering

# Evaluation

## Multi-class Classification

## Named Entity Recognition

## Question Answering

# Discussion

# Limitations

* Auch wenn diese Arbeit vielversprechende Resultate bezüglich eines robusten/ sophisticated BERT Models für die Kochdomäne liefert, gibt es einige Limitierungen, die die zu beachten sind.
  + Aus ressourcengründen wurde nur BERT model als Basis hergenommen 🡪 inzwischen existieren mehrere Verbesserte/ optimierte Modelle, die BERT outperformen
  + Es wurde nur der Aspekt der Domänenadaption angeguckt. Andere Aspekte, die für Conversational Agents und die natürliche Sprache, die diese begegnen (Dialoge) werden nicht berücksichtigt.
  + Vielle ist BERT mit history embeddings als ausgangsmodell besser
  + Viell. Ist ConvBERT, der speziell schon auf Konversationsdaten vortrainiert wurde besser als standard BERT
  + Es ist unklar, wie sich das Einfügen von zusätzlichen Vokabeln auf die Perf ausgewirkt hat (Vorherige Arbeiten haben zwar schon positive Effekte festgesetellt)
  + Cooking recipes eigenschaften ; as mentioned multiple times throughout this thesis, data scarcity is a big problem, auch für die cooking domain: kein gutter Datensatz verfügbar, der natürliche Sprache in der Küche enthält
* <https://jurnal.polban.ac.id/inggris/article/view/3467>
* Meistens im Imperative (selten im declarative) formuliert
* Sind oft keine Grammatikalischen Sätze, Wörter, v.a. Artikel fehlen (“Add egg and beat well”)
* Sind nur Rezepte, und damit keine wirkliche natürliche Sprache. Besser wäre vielleicht natürliche Konversationen übers Kochen zu verwenden (z.B. via Podcasts oder Untertitel)
* Kochdomäne umfasst vielleicht mehr, als nur die Rezeptdaten
* Enthalten Abkürzungen (tbsp, oz., hrs)

# Conclusion

* Vorschlag: andere Datenquelle zum Pretrainnen hernehmen, welche näher an der natülichen Sprache ist 🡪 Kommentare von Rezepten, Koch FAQs, Untertitel von Kochshows, …

# Bibliography

Alammar, J. (2018a). *The Illustrated Transformer [Blog post]*. https://jalammar.github.io/illustrated-transformer/

Alammar, J. (2018b). *Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention) [Blog post]*. https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Angara, P., Jimenez, M., Agarwal, K., Jain, H., Jain, R., Stege, U., Ganti, S., Müller, H. A., & Ng, J. W. (2017). Foodie fooderson a conversational agent for the smart kitchen. *CASCON*, 247–253.

Bahdanau, D., Cho, K., & Bengio, Y. (2014, September 1). *Neural Machine Translation by Jointly Learning to Align and Translate*. http://arxiv.org/pdf/1409.0473v7

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. (2016, March). End-to-end attention-based large vocabulary speech recognition. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4945–4949. https://doi.org/10.1109/ICASSP.2016.7472618

Bickmore, T. W., Caruso, L., & Clough-Gorr, K. (2005). Acceptance and usability of a relational agent interface by urban older adults. *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, 1212–1215. https://doi.org/10.1145/1056808.1056879

Bień, M., Gilski, M., Maciejewska, M., Taisner, W., Wisniewski, D., & Lawrynowicz, A. (2020). RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation. *Proceedings of the 13th International Conference on Natural Language Generation*, 22–28. https://aclanthology.org/2020.inlg-1.4

Brandtzaeg, P. B., & Følstad, A. (2017). Why People Use Chatbots. In I. Kompatsiaris, J. Cave, A. Satsiou, G. Carle, A. Passani, E. Kontopoulos, S. Diplaris, & D. McMillan (Eds.), *Lecture Notes in Computer Science. Internet Science* (Vol. 10673, pp. 377–392). Springer International Publishing. https://doi.org/10.1007/978-3-319-70284-1\_30

Chao, G.-L., & Lane, I. (2019, July 6). *BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer*. http://arxiv.org/pdf/1907.03040v1

Chen, Q., Zhuo, Z., & Wang, W. (2019). *BERT for Joint Intent Classification and Slot Filling*. http://arxiv.org/pdf/1902.10909v1

Cheng, J., Dong, L., & Lapata, M. (2016, January 25). *Long Short-Term Memory-Networks for Machine Reading*. http://arxiv.org/pdf/1601.06733v7

Chu, J. (2021, September 24–26). Recipe Bot: The Application of Conversational AI in Home Cooking Assistant. In *2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)* (pp. 696–700). IEEE. https://doi.org/10.1109/ICBASE53849.2021.00136

Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., & Zhou, M. (2017). Superagent: A customer service chatbot for e-commerce websites. *Proceedings of ACL 2017, System Demonstrations*, 97–102.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised Sequence Learning. *Advances in Neural Information Processing Systems*, *28*, 3079–3087. http://arxiv.org/pdf/1511.01432v1

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. http://arxiv.org/pdf/1810.04805v2

Elsweiler, D., Harvey, M., Ludwig, B., & Said, A. (2015). Bringing the "healthy" into Food Recommenders. *DMRS*, 33–36.

Elsweiler, D., Trattner, C., & Harvey, M. (2017). Exploiting food choice biases for healthier recipe recommendation. *Proceedings of the 40th International Acm Sigir Conference on Research and Development in Information Retrieval*, 575–584. https://doi.org/10.1145/3077136.3080826

Freyne, J., & Berkovsky, S. (2010). Intelligent food planning: personalized recipe recommendation. *Proceedings of the 15th International Conference on Intelligent User Interfaces*, 321–324. https://doi.org/10.1145/1719970.1720021

Frummet, A., Elsweiler, D., & Ludwig, B. (2021, December 9). *"What can I cook with these ingredients?" -- Understanding cooking-related information needs in conversational search*. http://arxiv.org/pdf/2112.04788v2

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., & Kreuz, R. (1999). AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, *1*(1), 35–51. https://doi.org/10.1016/S1389-0417(99)00005-4

Gururangan, S., Marasović, A., Swayamdipta, S., Lo Kyle, Beltagy, I., Downey, D., & Smith, N. A. (2020, April 23). *Don't Stop Pretraining: Adapt Language Models to Domains and Tasks*. http://arxiv.org/pdf/2004.10964v3

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Howard, J., & Ruder, S. (2018, January 18). *Universal Language Model Fine-tuning for Text Classification*. http://arxiv.org/pdf/1801.06146v5

Huffman, S. (2019). *Here’s how the Google Assistant became more helpful in 2018.* Google. https://www.blog.google/products/assistant/heres-how-google-assistant-became-more-helpful-2018/

Kobayashi, S. (2018). *Homemade BookCorpus*. https://github.com/BIGBALLON/cifar-10-cnn

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). Biobert: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics (Oxford, England)*, *36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017, March 9). *A Structured Self-attentive Sentence Embedding*. http://arxiv.org/pdf/1703.03130v1

Luong, M.-T., Pham, H., & Manning, C. D. (2015, August 17). *Effective Approaches to Attention-based Neural Machine Translation*. http://arxiv.org/pdf/1508.04025v5

Marin, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., Weber, I., & Torralba, A. (2019). Recipe1m+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 187–203. https://doi.org/10.1109/TPAMI.2019.2927476

McTear, M. (2020). Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots. *Synthesis Lectures on Human Language Technologies*, *13*(3), 1–251. https://doi.org/10.2200/S01060ED1V01Y202010HLT048

Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016, September 26). *Pointer Sentinel Mixture Models*. http://arxiv.org/pdf/1609.07843v1

Nallapati, R., Xiang, B., & Zhou, B. (2016). Sequence-to-sequence rnns for text summarization.

Ni, L., Lu, C., Liu, N., & Liu, J. (2017). MANDY: Towards a Smart Primary Care Chatbot Application. In J. Chen, T. Theeramunkong, T. Supnithi, & X. Tang (Eds.), *Communications in Computer and Information Science. Knowledge and Systems Sciences* (Vol. 780, pp. 38–52). Springer Singapore. https://doi.org/10.1007/978-981-10-6989-5\_4

Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016, June 6). *A Decomposable Attention Model for Natural Language Inference*. http://arxiv.org/pdf/1606.01933v2

Paulus, R., Xiong, C., & Socher, R. (2017, May 11). *A Deep Reinforced Model for Abstractive Summarization*. http://arxiv.org/pdf/1705.04304v3

Pellegrini, C., Özsoy, E., Wintergerst, M., & Groh, G. (2021, February 11–13). Exploiting Food Embeddings for Ingredient Substitution. In *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies* (pp. 67–77). SCITEPRESS - Science and Technology Publications. https://doi.org/10.5220/0010202000670077

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, February 15). *Deep contextualized word representations*. http://arxiv.org/pdf/1802.05365v2

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training.*

Rush, A. M. (2018, July). The Annotated Transformer. *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 52–60. https://doi.org/10.18653/v1/W18-2509

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Dvances in Neural Information Processing Systems*, *27*.

Vakulenko, S., Longpre, S., Tu, Z., & Anantha, R. (2021). Question Rewriting for Conversational Question Answering. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 355–363. https://doi.org/10.1145/3437963.3441748

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention Is All You Need*. http://arxiv.org/pdf/1706.03762v5

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., & Saenko, K. (2015). Sequence to sequence-video to text. *Proceedings of the IEEE International Conference on Computer Vision*, 4534–4542.

Voskarides, N., Li, D., Ren, P., Kanoulas, E., & Rijke, M. de (2020). Query Resolution for Conversational Search with Limited Supervision. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 921–930. https://doi.org/10.1145/3397271.3401130

Weizenbaum, J. (1966). ELIZA-a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45.

Winkler, R., Hobert, S., Salovaara, A., Söllner, M., & Leimeister, J. M. (2020). Sara, the Lecturer: Improving Learning in Online Education with a Scaffolding-Based Conversational Agent. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3313831.3376781

Wu, Y., Schuster, M., Chen, Z., Le V, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., . . . Dean, J. (2016, September 26). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. http://arxiv.org/pdf/1609.08144v2

Xu, L., Zhou, Q., Gong, K., Liang, X., Tang, J., & Lin, L. (2019). End-to-End Knowledge-Routed Relational Dialogue System for Automatic Diagnosis. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 7346–7353. https://doi.org/10.1609/aaai.v33i01.33017346

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021, June 21). *Dive into Deep Learning*. http://arxiv.org/pdf/2106.11342v2

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Proceedings of the IEEE International Conference on Computer Vision*, 19–27.

Anhang A: Bausteine wissenschaftlicher Arbeiten

## A1 Theoretische Arbeit

1. Fragestellung (Ziele, Motivation)
2. Überblick über Stand der Forschung und Technik (dabei Bewertung der Ansätze, Beispiele, Identifikation von Defiziten)
3. Synthese: Erstellung einer Gesamtschau (allgemeine Prinzipien, Beschreibung einer eigenen Sicht auf das Problem, Formulierung von Empfehlungen )
4. Zusammenfassung (Was wurde in der Arbeit erreicht, Erklärung des Nutzens für andere)
5. Ausblick (optional)

## A2 Konstruktive Arbeit

1. Problemstellung (Ziele, Ausgangspunkt, Vorgesehener Benutzerkreis, Bedürfnisse der Benutzer)
2. Stand der Forschung und Technik (Bisherige Lösungen, Defizite)
3. Eigenes Konzept (Lösungsansatz, allgemeines Prinzip, Werkzeuge z.B. Programmiersprachen )
4. Vorgehensweise (Beschreibung der durchgeführten Arbeitsschritte)
5. Ergebnis (Vorstellung des System z.B. Screenshots mit Erläuterungen)
6. Evaluation des System (optional, was soll evaluiert werden, welche Methode, Ablauf, Ergebnisse)
7. Zusammenfassung (Was wurde in der Arbeit erreicht; Erklärung des Nutzens für andere)
8. Ausblick (optional)

## A3 Empirische Arbeit

1. Fragestellung der Arbeit (Was soll untersucht werden, warum)
2. Stand der Forschung und Technik (Bewertung der Untersuchungs-Ansätze und Ergebnisse, Identifikation von Defiziten)
3. Präzisierung der Fragestellung (Hypothesen)
4. Untersuchungsmethodik
5. Untersuchungsablauf (Untersuchungsmaterial, Raum, Probandenrekrutierung etc.)
6. Ergebnisse (Darstellung der Ergebnisse in sinnvoller Reihenfolge, Gesamtüberblick, Einzelergebnisse z. B. geordnet nach Testcases)
7. Zusammenfassung (Was wurde erreicht, Rückbezug zu Zielen, Hypothesen, Nutzen, Erkenntnisse für weitere Untersuchungen)
8. Ausblick (optional)

Erklärung zur Urheberschaft

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie alle Zitate und Übernahmen von fremden Aussagen kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version sind identisch.

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Unterschrift |

Erklärung zur Lizenzierung und Publikation dieser Arbeit

**Name:** Pascal Strobel

**Titel der Arbeit:** CookBERT

In der Regel räumen Sie mit Abgabe der Arbeit dem Lehrstuhl für Medieninformatik nur zwingend das Recht ein, dass die Arbeit zur Bewertung gelesen, gespeichert und vervielfältigt werden darf. Idealerweise liefern Seminararbeiten, Projektdokumentationen und Abschlussarbeiten aber einen Erkenntnisgewinn, von dem auch andere profitieren können. Wir möchten Sie deshalb bitten, uns weitere Rechte einzuräumen, bzw. idealerweise Ihre Arbeit unter eine freie Lizenz zu stellen.

Die in unseren Augen praktikabelsten Lösungen sind vorselektiert.

Hiermit gestatte ich die Verwendung der **schriftlichen Ausarbeitung** zeitlich unbegrenzt und nicht-exklusiv unter folgenden Bedingungen:

Nur zur Bewertung dieser Arbeit

Nur innerhalb des Lehrstuhls im Rahmen von Forschung und Lehre

Unter einer Creative-Commons-Lizenz mit den folgenden Einschränkungen:

BY – Namensnennung des Autors

NC – Nichtkommerziell

SA – Share-Alike, d.h. alle Änderungen müssen unter die gleiche Lizenz gestellt werden.

(An Zitaten und Abbildungen aus fremden Quellen werden keine weiteren Rechte eingeräumt.)

Außerdem gestatte ich die Verwendung des im Rahmen dieser Arbeit erstellen **Quellcodes** unter folgender Lizenz:

Nur zur Bewertung dieser Arbeit

Nur innerhalb des Lehrstuhls im Rahmen von Forschung und Lehre

Unter der CC-0-Lizenz (= beliebige Nutzung)

Unter der MIT-Lizenz (= Namensnennung)

Unter der GPLv3-Lizenz (oder neuere Versionen)

(An explizit mit einer anderen Lizenz gekennzeichneten Bibliotheken und Daten werden keine weiteren Rechte eingeräumt.)

Ich willige ein, dass der Lehrstuhl für Medieninformatik diese Arbeit – falls sie besonders gut ausfällt - auf dem Publikationsserver der Universität Regensburg veröffentlichen lässt.

Ich übertrage deshalb der Universität Regensburg das Recht, die Arbeit elektronisch zu speichern und in Datennetzen öffentlich zugänglich zu machen. Ich übertrage der Universität Regensburg ferner das Recht zur Konvertierung zum Zwecke der Langzeitarchivierung unter Beachtung der Bewahrung des Inhalts (die Originalarchivierung bleibt erhalten).

Ich erkläre außerdem, dass von mir die urheber- und lizenzrechtliche Seite (Copyright) geklärt wurde und Rechte Dritter der Publikation nicht entgegenstehen.

Ja, für die komplette Arbeit inklusive Anhang

Ja, für eine um vertrauliche Informationen gekürzte Variante (auf dem Datenträger beigefügt)

Nein

Sperrvermerk bis (Datum):

Regensburg, 10.03.2022

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Unterschrift |

Inhalt des beigefügten Datenträgers

Beispiel (Ordner + Beschreibung):

|  |  |  |
| --- | --- | --- |
| /1\_Ausarbeitung | Die schriftliche Ausarbeitung als PDF und DOC | |
| /2\_Code | Quellcode und kompilierte Anwendung des Prototypen | |
| /3\_Studie/Design | Fragebogen und Script für die Benutzerstudie | |
| /3\_Studie/Rohdaten | Rohdaten der Studie im CSV-Format, inkl. Beschreibung der Felder | |
| /4\_Quellen | Alle in der Arbeit zitierten Quellen im PDF-Format | |
| /5\_Bilder | Alle selbst erstellten und aus anderen Quellen übernommenen Bilder | |
| /6\_Vorträge | Folien von Antritts- und Abschlussvortrag im PDF-Format | |
| /7\_Sonstiges | Notizen aus Besprechungen, Gedanken, … | |
|  | |

[Datenträger (CD, SD-Karte, o.ä.) hier oder auf Umschlaginnenseite einkleben]