

Lecture Notes in Networks and Systems 1026


Wojciech Zamojski ·
Jacek Mazurkiewicz ·
Jarosław Sugier · Tomasz Walkowiak ·
Janusz Kacprzyk *Editors*

System Dependability - Theory and Applications

Proceedings of the Nineteenth
International Conference
on Dependability of Computer Systems
DepCoS-RELCOMEX. July 1–5, 2024,
Brunów, Poland

 Springer

Series Editor

Janusz Kacprzyk , *Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*

Advisory Editors

Fernando Gomide, *Department of Computer Engineering and Automation—DCA, School of Electrical and Computer Engineering—FEEC, University of Campinas—UNICAMP, São Paulo, Brazil*

Okyay Kaynak, *Department of Electrical and Electronic Engineering, Bogazici University, Istanbul, Türkiye*

Derong Liu, *Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA*

Institute of Automation, Chinese Academy of Sciences, Beijing, USA

Witold Pedrycz, *Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada*

Systems Research Institute, Polish Academy of Sciences, Warsaw, Canada

Marios M. Polycarpou, *Department of Electrical and Computer Engineering, KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia, Cyprus*

Imre J. Rudas, *Óbuda University, Budapest, Hungary*

Jun Wang, *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Wojciech Zamojski · Jacek Mazurkiewicz ·
Jarosław Sugier · Tomasz Walkowiak ·
Janusz Kacprzyk
Editors

System Dependability - Theory and Applications

Proceedings of the Nineteenth International
Conference on Dependability of Computer
Systems DepCoS-RELCOMEX.
July 1–5, 2024, Brunów, Poland

Editors

Wojciech Zamojski
Department of Computer Engineering
Wrocław University of Science
and Technology
Wrocław, Poland

Jacek Mazurkiewicz
Department of Computer Engineering
Wrocław University of Science
and Technology
Wrocław, Poland

Jarosław Sugier
Department of Computer Engineering
Wrocław University of Science
and Technology
Wrocław, Poland

Tomasz Walkowiak
Department of Computer Engineering
Wrocław University of Science
and Technology
Wrocław, Poland

Janusz Kacprzyk 
Polish Academy of Sciences
Systems Research Institute
Warsaw, Poland

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-3-031-61856-7

ISBN 978-3-031-61857-4 (eBook)

<https://doi.org/10.1007/978-3-031-61857-4>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

In this volume we would like to present proceedings of the 19 International Conference on Dependability of Computer Systems DepCoS-RELCOMEX which is scheduled to be held stationary in Brunów Palace, Poland, from 1 to 5 July 2024. It is the second time, after a three-year interruption caused by the COVID-19 pandemic, when the Conference will be organized in a regular manner in a beautiful Brunów Palace—our traditional venue—close to many palaces and castles located in the charming Lower Silesia’s Valley of Palaces and Gardens.

DepCoS-RELCOMEX scope has always been focused on diverse issues which are constantly arising in performability and dependability analysis of contemporary computer systems and networks. Dependability of computer processing means obtaining reliable (true and timely) results in the conditions of processing both quantitative and qualitative data, using precise and “fuzzy/imitating” models and algorithms. It should be emphasized that artificial intelligence algorithms and tools are increasingly used in modern information technology and computer engineering, and therefore we are expanding our view on the dependability of systems that are progressively using methods based on cognitive systems and deep learning tool. In our opinion, this approach (**dependability** as the credibility of systems, **AI** tools and computer applications) meets the challenges which the computer science presently faces, both in its theoretical studies and in engineering. Ever-growing number of research methods being continuously developed for such analyses apply the newest results of artificial intelligence (AI) and computational intelligence (CI). Topical diversity of papers in these proceedings illustrate broad variety of multi-disciplinary subjects which should be considered in contemporary dependability explorations.

The Conference is organized annually since 2006 by the Department of Computer Engineering at the Faculty of Information and Communication Technology, Wrocław University of Science and Technology, but its roots go back to the heritage of two much older events: RELCOMEX (1977–1989) and Microcomputer School (1985–1995) which were organized by the Institute of Engineering Cybernetics (predecessor of the Department) under the leadership of prof. Wojciech Zamojski, now also the DepCoS chairman. Since 2006 the proceedings were printed, chronologically, first by the IEEE Computer Society (till 2009), then by Wrocław University of Science and Technology Publishing House (2010–2012), and recently by Springer Nature in the “Advances in Intelligent Systems and Computing” volumes no. 97 (2011), 170 (2012), 224 (2013), 286 (2014), 365 (2015), 479 (2016), 582 (2017), 761 (2018), 987 (2019), 1173 (2020), and 1389 (2021). Since 2022 DepCoS proceedings are a part of the “Lecture Notes in Networks and Systems” series, and the two previous editions were included in vols. 484 and 737. Springer Nature is one of the largest and most prestigious scientific publishers, with the LNNS titles being submitted for indexing in CORE Computing Research & Education database, Web of Science, SCOPUS, INSPEC, DBLP, and other indexing services.

It is our pleasant and honorable obligation now to thank everyone who participated in organization of the Conference and in preparation of this volume: the authors, members of the Program and the Organizing Committees, and all other individuals who assisted in creation of this book. Special recognition should be given to the reviewers whose opinions and comments helped immensely to select and enhance the submissions. Alphabetically, our sincerest thanks this year go to: Ilona Bluemke, Frank Coolen, Łukasz Jeleń, Alexander Grakovski, Alexey Lastovetsky, Urszula Kuźelewska, Jacek Mazurkiewicz, Jan Magott, Marek Młyńczak, Yiannis Papadopoulos, Czesław Smutnicki, Janusz Sosnowski, Robert Sobolewski, Jarosław Sugier, Kamil Szyc, Tomasz Walkowiak, Marek Woda, and Wojciech Zamojski. Their work, not mentioned anywhere else in this volume, deserves to be highlighted in this introduction.

At the end of this preface, we would like to thank all authors who selected DepCoS-RELCOMEX as *the* platform to publish and discuss their research results. We believe that included papers will contribute to advances in design, analysis, and engineering of dependable computer systems and networks, offering an interesting source material for scientists, researchers, engineers, and students working in these areas.

Wojciech Zamojski
Jacek Mazurkiewicz
Jarosław Sugier
Tomasz Walkowiak
Janusz Kacprzyk

Organization

**Nineteenth International Conference on Dependability
of Computer Systems DepCoS-RELCOMEX
Brunów, Poland, July 1–5, 2024**

Program Committee

Wojciech Zamojski (Chairman)	Wrocław University of Science and Technology, Poland
Michael Affenzeller	Upper Austria University of Applied Sciences, Austria
Ali Al-Dahoud	Al-Zaytoonah University, Amman, Jordan
Andrzej Białas	Research Network ŁUKASIEWICZ - Institute of Innovative Technologies EMAG, Katowice, Poland
Ilona Bluemke	Warsaw University of Technology, Poland
Magdalena Bogalecka	Gdynia Maritime University, Poland
Wojciech Bożejko	Wrocław University of Science and Technology, Poland
Eugene Brezhniev	National Aerospace University “KhAI”, Kharkov, Ukraine
Dariusz Caban	Wrocław University of Science and Technology, Poland
De-Jiu Chen	KTH Royal Institute of Technology, Stockholm, Sweden
Frank Coolen	Durham University, UK
Denny B. Czejdo	Fayetteville State University, USA
Wiktor B. Daszczuk	Warsaw University of Science and Technology, Poland
Mieczysław Drabowski	Cracow University of Technology, Poland
Francesco Flammini	University of Linnaeus, Sweden
Peter Galambos	Óbuda University, Hungary
Manuel Gill Perez	University of Murcia, Spain
Aleksander Grakowskis	Transport and Telecommunication Institute, Riga, Latvia
Laszlo Gulacsi	Óbuda University, Hungary
Atsushi Ito	Chuo University, Tokyo, Japan
Dariusz Jagielski	4th Military Hospital, Wrocław, Poland

Łukasz Jeleń	Wrocław University of Science and Technology, Poland
Ireneusz Józwiak	Wrocław University of Science and Technology, Poland
Igor Kabashkin	Transport and Telecommunication Institute, Riga, Latvia
Janusz Kacprzyk	Polish Academy of Sciences, Warsaw, Poland
Vyacheslav S. Kharchenko	National Aerospace University “KhAI”, Kharkov, Ukraine
Ryszard Klempous	Wrocław University of Science and Technology, Poland
Krzysztof Kołowrocki	Gdynia Maritime University, Poland
Leszek Kotulski	AGH University of Science and Technology, Krakow, Poland
Vasilis P. Koutras	University of the Aegean, Chios, Greece
Levente Kovacs	Óbuda University, Hungary
Henryk Krawczyk	Gdansk University of Technology, Poland
Dariusz Król	Wrocław University of Science and Technology, Poland
Andrzej Kucharski	Wrocław University of Science and Technology, Poland
Marek Kulbacki	Polish – Japanese Academy of Information Technology, Warsaw, Poland
Urszula Kuzelewska	Białystok University of Technology, Białystok, Poland
Alexey Lastovetsky	University College Dublin, Ireland
Henryk Maciejewski	Wrocław University of Science and Technology, Poland
Jan Magott	Wrocław University of Science and Technology, Poland
Jacek Mazurkiewicz	Wrocław University of Science and Technology, Poland
Daniel Medyński	Collegium Witelona, Legnica, Poland
Marek Młyńczak	Wrocław University of Science and Technology, Poland
Yiannis Papadopoulos	Hull University, UK
Andrzej Pawłowski	University of Brescia, Italy
Ewaryst Rafajłowicz	Wrocław University of Science and Technology, Poland
Przemysław Rodwald	Polish Naval Academy, Gdynia, Poland
Jerzy Rozenblit	Arizona University, Tucson, USA
Imre Rudas	Óbuda University, Hungary
Mirosław Siegiejczyk	Warsaw University of Technology, Poland

Czesław Smutnicki	Wrocław University of Science and Technology, Poland
Robert Sobolewski	Białystok University of Technology, Poland
Janusz Sosnowski	Warsaw University of Technology, Poland
Carmen Paz Suarez-Araujo	Universidad de Las Palmas de Gran Canaria, Spain
Jarosław Sugier	Wrocław University of Science and Technology, Poland
Laszlo Szilagyí	Sapientia Hungarian University of Transylvania, Romania
Tomasz Walkowiak	Wrocław University of Science and Technology, Poland
Max Walter	Siemens, Germany
Tadeusz Więckowski	Wrocław University of Science and Technology, Poland
Bernd E. Wolfinger	University of Hamburg, Germany
Min Xie	City University of Hong Kong, Hong Kong SAR, China
Irina Yatskiv	Transport and Telecommunication Institute, Riga, Latvia

Organizing Committee

Chair

Wojciech Zamojski	Wrocław University of Science and Technology, Poland
-------------------	---

Members

Jacek Mazurkiewicz	Wrocław University of Science and Technology, Poland
Jarosław Sugier	Wrocław University of Science and Technology, Poland
Tomasz Walkowiak	Wrocław University of Science and Technology, Poland
Tomasz Zamojski	Wrocław University of Science and Technology, Poland
Mirosława Nurek	Wrocław University of Science and Technology, Poland

Contents

Large Language Models for Data Extraction in Slot-Filling Tasks	1
<i>Marek Bazan, Tomasz Gniazdowski, Dawid Wolkiewicz, Juliusz Sarna, and Maciej E. Marchwiany</i>	
Anonymization of Bids in Blockchain Auctions Using Zero-Knowledge Proof	19
<i>Marlena Broniszewska, Wiktor B. Daszczuk, and Denny B. Czejdo</i>	
Survival Signature for Reliability Quantification of Large Systems and Networks	29
<i>Frank P. A. Coolen and Tahani Coolen-Maturi</i>	
Using Resizing Layer in U-Net to Improve Memory Efficiency	38
<i>Lehel Dénes-Fazakas, Szabolcs Csaholczi, György Eigner, Levente Kovács, and László Szilágyi</i>	
Multiprocessor Task Scheduling with Probabilistic Task Duration	49
<i>Dariusz Dorota</i>	
On-Line Scheduling Multiprocessor Tasks in the Non-predictive Environment	59
<i>Dariusz Dorota and Czeslaw Smutnicki</i>	
Solving a Vehicle Routing Problem for a Real-Life Parcel Locker-Based Delivery	69
<i>Radosław Idzikowski, Jarosław Rudy, and Michał Jaroszczuk</i>	
Hammering Test for Tile Wall Using AI	80
<i>Atsushi Ito, Yuma Ito, Masafumi Koike, and Katsuhiko Hibino</i>	
Models of Resilient Systems with Online Verification Considering Changing Requirements and Latent Failures	90
<i>Vyacheslav Kharchenko, Yuriy Ponochovnyi, Sergiy Dotsenko, Oleg Illiashenko, and Oleksandr Ivasiuk</i>	
Performance Optimizations of Real World Map Transformations for 3D Realtime Mobile Games	100
<i>Maciej Kopczynski</i>	

Preliminary Study on the Detection of Subtle Variations in Image Sequences for Identifying False Starts in Speedway Racing	111
<i>Jacek Krakowian and Łukasz Jeleń</i>	
Artificial Intelligence in Renewable Energy: Bibliometric Review of Current Trends and Collaborations	121
<i>Paweł Kut, Katarzyna Pietrucha-Urbanik, Martina Zelenakova, and Hany F. Abd-Elhamid</i>	
Impact of Learning Data Statistics on the Performance of a Recommendation System Based on MovieLens Data	132
<i>Urszula Kuźelewska and Michał Falkowski</i>	
Randomly Initiated Cyclostationary Excitations for Dimensionality Reduction in Wiener System Identification	143
<i>Gabriel Maik and Grzegorz Mzyk</i>	
Wireless Employee Safety Monitoring System with Measurement of Biomedical Parameters	152
<i>Marcel Maj</i>	
Artificial Intelligence Methods for Pet Emotions Recognition	163
<i>Jacek Mazurkiewicz</i>	
Parallel Swarm Intelligence: Efficiency Study with Fast Range Search in Euclidean Space	177
<i>Łukasz Michalski, Andrzej Sołtysik, and Marek Woda</i>	
Digital Transformation Impacts on Industry 4.0 Evolution	187
<i>Issam A. R. Moghrabi</i>	
Optimization of Procurement Strategy Supported by Simulated Annealing and Genetic Algorithm	196
<i>Szymon Niewiadomski and Grzegorz Mzyk</i>	
Tumor Volume Measurements in Animal Experiments: Current Approaches and Their Limitations	206
<i>Melánia Puskás, Borbála Gergics, Levente Kovács, and Dániel András Drexler</i>	
Utilizing CNN Architectures for Non-invasive Diagnosis of Speech Disorders	218
<i>Filip Ratajczak, Mikołaj Najda, and Kamil Szyc</i>	

Preparing a Dataset of Ransomware BTC Addresses for Machine Learning Purpose 227
Przemysław Rodwald

Human Technology Frontier: A Retrospective and Challenges for the Future in the Era of Artificial Intelligence 237
Jerzy W. Rozenblit

Styles for Describing Reliable Finite State Machines in Verilog HDL 241
Valery Salauyou

Smartphone-Based Biometric System Involving Multiple Data Acquisition Sessions 252
A. Sawicki and K. Saeed

An Ontology for the Fashion Domain Based on Knowledge Retrieval 261
Karolina Selwon and Julian Szymański

Comparative Efficiency Study of Protective Relays Schemes in Wind Energy Conversion Systems 272
Robert Adam Sobolewski

Dedicated FPGA Resources in Improving Power Efficiency of Implementations of BLAKE3 Hash Function 283
Jarostaw Sugier

Assessing Inference Time in Large Language Models 296
Bartosz Walkowiak and Tomasz Walkowiak

Robustness of Named Entity Recognition Models 306
Paweł Walkowiak

The Impact of Modes of Locomotion in a Virtual Reality Environment on the Human Being 316
Marek Woda and Jakub Michalski

Data Augmentation Techniques to Detect Cervical Cancer Using Deep Learning: A Systematic Review 325
Betlehem Zewdu Wubineh, Andrzej Rusiecki, and Krzysztof Halawa

Orthogonal Transforms in Neural Networks Amount to Effective Regularization 337
Krzysztof Zajac, Wojciech Sopot, and Paweł Wachel

Classification of European Residential Tall Buildings – Application
Assumptions 349
Tomasz Zamojski

Author Index 361



Large Language Models for Data Extraction in Slot-Filling Tasks

Marek Bazan^{1,2} , Tomasz Gniazdowski² , Dawid Wolkiewicz² ,
Juliusz Sarna² , and Maciej E. Marchwiany² 

¹ Department of Computer Engineering, Wrocław University of Science and Technology, ul. Janiszewskiego 11/17, 50-370 Wrocław, Poland
marek.bazan@pwr.edu.pl

² JT Weston sp. z o.o, Atrium Plaza, al. Jana Pawła II 29, 00-867 Warszawa, Poland

Abstract. Large language models (LLMs) have turned out recently to be a powerful tool for solving natural language processing and understanding tasks. In this paper, we investigate the usage of three open-source large language models in a slot-filling task, which is a crucial task in chatbot development. Apart from testing the method on an in-house created dataset, we checked the methodology on two main benchmarks in this field. The obtained results for models with 7B parameters are comparable with those achieved by closed-source chatGPT family models, which are more than 20 times bigger.

Keywords: Open source LLMs · Slot Filling · LLaMA · Orca · Mistral

1 Introduction

Large language models (LLMs) are commonly used for data extraction and understanding tasks such as: named entity recognition [38], slot-filling tasks [7, 28, 34], event extraction [9], relation extraction [18, 37] as well as information extraction, in general [8, 11, 31].

A general, effective solution of a slot-filling problem is the main component of modern chatbots. A slot-filling task is encountered e.g. while using chatbots to guide a customer in filling forms for e-commerce web pages. It can also be found in analyses of long detailed case studies that are not driven by a single question chatbot.

The overall pipeline of using an LLM in the above scenario requires:

1. an automatic prompt and context generation,
2. sending the prompt and context to the LLM,
3. parsing the LLM's response and sending back results.

In this paper, we study the performance of LLMs: Orca [22, 25], LLaMA [36] and Mistral [13]. Our investigations cover an in-house built dataset (that we

publish) in Polish and two benchmarks for a slot-filling task in English named SNIPS [5] and ATIS [19]. To our knowledge, the presented results are the first attempt at the evaluation of Open Source LLMs on a slot-filling task that can be compared with closed models of the ChatGPT family [27].

The few-shot learning technique for LLMs was discovered at the beginning of 2020s and presented by Braun et al. [4]. Based on this research we focus on three types of prompt creation methods: zero-shot learning, few-shot learning, and chain of thought. Moreover, following [15] we investigated much bigger models.

Benchmarks that we used are in English. However, the application of our priority is message understanding in Polish. The in-house datasets coming from the business processes of *JT Weston sp. z o.o.* are in Polish. Since LLMs exploited in this paper have been trained on Polish language corpuses, our method can be used for both languages.

The remainder of the paper is organized in the following way. First, we review related research literature in the area of slot-filling tasks in the deep learning domain as well as LLMs. We then describe usage of three prompting methods. In Sect. 4 we present the results on the most popular publicly available benchmarks as well as on our in-house datasets. Results on benchmarks are compared with ChatGPT family models. Finally, we comment on the current limitations of open source LLMs and conclude the paper.

2 Related Work

A comprehensive survey on applications of LLMs can be found in [10]. A wide validation of closed LLMs (the ChatGPT family) on many natural language processing tasks can be found in [14].

Surveys on slot-filling tasks with the intent classification and the named entity recognition tasks can be found in [40] and [12], respectively.

A slot-filling task is also encountered in chatbots for business processes in the corporate world [1, 33]. The Neula Assistant is one of the examples of such a platform where business processes are tracked by executing BPMN diagrams.

The most commonly used benchmarks for slot-filling tasks are SNIPS, which contains 7 intents and 39 slot types, and ATIS, which contains 21 intents and 128 slot types. The splits for train, validation and test subsets of the above dataset may be found in [40].

The first attempt at a slot-filling task was based on Maximum Entropy models and Hidden Markov models [23]. Conditional random fields [30] had been common approaches used to solve sequence tagging problems. The history of the application of deep neural networks (DNN) for this task dates back to deep convex networks [6] and with kernel learning based on n -grams. The application of the LSTMs for this purpose was first shown in [41]. The first application of BERT to solve this task, shown in [17], included a comparison with other models such as BiGRU and a convolutional neural network. Apart from deep neural architectures, conditional random fields also achieve good results and have been investigated in parallel [42] to DNNs. The current state-of-the-art results for

the ATIS and SNIPS datasets are achieved with BERT architecture enhanced with convolutional layers [24]. Recently, LLMs were used for information extraction in a few-shot scenarios in [20]. The results a slot-filling task achieved for closed-source ChatGPT models are presented in [28].

Another approach to retrieve general information from text using LLMs is in-context learning [3]. It differs from zero-shot, few-shot and chain of thought. However, it also uses samples for feeding prompt fields.

3 Methods

In this section, we describe three prompting methods to command LLMs that we have investigated as well as the approach to process the response from models.

3.1 Prompting

Zero-Shot. Zero-shot prompting is a method in which LLM receives a natural language description of a given task [4]. An example of a zero-shot prompt in English and Polish can be found in Appendices A.1 and B.1, respectively.

Few-Shot. Few-shot prompting is a method that incorporates example pairs of context and completion, followed by the actual user question [4]. In our solution, five examples (general tasks of slot-filling) were provided each time during testing. Examples were not related to the actual task of the following user question. An example of few-shot prompts in English and Polish can be found in Appendices A.2 and B.2, respectively.

Zero-Shot Chain of Thought. The chain of thought (CoT) prompting (few-shot-CoT prompting) method [39] was proposed as a simple improvement over the original Few-shot prompting method. The main goal was to simplify an example tasks by showing a model of step-by-step solutions. The performance boost was impressive, especially with very sizable LLMs [27, 36]. The main disadvantage of this method was the need to prepare task-specific examples with explanations for the model. The zero-shot-CoT prompting method [16] solves this problem with a simple solution, by adding a “*Let’s think step by step*” phrase at the end of the zero-shot prompt. An example of zero-shot-CoT prompts in English and Polish can be found in Appendices A.3 and B.3, respectively.

3.2 Receiving the Output

Key Matching Algorithm. During the experiments, it was noticed that the LLMs tend to change the name of slots (entities) to be filled. Most often, models do this by using synonyms or by changing the language of the slot name (most often from Polish to English). To solve this problem we implemented a greedy-matching algorithm. First, the slot names were automatically translated to the

proper language (English or Polish). Then, sentence-level embeddings of slot names from the original prompt and the LLM were calculated with a Fasttext algorithm [2]. The original and the LLM’s names were matched by their maximum cosine similarity value (see Algorithm 1). One iteration of the algorithm is presented. After one iteration, the matching threshold value was decreased. The algorithm terminated when all slot names were matched.

Algorithm 1. One iteration of the greedy slot names matching algorithm

```

1: for  $i \leftarrow 1$  to  $N$  do ▷  $N$  is the number of slot names
2:    $slotNameLLMVec \leftarrow slotNamesLLMVec[i]$ 
3:   for  $j \leftarrow 1$  to  $length(slotNamesOrgVec)$  do
4:      $slotNameOrgVec \leftarrow slotNamesOrgVec[j]$ 
5:      $cosSimilarities[j] \leftarrow cosSimilarity(slotNameLLMVec, slotNameOrgVec)$ 
6:   end for
7:   if  $max(cosSimilarities) \geq 0.9 - \eta$  then
8:      $matchedSlotNames[i] \leftarrow slotNamesOrgVec[argMax(cosSimilarities)]$ 
9:      $slotNamesOrgVec.pop(argMax(cosSimilarities))$ 
10:  end if
11: end for ▷  $\eta$  lowers the threshold

```

4 Experiments

In this section, we describe the datasets on which we tested the LLMs, the evaluation process with four different metrics as well as discuss some limitations of one of these metrics. Finally, we share some comparison observations about the behavior of the models we tested on a lot-filling task.

4.1 Datasets

The statistical experiments we carried out were on test partitions of SNIPS [5] and ATIS [32] datasets.

Moreover, we created two slot-filling Polish datasets within *JT Weston*: (i) *leaves* based on a process of taking days off and (ii) *delegations* based on a process of accounting for a business trip. The main assumption of datasets prepared with the business team was to make the data reflect real business situations and problems. Datasets were prepared independently by people from *JT Weston* and verified by the business department. Many field names have to be inferred based on other values and various date formats were used. Statistics of all datasets used in experiments are presented in Appendix C.

4.2 Evaluation

LLMs’ hallucinations are more challenging in Polish than in English. Polish is sensitive to not only word order but also on extremely complicated inflection (which can change the meaning). Moreover, any given entity may have multiple values in the input text. This is why there was no one-to-one correspondence between the predicted and ground truth entities (as in the *IOB* format). It was not possible to determine the number of all false negative predictions (and *F1* metric) in in-house datasets. For this reason, only precision was calculated for in-house datasets. Moreover, from the business perspective precision is the most important metric. For the SNIPS and ATIS datasets it was possible to calculate the *F1* score, so for these datasets, we calculated a slot *F1* metric [35].

Algorithm 2. Precision source metric

```

1: for  $i \leftarrow 1$  to  $N$  do                                     ▷  $N$  is the number of entity predictions
2:    $entityPredClass \leftarrow entitiesPreds[i].class$ 
3:    $entityPred \leftarrow entitiesPreds[i].value$ 
4:   if  $entityPred \neq "n/a"$  then
5:      $notNAPredCounter \leftarrow notNAPredCounter + 1$ 
6:      $entityPredTokenLemma \leftarrow lemmatizeTokens(tokenizeText(entityPred))$ 
7:      $inputTextTokenLemma \leftarrow lemmatizeTokens(tokenizeText(inputText))$ 
8:     for  $j \leftarrow 1$  to  $k$  do                                 ▷  $k$  is a parameter of a evaluation algorithm
9:        $entityPredNGrams \leftarrow nGrams(entityPredTokenLemma, j)$ 
10:       $inputTextNGrams \leftarrow nGrams(inputTextTokenLemma, j)$ 
11:       $commonPartPredText \leftarrow commonPart(entityPredNGrams, inputTextNGrams)$ 
12:      if  $commonPartPredText \neq \emptyset$  then
13:        if  $j \geq 2 \vee commonPartPredText \notin stopWordsSet$  then
14:           $notHallucinations \leftarrow notHallucinations + 1$ 
15:        end if
16:      end if
17:    end for
18:  end if
19: end for
20:  $precisionSource \leftarrow notHallucinations/notNAPredCounter$  ▷ precision source metric value
    is calculated for all entities predictions for all input texts

```

To calculate the *precision source* metric [26], we began with checking the entity value hallucination. Several steps were necessary. First the input text and predicted entity value were tokenized and lemmatized (lines 6–7 in Algorithm 2). The lemmatization step is our development of the original method, because LLMs often change the form of words, especially in Polish. Then all the words were changed to lowercase. Finally, a common part of any n -grams from the prediction of the entity value and the input text was calculated (line 12 in Algorithm 2) (for $n = 1$, we additionally check if the common part was not a stop word). If a common part exists and it was not a stop word, then the prediction was not recognized as hallucination. The precision source is considered as a *percentage of named entities in the summary that can be found in the source* [26]. In our case, it was a percentage of entity predictions that were present in the input text (line 20 in Algorithm 2).

We calculated precision for the *leaves* and *delegations JT Weston* datasets. Due to the frequent changes in word forms in Polish, we considered the predictions to be correct when the cosine similarity between sentence-level embedding vectors of ground truth and entity value prediction calculated with the *Fasttext* (c.f. [2]) algorithm was above a given threshold (0.8 and 0.9). We mark the metrics as $precision_{0.8}$ and $precision_{0.9}$. The results for $precision_{0.9}$ are presented in Fig. 1. All numerical values can be found in Appendix D.

Similarly, for the SNIPS and ATIS datasets, the $F1$ metric with thresholds 0.8 and 0.9 was used, which enabled us to compute $F1_{0.8}$ and $F1_{0.9}$. The slot $F1$ score with threshold is calculated for a slot with an assumption that the answer is correct if cosine similarity between the sentence-level embedding vectors of the predicted value and ground truth is above a given threshold. For the slot $F1$ score, a perfect match between the predicted value and ground truth was necessary to recognize the prediction as a true positive. Numerical values are presented in Table 1. For the *Orca mini 3* we used the few-shot method with examples in the system prompt. For the *Mistral 02 instruct* the few-shot prompting method was used, with examples in the user prompt. For the *LLaMA 2 chat* we used the zero-shot chain of thought method.

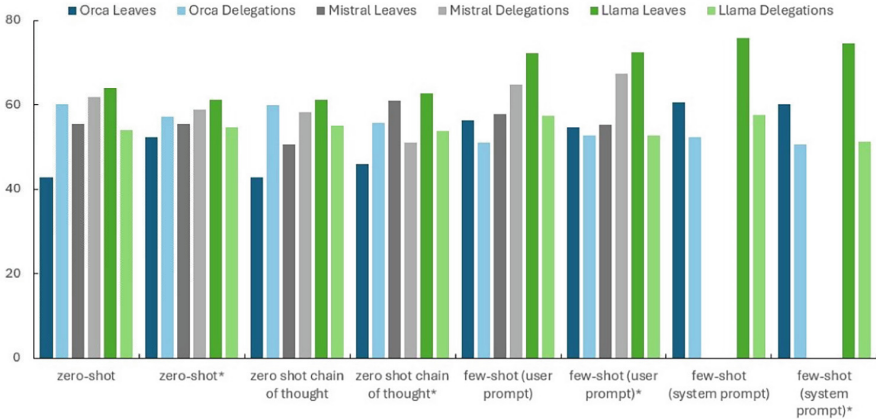


Fig. 1. Comparison of $F1_{0.9}$ for different models and prompting techniques. Prompting methods marked with a star symbol (*) are the prompts, in which an additional dataset name was passed as the sentence - “Text is about leave” or “Text is about delegation” in order to to emphasize the context of the message.

Table 1. Slot *F1* metric calculated for *Orca mini 3*, *Mistral 02 instruct* and *LLaMA 2 chat* for the SNIPS and ATIS datasets. Results of our experiments were compared with the results of *Codex*, *Chat GPT*, *GPT-3.5* [28] and State-of-The-Art results (SOTA).

model	SNIPS		ATIS	
	slot <i>F1</i>	[%] of SOTA	slot <i>F1</i>	[%] of SOTA
<i>Orca mini 3</i> (our experiments)	56.48	57.46	8.85	8.99
<i>Mistral 02 instruct</i> (our experiments)	55.56	56.52	51.93	52.74
<i>LLaMA 2 chat</i> (our experiments)	35.13	35.74	1.71	1.74
<i>Codex</i> [21]	68.90	70.09	57.29	58.19
<i>GPT-3.5</i>	68.90	70.09	55.72	56.59
<i>Chat GPT</i>	58.24	59.25	15.71	15.96
finetuned SOTA (<i>CTRAN</i>) [29]	98.30	–	98.46	–

4.3 Reasoning Problem and Hallucinations with the Precision Source Algorithm

We require models to make simple reasoning in the prompts. However we observed varied hallucinations. The precision source algorithm cannot generalize synonymous phrases, for example if there is “sick leave”, then the reason for the leave is “sickness”. However, this leads to the prediction being considered a hallucination by a precision source metric (if the word “sickness”, or its other form does not appear elsewhere in the input text). (The lemma of the word “sick” is “sick” and for “sickness” is “sickness” - these are different parts of speech). Models also changed the format of data or use synonyms (e.g., “n/a” may be changed into “no information”, or “1.12.2023” into “the first of December” by a model). This creates a much more difficult problem of hallucination detection. The detector has to differentiate incorrect values from synonyms or format changes. Moreover, we noticed that if the embedding vectors of these two phrases (e.g., “1.12.2024” and “tomorrow”) are close enough, the hallucinations will not be detected by commonly used algorithms (e.g., word embeddings). This seems as a very complex task and the problem is left open.

4.4 Output Generated by Models

During the research, several conclusions were drawn about the responses created by the *LLMs*.

The *Orca 3 mini* model rarely added hallucinations or additional, unnecessary information to the generated output. For both zero-shot and few-shot prompts, it usually returned only the extracted data. The format of the JSON returned by *Orca 3 mini* was also usually correct.

The *Mistral 02 instruct* changed the date format very often (e.g., May 5 to 5/5). This model often invents non-existent words in Polish. In most examples, it returned JSON in the valid format with some additional data. The *Mistral 02 instruct* sometimes hallucinated - it returned its own examples in a few-shot prompting method that were not present in the input prompt.

The *LLaMA 2 chat* changed the format of the returned data a lot and it converted JSON into formats that cannot be parsed. It added a lot of redundant text explaining its decisions (instead of returning just JSON with the extracted data). The most difficult task was to extract the returned data from the model response.

To generate the correct output (valid JSON format), the LLM had to process the same input data many times. All investigated models were prompted for each message (up to 5 times) until they returned the output that could have been parsed. The performance of models measured by requested number of unconnected output format is presented in Appendix D. We observed that prompting techniques can change performance of models, it is specialty visible for no-English texts. In general LLMs has better performance for English. For English prompts and Polish texts, LLMs most often automatically translated the extracted data and entity names into English. This made the approach of prompting models in English with Polish data less effective. We also tested single value extraction by prompt (with multiple prompts for single test) , what usually produced an output in an incorrect format and was much slower. Models that extracted a single entity value instead of multiple ones produced more hallucinations. Therefore this approach was also abandoned.

5 Conclusions

In this paper we have presented the results of the application of chosen small (up to 7B parameters), open-source LLMs to a slot-filling tasks for complex texts. This task is a key ingredient of natural language understanding of dialogs for chatbot development. We compare our methodology with two benchmarks with the chatGPT 3.5 family models. The achieved results show that open-source LLMs can be as good, and in some cases even better than closed-source models.

For model evaluation, four metrics were used: (i) precision based on cosine similarity with thresholds 0.8 and 0.9, (ii) *F1* based on cosine similarity with thresholds 0.8 and 0.9, (iii) slot-*F1* and (iv) precision source. The slot-*F1* was calculated for the SNIPS and ATIS benchmarks. *LLaMA 2* models such as *LLaMA chat* and *Orca* achieve very low performance on this metric due to many false positives, though the results of *Mistral* outperform even ChatGPT.

The best performance (measured by precision source) on in-house datasets was achieved with the *LLaMA 2 Chat*, for independent benchmarks, the *Mistral* model was the best, since it limits the output of the information that is not supported by the input message, that is visible on the ATIS benchmark. Based on our results, the best-performing prompt techniques have been chosen: Few-shot for the *Orca 3 mini* and *Mistral*, and the zero-shot chain of thought for the *LLaMA chat*. Detailed information can be found in Appendix D.

In our investigations, we experimented with different LLMs in the slot-filling task. Our problem was complicated. The texts were written in Polish and were written in a tricky way. Moreover, we used various date types, and many slot values had to be inferred by other values or from the input text. Obtaining the valid JSON format from the model also turned out to be a difficult task. During

the research, numerous indirect problems, such as changing the name of the keys in the returned JSON were solved. Due to the high level of complexity, various LLMs turned out to be too weak to be used in our problem. The *MPT*, *Falcon*, *Bloomz* or even base *LLaMA 2* models were rejected for this reason.

Acknowledgement. This work was partially financed from the grant POIR.01.01.01-00-0930/19 entitled “Development of self-configuring personal assistant module on the BPM platform using natural language and artificial intelligence processing tools and algorithms”.

A Examples of Prompts in English

A.1 Example of a Zero-Shot Prompt

You are an AI assistant who is very good at extracting data from short texts written in English. Extract the given values: ENITY_NAMES_TO_EXTRACT. Return the extracted data in valid json format. Return key names and values in English. If a given value is not in the text or you are not sure of the answer - return “n/a”. If a given value is not found in the text, try to infer it from other values. Return extracted values in double quotes:“”. Do not return other information. Help as much you can.

TEXT_FROM_WHICH_DATA_IS_EXTRACTED

A.2 Example of a Few-Shot Prompt

You are an AI assistant who is very good at extracting data from short texts written in English. Return only extracted data in valid json format. Examples are presented below.

Text: I noticed a fault in the elevator on the fourth floor and in the elevator on the fifth floor (the doors get stuck when opening). Best regards, Anna Nowak (555-123-456)

Entities: [“Name”, “Surname”, “Contact telephone number”, “Description of the fault”, “Location of the fault”, “Date of the fault noticed”]

Output:

```
{
  "Name": ["Anna"],
  "Surname": ["Nowak"],
  "Contact telephone number": ["555-123-456"],
  "Description of the fault": ["door stuck when opening"],
  "Location of the fault": ["elevators on the fourth floor", "elevators on the fifth floor"],
  "Date of the fault noticed": "n/a"
}
```

Text: As part of the new “Business Process Automation” project, we are determining the details. The project will last for 12 months. The total budget of the project is PLN 1,000,000. We will implement the project in Warsaw. Regards

Mariusz.

Entities: ["Project name", "Project budget", "Project team name", "Project duration", "City of project implementation", "Country of project implementation"]

Output:

```
{
  "Project name": ["Business Process Automation"],
  "Project budget": ["PLN 1,000,000"],
  "Project team name": "n/a",
  "Project duration": ["12 months"],
  "City of project implementation": ["Warsaw"],
  "Country of project implementation": ["Poland"]
}
```

Text: I am an employee of ABC Electronics, my order number is 20231001. My shopping cart is deleting (emptying) itself. The second error is that it logs me out of the site at random moments. Please help.

Entities: ["Customer name", "Order number", "Department", "Importance", "Report date", "Error description", "Country", "Supervisor"]

Output:

```
{
  "Customer name": ["ABC Electronics"],
  "Order number": ["20231001"],
  "Department": "n/a",
  "Importance": "n/a",
  "Report date": "n/a",
  "Error description": ["deleting the shopping cart", "logging out of the website at random moments"],
  "Country": "n/a",
  "Supervisor": "n/a"
}
```

Text: I am reporting a problem with the printer. Marek

Entities: ["Device name", "Name"]

Output:

```
{
  "Device name": ["printer"],
  "Name": ["Marek"]
}
```

Text: This is my candidacy for the position of data analyst.

Entities: ["Position"]

Output:

```
{
  "Position": ["data analyst"]
}
```

Text: TEXT_FROM_WHICH_DATA_IS_EXTRACTED

Entities: ENTITY_NAMES_TO_EXTRACT

Output:

A.3 Example of a Zero-Shot-CoT Prompt

You are an AI assistant who is very good at extracting data from short texts written in English. Extract the given values from the text: ENTITY_NAMES_TO_EXTRACT. Return the extracted data in valid json format. Return key names and values in English. If a given value is not in the text or you are not sure of the answer - return "n/a". If a given value is not found in the text, try to infer it from other values. Return extracted values in double quotes: "". Do not return other information. Let's think step by step.
TEXT_FROM_WHICH_DATA_IS_EXTRACTED

B Examples of Prompts in Polish

B.1 Example of a Zero-Shot Prompt

Jesteś asystentem AI, który bardzo dobrze wydobywa dane z krótkich tekstów napisanych po polsku. Wyciągnij z tekstu podane wartości: ENTITY_NAMES_TO_EXTRACT. Zwróć wydobyte dane w prawidłowym formacie json. Nazwy kluczy oraz wartości zwróć w języku polskim. Jeśli w tekście nie ma danej wartości lub nie jesteś pewny odpowiedzi - zwróć "n/a". Jeśli w tekście nie ma danej wartości, spróbuj ją wywnioskować na podstawie innych wartości. Wyciągnięte wartości zwróć w apostrofach: ". Nie zwracaj innych informacji. Pomóż jak najlepiej potrafisz.
TEXT_FROM_WHICH_DATA_IS_EXTRACTED

B.2 Example of a Few-Shot Prompt

Jesteś asystentem AI, który bardzo dobrze wydobywa dane z krótkich tekstów napisanych po polsku. Zwróć jedynie wyciągnięte dane w prawidłowym formacie json. Przykłady zaprezentowano poniżej.

Text: Zauważyłem usterkę w windzie na czwartym piętrze oraz w windzie na piątym piętrze (drzwi się zacinają przy otwieraniu). Pozdrawiam, Anna Nowak (555-123-456)

Entities: ["Imię", "Nazwisko", "Telefon kontaktowy", "Opis usterki", "Miejsce usterki", "Data zauważenia usterki"]

Output:

```
{
  "Imię": ["Anna"],
  "Nazwisko": ["Nowak"],
  "Telefon kontaktowy": ["555-123-456"],
  "Opis usterki": ["zacinanie się drzwi przy otwieraniu"],
  "Miejsce usterki": ["windy na czwartym piętrze", "windy na piątym piętrze"],
  "Data zauważenia usterki": ["n/a"]
}
```

Text: W ramach nowego projektu "Automatyzacja Procesów Biznesowych" ustalamy szczegóły. Projekt będzie trwać przez 12 miesięcy. Całkowity budżet projektu

to 1,000,000 PLN. Projekt będziemy realizować w Warszawie. Pozdrawiam, Mariusz.

Entities: ["Nazwa projektu", "Budżet projektu", "Nazwa zespołu projektu", "Czas trwania projektu", "Miasto realizacji projektu", "Państwo realizacji projektu"]

Output:

```
{
  "Nazwa projektu": ["Automatyzacja Procesów Biznesowych"],
  "Budżet projektu": ["1,000,000 PLN"],
  "Nazwa zespołu projektu": "n/a",
  "Czas trwania projektu": ["12 miesięcy"],
  "Miasto realizacji projektu": ["Warszawa"],
  "Państwo realizacji projektu": ["Polska"]
}
```

Text: Jestem pracownikiem ABC Electronics, mój numer zamówienia to 20231001. Mój koszyk zakupowy sam się usuwa (opróżnia). Drugi błąd polega na wylogowywaniu mnie ze strony w losowych momentach. Proszę o pomoc.

Entities: ["Nazwa klienta", "Numer zamówienia", "Oddział", "Waga błędu", "Data zgłoszenia", "Opis błędów", "Kraj", "Przełożony"]

Output:

```
{
  "Nazwa klienta": ["ABC Electronics"],
  "Numer zamówienia": ["20231001"],
  "Oddział": "n/a",
  "Waga błędu": "n/a",
  "Data zgłoszenia": "n/a",
  "Opis błędów": ["usuwanie koszyka zakupowego", "wylogowywanie ze strony w losowych momentach"],
  "Kraj": "n/a",
  "Przełożony": "n/a"
}
```

Text: Zgłaszam problem z drukarką. Marek

Entities: ["Nazwa urządzenia", "Imię"]

Output:

```
{
  "Nazwa urządzenia": ["drukarka"],
  "Imię": ["Marek"]
}
```

Text: To moja kandydatura na stanowisko analityka danych.

Entities: ["Stanowisko"]

Output:

```
{
  "Stanowisko": ["analityk danych"]
}
```

Text: TEXT_FROM_WHICH_DATA_IS_EXTRACTED

Entities: ENTITY_NAMES_TO_EXTRACT

Output:

B.3 Example of a Zero-Shot-CoT Prompt

Jesteś asystentem AI, który bardzo dobrze wydobywa dane z krótkich tekstów napisanych po polsku. Wyciągnij z tekstu podane wartości: ENTITY_NAMES_TO_EXTRACT. Zwróć wydobyte dane w prawidłowym formacie json. Nazwy kluczy oraz wartości zwróć w języku polskim. Jeśli w tekście nie ma danej wartości lub nie jesteś pewny odpowiedzi - zwróć "n/a". Jeśli w tekście nie ma danej wartości, spróbuj ją wywnioskować na podstawie innych wartości. Wyciągnięte wartości zwróć w apostrofach: "'". Nie zwracaj innych informacji. Myśl krok po kroku.

TEXT_FROM_WHICH_DATA_IS_EXTRACTED

C Dataset Details

The details of the datasets are presented in Table 2.

Table 2. Statistics of the datasets that were used in our experiments. Texts count refers to the total number of texts, slots to fill refers to the total number of slots to fill, slots sets refer to the number of slots sets, unique slots refer to the number of unique slots in the given dataset. Some slots were present in multiple slot sets.

dataset name	text count	slots to fill	slots sets	unique slots
<i>JT Weston leaves</i>	44	169	1	5
<i>JT Weston delegations</i>	30	314	1	12
SNIPS [5]	700	1790	7	39
ATIS [32]	893	2837	20	69

D Results

(See Tables 3, 4, 5 and 6).

Table 3. Metric values for the *Orca 3 mini*, *Mistral 02 instruct* and *LLaMA 2 chat* models received for the *JT Weston leaves* dataset with metrics: $precision_{0.8}$, $precision_{0.9}$ and precision source (see Sect. 4.2). Prompting methods marked with a star symbol (*) had an additional dataset name passed as a sentence - *Text is about leave*.

prompting method	$precision_{0.8}$	$precision_{0.9}$	precision source
Orca 3 mini			
zero-shot	51.81	42.77	81.33
zero-shot*	59.75	52.2	82.39
zero-shot chain of thought	53.01	42.77	78.92
zero-shot chain of thought*	52.35	45.88	78.24
few-shot (user prompt)	66.25	56.25	88.13
few-shot (user prompt)*	66.06	54.55	89.7
few-shot (system prompt)	72.22	60.49	93.21
few-shot (system prompt)*	73.01	60.12	92.64
Mistral 02 instruct			
zero-shot	62.5	55.36	82.14
zero-shot*	63.16	55.56	81.29
zero-shot chain of thought	62.07	50.58	80.46
zero-shot chain of thought*	69.77	61.05	80.23
few-shot (user prompt)	69.13	57.72	81.21
few-shot (user prompt)*	67.11	55.26	84.21
few-shot (system prompt)	–	–	–
few-shot (system prompt)*	–	–	–
LLaMA 2 chat			
zero-shot	71.43	64.00	82.86
zero-shot*	68.89	61.11	85.00
zero-shot chain of thought	68.33	61.11	83.89
zero-shot chain of thought*	70.62	62.71	85.88
few-shot (user prompt)	77.37	72.26	89.78
few-shot (user prompt)*	80.44	72.46	90.58
few-shot (system prompt)	80.88	75.74	94.85
few-shot (system prompt)*	80.85	74.47	91.49

Table 4. Metric values for the *Orca 3 mini*, *Mistral 02 instruct* and *LLaMA 2 chat* models received for the *JT Weston delegations* dataset. Metrics: $precision_{0.8}$, $precision_{0.9}$ and precision source (see Sect. 4.2). Prompting methods marked with a star symbol (*) had an additional dataset name passed as a sentence - *Text is about delegation*.

prompting method	$precision_{0.8}$	$precision_{0.9}$	precision source
Orca 3 mini			
zero-shot	68.05	60.06	73.67
zero-shot*	63.91	57.10	74.56
zero-shot chain of thought	67.81	59.83	73.79
zero-shot chain of thought*	64.27	55.62	72.62
few-shot (user prompt)	56.40	50.99	71.68
few-shot (user prompt)*	57.71	52.74	70.15
few-shot (system prompt)	57.28	52.35	72.35
few-shot (system prompt)*	55.93	50.61	70.94
Mistral 02 instruct			
zero-shot	66.23	61.91	69.26
zero-shot*	63.74	58.79	73.08
zero-shot chain of thought	60.66	58.29	69.67
zero-shot chain of thought*	55.09	50.94	69.43
few-shot (user prompt)	66.67	64.87	78.98
few-shot (user prompt)*	67.98	67.37	77.04
few-shot (system prompt)	–	–	–
few-shot (system prompt)*	–	–	–
LLaMA 2 chat			
zero-shot	62.11	54.04	72.98
zero-shot*	61.88	54.69	74.38
zero-shot chain of thought	61.42	54.94	72.84
zero-shot chain of thought*	61.34	53.78	72.38
few-shot (user prompt)	61.48	57.41	79.26
few-shot (user prompt)*	56.36	52.73	65.00
few-shot (system prompt)	60.79	57.55	83.45
few-shot (system prompt)*	53.98	51.14	63.07

Table 5. Metric values for the *Orca mini 3*, *Mistral 02 instruct* and *LLaMA 2 chat* models received for the SNIPS and ATIS datasets. For the *Orca mini 3* we used the few-shot method with the system prompt. For *Mistral 02 instruct* the few-shot prompting method was used, with the user prompt. For *LLaMA 2 chat* we used the zero-shot-chain of thought method.

model	slot $F_{10.8}$	slot $F_{10.9}$	precision source
SNIPS			
<i>Orca mini 3</i>	67.52	64.48	67.68
<i>Mistral 02 instruct</i>	67.08	63.89	72.03
<i>LLaMA 2 chat</i>	56.82	53.26	61.77
ATIS			
<i>Orca mini 3</i>	23.82	22.93	25.69
<i>Mistral 02 instruct</i>	66.99	64.56	77.98
<i>LLaMA 2 chat</i>	10.83	9.96	20.03

Table 6. The number of times for a given model for the *leaves* and *delegations* datasets when the data returned by the model was not parsable. Prompting methods marked with a star symbol (*) had an additional dataset name passed as a sentence - *Text is about leave* (for *leaves* dataset) and *Text is about delegation* (for *delegations* dataset).

	leaves			delegations		
	<i>Orca 3</i>	<i>LLaMA 2</i>	<i>Mistral 02</i>	<i>Orca 3</i>	<i>LLaMA 2</i>	<i>Mistral 02</i>
	<i>mini</i>	<i>chat</i>	<i>instruct</i>	<i>mini</i>	<i>chat</i>	<i>instruct</i>
zero-shot	0	0	0	0	4	9
zero-shot*	0	0	0	1	1	12
zero-shot chain of thought	0	0	0	0	2	11
zero-shot chain of thought*	0	0	0	0	0	7
few-shot (user prompt)	0	0	0	0	3	0
few-shot (user prompt)*	0	2	0	0	5	0
few-shot (system prompt)	0	0	–	0	4	–
few-shot (system prompt)*	1	0	–	0	8	–

References

1. Barros, A., Sindhgatta, R., Nili, A.: Scaling up chatbots for corporate service delivery systems. *Commun. ACM* **64**(8), 88–97 (2021)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **5**, 135–146 (2017)
3. Bölücü, N., Rybinski, M., Wan, S.: Impact of sample selection on in-context learning for entity extraction from scientific writing. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5090–5107 (2023)
4. Brown, T.E.A.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)