СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО
МАТЕМАТИКА И ИНФОРМАТИКА

SOFIA UNIVERSITY
St. KLIMENT OHRIDSKI
FACULTY OF
MATHEMATICS AND INFORMATICS

Doctoral Thesis

Intelligent Context-Aware Natural Language Dialogue Agent

by

Momchil Emilov Hardalov

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in

Professional field: 4.6 Informatics and Computer Science
Doctoral program: "Software Technologies" – Knowledge Discovery
Department of Software Technologies

Advisor: Professor Ivan Koychev, Ph.D.
Consultant: Professor Preslav Nakov, Ph.D.

October 2022

Abstract

Intelligent Context-Aware Natural Language Dialogue Agent

by

Momchil Hardalov

Doctor of Philosophy in Informatics and Computer Science

Sofia University "St. Kliment Ohridski"

Conversational agents or dialogue systems are computer programs that try to generate human-like responses during a conversation. In task-oriented scenarios, these responses should go beyond simple chitchat, focusing on performing task-specific functions and responding to user requests. In recent years, chatbots and dialogue systems had their Renaissance and gained a lot of attention not only from the research community and also from the industry. Moreover, their architectures evaluated from single domain and rule-based to complex modular multi domain pipelines, built on top of deep neural networks and even end-to-end differentiable models.

In this thesis, I take up on the problem of building efficient task-oriented conversational agents for customer support, and in particular I investigate several important components that can improve the quality of the end-to-end conversational flow and serve better the customer's requests. First, in order to improve the natural language understanding abilities of the agent I propose a novel approach for slot tagging and intent detection based on a pre-trained Transformer that fuses the two tasks together, by using the predicted intent to guide the slot filling, and by using a pooled representation from the task-specific outputs of all tokens for intent detection. Next, I focus on curating answers from external knowledge sources, where I study the abilities of state-of-the-art models for zero-shot multilingual transfer and evaluated the effects of the retrieved evidence passages on the model's abilities to answer user questions. However, my research is not limited to producing short-form answers in a single language, but I also investigate multilingual and cross-lingual approaches for multiple-choice question answering, and retrieval of long-form documents and articles that can serve as explanations. I also explore end-to-end generative models for customer support showing that they can outperform information retrieval-based ones, but they still need additional knowledge grounding in order to overcome hallucination and internal biases. I also introduce a novel neural re-ranking model to improve multi-source response selection, which

is a step towards better contextualizing the conversational agents' responses, as they are ranked by their relevance, and thus expected to be generated by an expert model.

In addition, I collect, and release to the public, three new datasets: (*i*) for multiple-choice QA from high-school exams – one monolingual in Bulgarian, and one multilingual, which covers sixteen diverse languages from eight language families; (*ii*) for detecting previously fact-checked claims, a large-scale English dataset from claims made in social media and their corresponding fact-checking articles.

Acknowledgements

First, I would like like to express my deepest gratitude to my advisors Professor Ivan Koychev and Professor Preslav Nakov, for their guidance and immense support, and for giving me the freedom to develop and pursue my ideas. You have not only been my academic advisors, but also incredible mentors and role models to me. It was a joy and a privilege to work with you.

Next, I would like to thank my collaborators and mentors over the years who supported and guided me. I also thank Professor Preslav Nakov, Professor Ivan Koychev, Professor Galia Angelova, Professor Maria Nisheva, and Milen Chechev, Ph.D. for introducing me to the world of artificial intelligence and natural language processing during my time as a master student in "Information Retrieval and Knowledge Discovery". I thank them for nurturing curiosity and passion for research in me. I am especially grateful to Todor Mihaylov, Dimitrina Zlatkova and Yoan Dinkov for the fruitful discussions and collaborations throughout the years both as part of my doctoral studies and on external projects; it has been a pleasure to work with all of you.

I would like to thank the many wonderful faculty and staff members in the department of "Software Technologies" for giving me the continued opportunity and support to succeed at all phases of my doctoral program.

Last but not least, I would like to thank my wonderful partner Albena, for her unwavering support, patience and for always having faith in me. Finally, I would like to express my appreciation and gratitude to my family for their continuous support and encouragement, throughout the many years of my graduate studies.

Contents

Abstract i

Acknowledgements iii

1 Introduction 1
- 1.1 Dissertation Aims and Objectives . . . 4
- 1.2 Dissertation Structure . . . 5
- 1.3 Published Papers . . . 6

2 Background and Related Work 7
- 2.1 Task-Oriented Conversational Agents . . . 7
- 2.2 Intent Detection and Slot Filling . . . 10
  - 2.2.1 Intent Classification . . . 10
  - 2.2.2 Slot Filling . . . 11
  - 2.2.3 Joint Models . . . 11
- 2.3 Question Answering . . . 12
  - 2.3.1 Machine Reading Comprehension . . . 12
  - 2.3.2 Science QA . . . 13
  - 2.3.3 Multilinguality . . . 14
    - Multi- and Cross-lingual Models . . . 14
    - (Zero-Shot) Multilingual Models . . . 14
- 2.4 Retrieving Long-Form Explanations . . . 15
  - 2.4.1 Previously Fact-Checked Claims . . . 15
  - 2.4.2 Training with Noisy Data . . . 16
- 2.5 Generation Models for Dialogue . . . 16
- 2.6 Multi-Source Response Selection . . . 19
- 2.7 Summary . . . 20

3 Semantic Parsing of Human-Generated Utterances 21
- 3.1 Introduction . . . 22
- 3.2 Dataset . . . 23
- 3.3 Proposed Approach . . . 24
  - 3.3.1 Intent Pooling Attention . . . 24
  - 3.3.2 Slots Modeling . . . 26
  - 3.3.3 Interaction and Learning . . . 27

3.4 Experimental Setup . . . 27
3.4.1 Measures . . . 28
3.4.2 Baselines . . . 28
3.4.3 State-of-the-Art Models . . . 29
3.4.4 Model Details . . . 29
3.5 Experiments and Analysis . . . 30
3.5.1 Evaluation Results . . . 30
3.5.2 Transformer-NLU Analysis . . . 32
3.5.3 Intent Pooling Attention Visualization . . . 32
3.5.4 BERT Knowledge Analysis . . . 33
3.5.5 Error Analysis . . . 35
3.6 Summary . . . 36

4 Curating Answers from External Knowledge Sources 37
4.1 Introduction . . . 38
4.2 Knowledge Retrieval . . . 40
4.2.1 Model . . . 40
Context Retriever . . . 41
BERT for Multiple-Choice RC . . . 41
Answer Selection Strategies . . . 42
4.2.2 Data . . . 43
4.2.3 Experiments and Evaluation . . . 45
BERT Fine-Tuning . . . 45
Wikipedia Retrieval and Indexing . . . 46
Experimental Results . . . 47
4.2.4 Case Study . . . 50
4.3 Answer Retrieval from a Pool of Explanations . . . 53
4.3.1 My Newly Collected Dataset: CrowdChecked . . . 55
Dataset Collection . . . 55
Tweet Collection (Conversation Structure) . . . 56
Comparison to Existing Datasets . . . 56
Data Labeling (Distant Supervision) . . . 57
Feasibility Evaluation . . . 58
Fact-checking Articles Collection . . . 60
4.3.2 Method . . . 60
Training with Noisy Data . . . 62
Re-ranking . . . 62
4.3.3 Experiments . . . 63
Experimental Setup . . . 63
Experimental Results . . . 64
4.3.4 Discussion . . . 69
4.4 Summary . . . 70

5 Advanced Conversation **72**
5.1 Introduction . . . 73
5.2 Dataset for Customer Support Conversations . . . 75
5.3 End-to-End Generative Agent . . . 76
5.3.1 Method . . . 78
Preprocessing . . . 78
Information Retrieval . . . 78
Sequence-to-Sequence . . . 78
Transformer . . . 79
5.3.2 Experiments . . . 79
Evaluation Measures . . . 80
Results . . . 81
5.3.3 Discussion . . . 83
5.4 Multi-Source Response Selection . . . 83
5.4.1 Re-Ranking Model . . . 84
Negative Sampling . . . 85
QANet Architecture . . . 85
Answer Selection . . . 86
5.4.2 Experiments . . . 87
Preprocessing . . . 87
Training Setup . . . 87
Individual Models . . . 88
Evaluation Measures . . . 88
5.4.3 Evaluation Results . . . 88
Auxiliary Task: Question–Answer Appropriateness Classification . . . 88
Answer Selection/Generation: Individual Models . . . 89
Main Task: Multi-Source Answer Re-Ranking . . . 90
5.5 Multi- and Cross-Linguality . . . 92
5.5.1 *Exams* Dataset . . . 93
Dataset Statistics . . . 94
Subjects and Categories . . . 95
Collection and Preparation . . . 96
Data Splits . . . 96
Reasoning and Knowledge Types . . . 97
Subject Analysis . . . 99
5.5.2 Background Knowledge Corpus . . . 100
5.5.3 Baseline Models . . . 101
No Additional Training . . . 102
Fine-Tuned Models . . . 102
5.5.4 Experiments and Results . . . 103
Multilingual Evaluation . . . 103

Knowledge Evaluation . . . 104
Cross-lingual Evaluation . . . 105
5.5.5 Per-Subject Fine-Grained Evaluation . . . 106
5.5.6 Discussion . . . 107
5.6 Summary . . . 108

6 Conclusion and Future Work **110**
6.1 Summary . . . 110
6.2 Contributions . . . 112
6.3 Directions for Future Research . . . 113

Bibliography **117**

A Curating Answers from External Knowledge Sources **155**
A.1 Answer Retrieval from a Pool of Explanations . . . 155
A.1.1 Hyperparameters and Fine-Tuning . . . 155
A.1.2 Annotations . . . 157

B Advanced Conversation **160**
B.1 Multi- and Cross-Linguality . . . 160
B.1.1 Hyperparameters and Fine-Tuning . . . 160
B.1.2 Subject Definitions . . . 162

List of Figures

1.1 Conceptual diagram illustrating the information flow pipeline of a **task-oriented conversational agent**. The agent consists of four main components: (*i*) **natural language understanding** component – that detects the intents and extracts slots from the user's input, (*ii*) **Dialogue Manager** – estimates the user's goal by taking the entire dialog context as an input (*dialogue state tracking*) and generates the next system action (*dialogue policy*), (*iii*) **Utterance Generation** components – maps the dialog act generated by the dialog policy to a natural language utterance, often multiple strategies are implemented (e.g., *language generation* model, filling pre-defined *templates*, querying *external sources* such as relational or graph databases, document stores, ontologies, etc.), and (*iv*) **Utterance Selection** – ranking and selecting the most appropriate utterance for the agent's next turn. Finally, the components I explore in this thesis are marked with ✓, and the ones that are *not* – with ✗. 2

3.1 Model architectures for joint learning of intent and slot filling: (a) classical joint learning with BERT/RoBERTa, and (b) proposed enhanced version of the model. 24
3.2 Intent pooling attention weight for one example per dataset. The thicker the line, the higher the attention weight. 33
3.3 Per-class mean reciprocal rank (MRR) for the two datasets used in my study. 34

4.1 BERT for multiple-choice reasoning. 42
4.2 Accuracy per question category based on the number of query results per answer option. 49
4.3 Crowd fact-checking thread on Twitter. The first tweet (**Post w/ claim**) makes the claim that *Ivermectin causes sterility in men*, which then receives **replies**. A **(crowd) fact-checker** replies with a link to a **verifying article** from a fact-checking website. I pair the *article* with the *tweet that made this claim* (the first post ✓), as it is irrelevant (✗) to the other *replies*. 54
4.4 Distribution of the Jaccard similarity scores. The score is an average of the *sim(tweet, title)* and *sim(tweet, subtitle)*. 59

4.5 Histogram of the year of publication of the Snopes articles included in *CrowdChecked* (my dataset) vs. those in *CheckThat '21*. 61
4.6 MAP@5 for different thresholds and distant supervision approaches. The *Jaccard* and the *Cosine* models are trained only on CrowdChecked, while (*Seq*) and (*Mix*) were trained also on CheckThat '21. 66

5.1 Number of user tweets with replies from customer support per company. 76
5.2 My answer re-ranking framework, based on the QANet architecture. . 84
5.3 Properties and examples from *Eχαμs*. 92
5.4 Relative sizes of the subjects. Those that cover less than 1.5% of the examples are in *Other*. 95
5.5 Relative sizes of reasoning types in *Eχαμs*. 98
5.6 Relative size of the *Eχαμs* knowledge types. 99
5.7 Fine-grained evaluation by language and school subjects. 106
5.8 Fine-grained evaluation by language and school subjects in *Social Science* and *Other*. 107

List of Tables

3.1 Example from the SNIPS dataset with slots encoded in the BIO format. The utterance's intent is *PlayMusic*, and the given slots are *year* and *artist*. . . . 22
3.2 Statistics about the ATIS and SNIPS datasets. . . . 24
3.3 Intent detection and slot filling results on the SNIPS and the ATIS datasets. Highest results in each category are written in **bold**. My models are shown in *italic*; the non-italic models on top come from the literature. Qin et al. (2019, 2020) report their results with single precision. . . . 30
3.4 Comparing *Transformer-NLU:BERT* to the two baselines: *(i)* current SOTA for each measure, and *(ii)* conventionally fine-tuned BERT-Joint without the improvements, in terms of relative error reduction (Eq. 3.10). . . . 31

4.1 Statistics about my Bulgarian dataset compared to the RACE dataset. 43
4.2 Example questions, one per subject, from the Bulgarian dataset. The correct answer is marked in green. . . . 44
4.3 Accuracy measured on the dev RACE dataset after each training epoch. 46
4.4 Accuracy on the Bulgarian testset: ablation study when sequentially adding/removing different model components. . . . 48
4.5 Evaluation results for the Bulgarian multiple-choice reading comprehension task: comparison of various indexing and query strategies. . 51
4.6 Retrieved unique top-1 contexts for the example questions in Table 4.2. The passages are retrieved using queries formed by concatenating a question with an answer option. . . . 52
4.7 Illustrative examples for the task of detecting previously fact-checked claims. The **post contains a claim** (related to *legislation and dictatorship*), the **Verified Claims** are part of a search collection of previous fact-checks. In row (*1*), the fact-check is a correct match for the claim made in the tweet (✓), whereas in (2), the claim still discusses *Sen. Mitch McConnell*, but it is a different claim (✗), and thus it forms an incorrect pair. . . . 55
4.8 Statistics about out dataset vs.CheckThat '21. ‡The number of unique tweets is lower compared to the total number of tweet–article pairs, as one tweet can be fact-checked by multiple articles. . . . 57

4.9 Proportion of examples in different bins based on average Jaccard similarity between the tweet ↔ the title/subtitle. Manual annotations of *correct pairs* (i.e., tweet–article pairs, where the article fact-checks the claim in the tweet). 58
4.10 Proportion of examples in different bins based on cosine similarity using Sentence-BERT trained on *CheckThat '21*. Manual annotations of *correct pairs*. 58
4.11 Statistics about my collected datasets in terms of tweet–verifying article pairs. 63
4.12 Evaluation on the CheckThat '21 testing set. In parenthesis is name of the training split, i.e., *Jac*card or *Cos*ine selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two. The highest results are in **bold**. 65
4.13 Results on CheckThat '21 (dev and test). I compare my model and its components (added sequentially) to the state of the art. The best results are in **bold**. 67
4.14 Evaluation on the CheckThat '21 **development** set. In parenthesis is name the training split, i.e., Jaccard (*jac*) or Cosine (*cos*) data selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two datasets. 68
4.15 Evaluation on the CheckThat '21 **testing** set. In parenthesis is name the training split, i.e., Jaccard (*jac*) or Cosine (*cos*) data selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two datasets. 69
4.16 Results on the CheckThat '21 **testing** set. I compare my model and its components (added sequentially) to state-of-the-art approaches. . . 70

5.1 Overall statistics about the dataset. 77
5.2 Statistics about the dataset. 77
5.3 Results based on word-overlap measures. 81
5.4 Results based on semantic measures. 81
5.5 Chatbot responses. The first column shows the original question and the gold customer support answer, while the second column shows responses by the models. 82
5.6 Auxiliary task: question–answer appropriateness classification results. 89
5.7 Main task: performance of the individual models. Single model results results are reported in Section 5.3.2, Tables 5.3 and 5.4 90
5.8 Main task: re-ranking the top $K = 5$ answers returned by the IR and the Seq2seq models. 91
5.9 Statistics about *Eχαμs*. The average length of the question (*Question Len*) and the choices (*Choice Len*) are measured in number of tokens, and the vocabulary size (*Vocab*) is measured in number of words. . . 93

5.10 Parallel questions for different language pairs. 94
5.11 Number of examples in the data splits based on the experimental setup. 97
5.12 Per-subject statistics. The grade is High (H), and Middle (M). The average length of the question (*Q Len*) and the choices (*Ch Len*) are measured in number of tokens, and the vocabulary size (*Vocab*) is shown in number of words. 100
5.13 Description of the per-language indices used as a source of background knowledge in my experiments. 101
5.14 Overall per-language evaluation. The first three columns show the results on ARC Easy (E), ARC Challenge (C), and Regents 12 LivEnv (en). The following columns show the per-language and the overall results (the last column All) for all languages. *All* is the score averaged over all $E\chi\alpha\mu s$ questions. 104
5.15 Cross-lingual zero-shot performance on $E\chi\alpha\mu s$. The first three columns show the performance on the test set of the AI2 science datasets (English), followed by per-language evaluation. The underlined values mark languages that have parallel data with the source language, and the ones with an asterisk* are from the same family. . . 105

A.1 Fleiss Kappa inter-annotator agreement between my three annotators (A, B, C). 157
A.2 Cohen Kappa inter-annotator agreement between the three annotators (A, B, C). 157
A.3 Examples from CrowdChecked, showing correct (✓) and incorrect matches (✗). The examples in each group are sorted by their overlap with the claim made in the tweet. 159

B.1 The hyper-parameter values I used for fine-tuning. 161

List of Abbreviations

BLEU bilingual evaluation understudy xii, 74, 77, 80, 81, 83, 84, 89, 90, 91, 108, 112

BPE byte pair encoding xii, 27

CNN Convolutional Neural Network xii, 11, 18, 19, 22

CRF conditional random fields xii, 11, 22, 26, 27, 28, 32, 36

IR information retrieval i, xi, xii, 5, 19, 38, 74, 77, 78, 79, 81, 82, 88, 89, 90, 91, 102, 108, 110, 112

LCS longest common subsequence xii, 81, 89

LPT large pre-trained Transformers xii, 18, 20, 29, 39, 73, 114

LSTM Long Short-Term Memory xii, 11, 12, 18, 29, 78, 79, 85, 88

MAP mean average precision ix, xii, 64, 65, 66, 67, 68, 69, 70, 71, 111, 112, 155, 156

MEMM maximum entropy Markov model xii, 11, 22

MNR multiple negatives ranking xii, 60, 62, 67

MRC machine reading comprehension xii, 5, 12, 13, 15, 42, 47

MRR mean reciprocal rank viii, xii, 34, 64, 68, 69, 70, 71, 112

NER named entity recognition xii, 11, 12, 26

NLG natural language generation xii, 3, 4, 7, 9, 10

NLL negative log-likelihood xii, 27, 32

NLP natural language processing xii, 4, 9, 14, 21, 27, 41, 73, 92, 110, 115

NLU natural language understanding i, viii, xii, 2, 3, 4, 5, 7, 8, 9, 19, 21, 22, 23, 33, 36, 72, 110, 112, 114

NMT Neural Machine Translation xii, 15

POMDPs partially observable Markov decision processes xii, 8

POS part-of-speech xii, 11, 13, 26

QA question answering ii, xii, 5, 7, 13, 14, 20, 75, 92, 93, 102, 103, 107, 108, 110, 111, 113

RER relative error reduction x, xii, 31, 36

RL reinforcement learning xii, 9, 18

RNN Recurrent Neural Network xii, 11, 18, 22, 29, 42, 78, 111

ROUGE recall-oriented understudy for gisting evaluation xii, 74, 77, 80, 81, 83, 84, 89, 90, 91, 108, 112

Seq2seq sequence-to-sequence xi, xii, 5, 9, 15, 17, 18, 19, 29, 72, 78, 79, 80, 81, 82, 83, 88, 89, 90, 91, 108

SMT Statistical Machine Translation xii, 19

SOTA state-of-the-art x, xii, 3, 5, 7, 11, 13, 14, 18, 30, 31, 64, 69, 72, 84, 111, 113

TF.IDF term frequency–inverse document frequency xii, 61, 63, 64, 67, 68, 70, 78, 79, 88

To Albena

Chapter 1

Introduction

Internet has transformed many areas of our everyday lives. It made a whole new range of services and products available to a global audience from around the world. In turn, this has changed the way companies and businesses operate and interact with their clients. A major, rapidly growing aspect of their operations is the demand for better and more reliable customer support, not only in terms of how accurate the information provided by the operator or an automated system is, but also how fast a solution to a particular problem or request is reached. Moreover, these services must be accessible by the customers, on one hand, throughout their preferred channels of communication, and on the other, in their most convenient language as well. Although, conversation with human experts is more likely to end in better customer experience, it becomes more and more clear that recruiting and training new employees becomes infeasibly fast, as it is an expensive and time-consuming process that cannot keep up with the ever growing rate of adopting new users. This is a clear sign that further automation with conversation agents and development of better question answering systems, in addition to new and improved tools for customer service operators, are urgently needed.

First, let me give a formal definition of *conversational agent*. The following definition will be used throughout this thesis: *"A conversational agent also referred to as chatbot is a computer program which tries to generate human like responses during a conversation."* (Ramesh et al., 2017). Next, I focus on the following three research questions outlined by Gao et al. (2019), in order to scope the problems that conversational agents are expected to solve:

- **question answering**: *"the agent needs to provide concise, direct answers to user queries based on rich knowledge drawn from various data sources including text collections such as Web documents and pre-compiled knowledge bases such as sales and marketing datasets"*;
- **task completion**: *"the agent needs to accomplish user tasks ranging from restaurant reservation to meeting scheduling, and to business trip planning"*;

- **social chat**: *"the agent needs to converse seamlessly and appropriately with users – like a human as in the Turing test – and provide useful recommendations."*.

In order to further quantify the current state of the field, I focus on recently reported metrics in real-world studies. First, it is important to emphasize that conversational agents are gaining more trust both from the companies and from the customers, and they are becoming an integral part of the customer service pipeline. Drift's 2020 *State of Conversational Marketing report*,[1] reported that the usage of chatbots as a brand communication channel increased by 92% compared to the previous year. In the Zendesk report,[2] it is noted that 69% of the customers say they are willing to interact with a bot on simple issues, which is a 23% increase from the previous year. According to Invesp, 33% of the consumers would rather contact a company's customer service via social media rather than by phone.[3] However, 54% of the customers said that their biggest frustration with chatbots was the number of questions they must answer before being transferred to a human agent.[2] Moreover, customers are concerned with the "understanding" capabilities of the conversational agents, 60% of them think humans are able to understand their needs better than chatbots.[4] Furthermore, users note the chatbots' "inability to solve complex issues" as another major concern of theirs.[5]

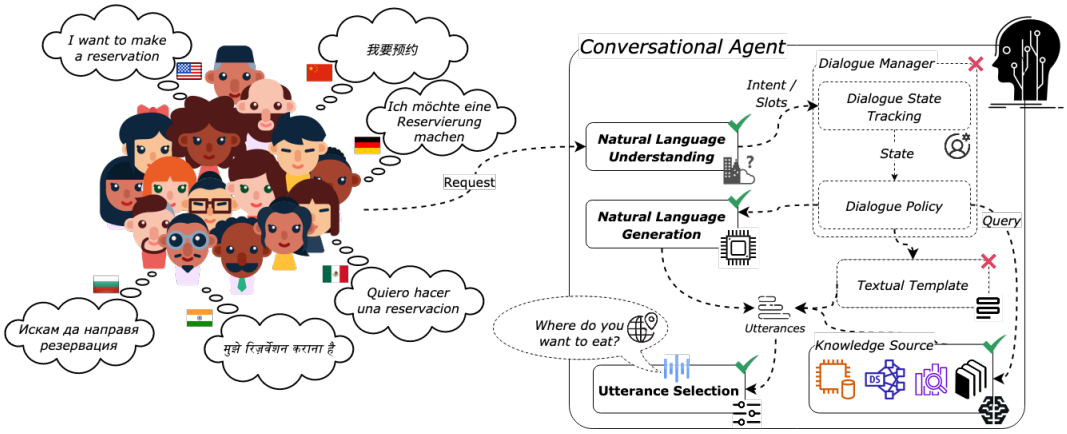


Figure 1.1: Conceptual diagram illustrating the information flow pipeline of a **task-oriented conversational agent**. The agent consists of four main components: (*i*) **natural language understanding** component – that detects the intents and extracts slots from the user's input, (*ii*) **Dialogue Manager** – estimates the user's goal by taking the entire dialog context as an input (*dialogue state tracking*) and generates the next system action (*dialogue policy*), (*iii*) **Utterance Generation** components – maps the dialog act generated by the dialog policy to a natural language utterance, often multiple strategies are implemented (e.g., *language generation* model, filling pre-defined *templates*, querying *external sources* such as relational or graph databases, document stores, ontologies, etc.), and (*iv*) **Utterance Selection** – ranking and selecting the most appropriate utterance for the agent's next turn. Finally, the components I explore in this thesis are marked with ✓, and the ones that are *not* – with ✗.

[1] https://www.drift.com/blog/state-of-conversational-marketing/
[2] ,https://cx-trends-report-2022.zendesk.com/growth-areas
[3] https://www.invespcro.com/blog/social-media-customer-support/
[4] https://userlike.com/en/blog/consumer-chatbot-perceptions
[5] https://startupbonsai.com/chatbot-statistics/

Based on the definitions and the outlined trends, I direct the main focus of this thesis towards task-oriented conversational agents and their building components in the domain of customers support. However, maintaining efficient dialogue is a complex multi-turn process that depends widely on the context. This context is not limited only to the current conversation, but it also depends on prior commonsense and domain knowledge about the world that the participants have. All of these aforementioned factors make it extremely challenging for machine learning models to produce consistent and factual utterances, especially in an environment that requires extensive domain expertise, and going beyond simple chit-chat. That is the reason conversational agents often depend on multiple components to interpret and to respond to the users' requests and queries.

In Figure 1.1, I illustrate the main components in the pipeline of a conversational agent. The first component that the user request is processed through is the *natural language understanding* (Weld et al., 2022) one. It is responsible for the general understanding of the input, and thus the name of the module. Its main tasks are (*i*) to detect the intent and (*ii*) extract the values for the relevant slots from the input tokens and pass them to the *Dialogue Manager*. In turn, the *Dialogue Manager* aggregates the entire dialogue context, called dialogue state tracking (Williams et al., 2016), estimates the user's goal and generates the next system action, i.e., the *dialogue policy*. A simple solution for implementing a dialogue manager is to create a large hand-designed semantic grammars with thousands of rules (Larsson and Traum, 2000; Zue et al., 2000; Henderson, 2015; Yan et al., 2017); however, such rule-based systems are hard to scale and update in a multi-domain scenario. Currently, state-of-the-art neural networks architectures and refinement learning approaches are becoming an integral part of dialogue managers, both for state tracking (Wen et al., 2017; Wu et al., 2019; Zhong et al., 2018) and dialog policy (Young et al., 2010; Cuayáhuitl et al., 2015; Peng et al., 2018; Su et al., 2018; Wu et al., 2019). Nonetheless, in this thesis I do not study approaches related to the *Dialogue Manager*. My focus is on improving the natural language understanding abilities and the quality of the answers and the generated utterances (discussed below), not only in terms of factually, but also in terms of consistency and relevance to the user's input.

The next step in the conversational agent's pipeline is to map the dialog act generated by the dialog policy to a natural language utterance (Gatt and Krahmer, 2018; Dong et al., 2022). To achieve this, often multiple strategies are implemented such as natural language generation (NLG) models, filling pre-defined textual templates or extracting data from external knowledge sources. The templates are an integral part of task-oriented dialogue (Williams and Zweig, 2016; Wen et al., 2017). They guarantee consistent and well-written sentences, albeit they suffer from the same issues as rule-based systems – they are static and should be prepared beforehand. Moreover, the agent becomes less flexible, and the dialogue sounds less natural and diverse. Hereby, I do not study them further in this thesis. On the

other hand, the NLG models are answering user questions with external knowledge sources such as retrieving long-form answers or finding evidence passages.[6]

The final part of the pipeline is the *next utterance selection* model. In the case of a single natural language generation (or similar) source, this model should copy the text as the chatbot's next turn, i.e., to be bypassed in the pipeline. However, in the case of multiple generation strategies, the conversational agent needs to choose the most relevant sentence from the list of candidates, and thus this component is responsible for re-ranking and choosing the most appropriate option from this list. The decision can again be based on a pre-defined scenario. Here, I explore more complex methods based on deep neural networks (Qiu et al., 2017; Cui et al., 2017; Clarke et al., 2022).

1.1 Dissertation Aims and Objectives

The *aims* of this thesis can be summarized as follows:

1. Develop efficient natural language processing-based approaches for building multi-component, task-orient, context-aware conversational agents, with the specific application for serving as customer support chatbots.
2. Create new resources and corpora that can help in the development of dialogue agents, on one hand, extending them to multiple languages, and on the other hand, allowing for generating long-form answers (e.g., articles from knowledge bases), as opposed to the common short ones.

In this regard, I outline the following research *objectives*:

- Survey the existing literature, previous work and approaches on conversational agents and their components.
- Design, describe, implement, and evaluate an natural language understanding (NLU)-based component that jointly identifies the user intent and recognizes what is relevant to its slots.
- Design, describe, implement, and evaluate an algorithm for curating utterances from external knowledge sources.
- Design, describe, implement, and evaluate an end-to-end generative models for customer support chatbots.
- Design, describe, implement, and evaluate a pipeline for multilingual and cross-lingual dialogue.

[6]Customers prefer knowledge bases over all other self-service channels. https://www.hubspot.com/knowledge-base

1.2 Dissertation Structure

The rest of this thesis is organized as follows:

- In Chapter 2, I review state-of-the-art approaches related to conversation agents and their building components. First, I start by reviewing previous work on task-oriented conversational agents – including modularized and end-to-end (differentiable) dialogue systems. Second, I cover approaches relevant to two of the main natural language understanding tasks in task-oriented dialogue – intent classification, slot filling and their joint modeling. Then, I survey methods for question answering (QA) and machine reading comprehension (MRC), zooming into science QA datasets, multilingual models and approaches for cross-lingual transfer. Next, I summarize previous work on retrieval long-form explanations through the lenses of the task of *detecting previously fact-checked claims*. Finally, I discuss advanced conversational agents such as end-to-end generative ones, and strategies to combine responses from different sources, e.g., retrieved from previous conversations, generated using a sequence-to-sequence model or by filling pre-defined templates.

- In Chapter 3, I describe a novel method for joint intent detection and slot filling. The main idea is to better leverage the connection between the two tasks. For this purpose, the representations of the two tasks are fused together while training the model, on one hand, by an intent pooling attention mechanism, and on the other, by slot modeling via concatenating the token-level representations from the language model with the predicted intent distribution, and finally adding hand-crafted features. I further demonstrate SOTA results on two standard NLU datasets, namely ATIS (Hemphill et al., 1990) and SNIPS (Coucke et al., 2018).

- Chapter 4 introduces new methods for curating answers from external knowledge sources. First, I present a new dataset for multiple-choice question answering in Bulgarian, and I evaluate information retrieval-based methods, in order to obtain evidence passages. This is further combined with zero-shot transferred model from high-resource language (i.e., English). Next, I present a novel method for obtaining long-form answers, i.e., explanations in the context of detecting previously fact-checked claims. In particular, I describe a novel weakly supervised method for collecting large-scale datasets of article–claim pairs, and learning from them with techniques for model self-adaptation to training on noisy data.

- In Chapter 5, I explore methods for advanced conversation. First, I study end-to-end generative agents learned from conversation logs, collected from Social Media, between a company's customer support operator and a client. Next, I introduce a new framework for multi-source response selection using a neural network-based re-ranking model. Finally, I present a new multi- and

cross-lingual, question answering dataset, and explore the abilities of several state-of-the-art multilingual models to transfer knowledge across languages.

- Chapter 6 concludes the thesis, summarizes the contributions, and discusses future research directions.

1.3 Published Papers

- Momchil Hardalov, Anton Chernyavskiy, Ivan Koychev, Dmitry Ilvovsky, and Preslav Nakov. 2022b. CrowdChecked: Detecting Previously Fact-Checked Claims in Social Media. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, AACL-IJCNLP '22, Online
- Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020b. EXAMS: A Multi-subject High School Examinations Dataset for Cross-lingual and Multilingual Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 5427–5444, Online
- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2020a. Enriched Pre-trained Transformers for Joint Slot Filling and Intent Detection. *arXiv preprint arXiv:2004.14848*
- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019a. Beyond English-Only Reading Comprehension: Experiments in Zero-shot Multilingual Transfer for Bulgarian. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '19, pages 447–459, Varna, Bulgaria
- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019b. Machine Reading Comprehension for Answer Re-Ranking in Customer Support Chatbots. *Information*, 10(3)
- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards Automated Customer Support. In *Proceedings of the 18th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, AIMSA '18, pages 48–59, Varna, Bulgaria

Chapter 2

Background and Related Work

In this chapter, I review recent work in the field of conversational agents that are relevant to this thesis. First, in Section 2.1, I survey a wide range of holistic approaches to conversational agents, including datasets used for training. Here, I provide sufficient background for the rest of the thesis. In the rest of the chapter, I zoom into the building components and closely related tasks, included in the agent's pipeline such as natural language understanding, knowledge retrieval from external sources, question answering, natural language generation, among others.

In Section 2.2, I summarize the literature for two conversational NLU tasks: (*i*) intent detection, i.e., understanding the user's current goal, and (*ii*) slot filling, i.e., identifying different slots in the running dialog, which correspond to different parameters of the user's query. Next, I focus on the related topic of question answering (see Section 2.3), covering full resource and zero-shot approaches applied in mono- and multilingual scenario.

One of the main research directions covered in this thesis is the problem of curating answers from external knowledge sources. I study this problem through the lenses of *finding previously fact-checked claims*, and thus in Section 2.4 I give the needed background for this task and the state-of-the-art approaches and models, including training with noisy data and distant supervision.

Finally, in Sections 2.5 and 2.6, I focus on advances in conversation techniques, i.e., generative models for dialogue and combining answers obtained from multiple sources in order to find the best next utterance in a conversation.

2.1 Task-Oriented Conversational Agents

Conversational agents, also referred to as dialogue systems or chatbots, cover a wide range of applications, based on which they can be classified into two major categories – open-domain and task-oriented. They are built on top of a multi-stage pipeline that receives a user request or query as an input, and takes several steps towards understating and processing the request, followed by generating a response or taking a pre-defined action. There are several recent survey that focus

on different aspects of the conversational agents: Tsur et al. (2016) and Gao et al. (2019) give a holistic overview of dialogue systems and their challenges, Zhang et al. (2020) covers all components of end-to-end task-oriented dialogue, some of the work even discusses individual modules such as dialogue state tracking (Balaraman et al., 2021), natural language understanding (Weld et al., 2022), and response generation (Fan et al., 2020; Tao et al., 2021), among others.

A common approach towards building end-to-end task-oriented dialogue systems is to use a modular architecture, i.e., an architecture that has a separate component (or module) that is responsible for specific conversational skill, or combines multiple, related skills. Nonetheless, these components are often trained in isolation, and combined only through the pipeline, thus making the system non-differentiable. However, these systems are easy to implement and train, and therefore they are a viable solution from practical standpoint. One of the first frameworks built for dialogue agents were based on an explicit Bayesian model of uncertainty, optimizing a policy via a reward-driven process, and partially observable Markov decision processes (POMDPs) Young et al. (2013). However they covered limited domains such as tourist service, appointment scheduling, hardware troubleshooting, among others Williams (2007); Kim et al. (2008); Williams (2008); Thomson and Young (2010); Janarthanam et al. (2011). More recently, neural networks have become the dominant approach for training such conversational agents. Wen et al. (2017) introduced a system, where each module is backed by a neural network. The proposed approach relies on non-differentiable knowledge-base lookup operators, and each component is learned separately in a supervised fashion. Zhang et al. (2020) proposed a framework to utilize the conversations' property that one context can have multiple responses, in order to generate diverse responses. In particular, they first summarize the dialogue history into the dialogue state, and then map all valid system actions. Moreover, during training they augment the dataset by generating new state-action pairs from the history. Sun et al. (2022) took a step further towards better encoding the context into the dialogue state and proposed a back and denoising procedure, to allow the model to recover from mistakes made by preceding modules in the pipeline; in turn, this reduces the noise in the generation output.

Another promising research direction in building conversational systems is end-to-end differential approaches. They enable the agent to propagate errors and share knowledge between the different modules throughout the system. One option is to use methods based on a single model, i.e., training a multi-skill neural network. Zhao and Eskenazi (2016) showed one of the first attempts for building such systems; they combined the natural language understanding and the dialogue manager components into a single module. Next, Li et al. (2017) argued that downstream modules are affected by earlier modules, and that the performance of the entire system was not robust to the accumulated errors; thus, they presented an

end-to-end learning framework based on reinforcement learning (RL) and direct database queries. Lei et al. (2018) introduced a simplified approach based on a two-stage sequence-to-sequence (Seq2seq) model with an additional copy mechanism that combines state tacking and response generation into a joint model. Eric et al. (2017) addressed the aforementioned challenges in non-differentiable agents using "soft" knowledge-base lookups. Memory networks (Sukhbaatar et al., 2015) are another promising approach for learning task-oriented dialogue agents. Bordes et al. (2017) evaluated their capabilities on a newly introduced testbed designed to outline the strengths and weaknesses of end-to-end conversational systems in goal-oriented applications. Similarly, Madotto et al. (2018) adopted memory networks with an improved mechanisms to incorporate external information from knowledge bases using pointer networks Vinyals et al. (2015). However, these approaches have an internal memory with limited capacity, that cannot store all the needed information. In order to mitigate this, Xu et al. (2019) tried to extend the memory and to encode additional expert knowledge with graphs using a Knowledge-routed Deep Q-network (KR-DQN). The proposed framework is responsible for both managing topic transitions, and a knowledge-routed graph branch for topic decision-making.

Currently, the Transfomer-based Vaswani et al. (2017) models are the dominating architecture in NLP; unsurprisingly, they have also been adopted in conversational agents as well. Romero et al. (2021) tried to improve the agent's context retrieval and symbolic reasoning abilities by embedding a Transfomer model in an operational loop that blends both natural language generation and symbolic injection. Hung et al. (2022) focused on domain specialization for task-oriented dialogue systems by first automatically extracting salient domain-specific terms, and then using the extracted terms to construct resources used for domain-specific pretraining using conventional mask language modeling objective (Devlin et al., 2019) and adapter-based approaches. Su et al. (2022) introduced a novel multi-task pretraining strategy that allows the model to learn the primary task completion skills embedded in dialogue systems — NLU, dialogue state tracking, dialogue policy, and NLG — in a sequence generation fashion using an encoder-decoder architecture and heterogeneous dialog corpora.

Typically, conversational agents are trained on static, manually annotated datasets. Moreover, the data sources and knowledge bases they rely on are also compiled by human experts. That said, it is clear that obtaining high-quality data is both time-consuming and expensive, especially when in comes to human–human or even human–machine dialogues. On one hand, this makes scaling conversational agents difficult, and on the other hand, it limits their NLU capabilities due to the limited data that they use both for training (data points) and for inference (knowledge base size). Continual learning is one promising methodology that can help models learn from their real-world interactions with the users. Lee (2017) explored

continuous learning (CL) based on regularization techniques in order to avoid catastrophic forgetting in training a task-oriented conversational agent on three domains learned sequentially. More recently, Madotto et al. (2021) extended CL to more domains (37), introducing a new benchmark. Moreover, they proposed a simple yet effective approach based on residual adapters both in modularized and end-to-end scenario. Liu and Mazumder (2021) proposed a lifelong interactive learning that relays on automatically obtained training data from actual multi-party conversations. More precisely, they collect data from four different sources: (*i*) dialogue history, (*ii*) asking the current user for clarifications, (*iii*) asking another user from the conversational thread, (*iv*) observing user demonstrations. This allows the bot, on one hand, to converse better – learn the user behaviors, emotions, preferences, i.e., be context-aware, and on the other hand, enrich its knowledge base with additional facts, either by asking clarification questions, or inferring from the dialogue history. Finally, it is worth mentioning that continual learning can be applied also to specific components from the agent's pipeline, e.g., dialogue state tracking (Mi et al., 2020), natural language generation (Wu et al., 2019), etc.

Finally, there have been efforts towards building social bots, and open-domain conversation agents from big tech companies, some examples include – XioIce from Microsoft (Zhou et al., 2020), Amazon's Alexa Challange (Ram et al., 2018), Google's Meena (Adiwardana et al., 2020), Meta's BlenderBot (Roller et al., 2021; Shuster et al., 2022) and BlendedSkillTalk dataset, that allows bots to blend different skills into one cohesive conversational flow (Smith et al., 2020). While open-domain agents often share similar architectures and modules with task-oriented chatbots, they focus on chitchat rather than on performing task-specific functions. Nonetheless, building such bots is an important and challenging task, as they must retain a set of conversational skills: provide engaging talking points, and display knowledge, empathy and personality appropriately, while maintaining a consistent persona (Roller et al., 2021). In this thesis, my focus is on task-oriented agents, and thus I do not survey previous work on open-domain dialogue.

2.2 Intent Detection and Slot Filling

Previous work did not necessarily focus on the joint modeling of the intent classification and the slot filling tasks. In this section, I first cover approaches that address each task individually, and then I present joint models.

2.2.1 Intent Classification

Several approaches have focused only on the utterance intent, and have ignored slot information. For example, Hu et al. (2009) mapped each intent domain and user's queries into a Wikipedia representation space, Kim et al. (2017) and Xu and Sarikaya (2013) used log-linear models with multiple-stages and word features. Ravuri and

Stolcke (2015) investigate word and character *n*-gram language models based on Recurrent Neural Network and LSTMs (Hochreiter and Schmidhuber, 1997), Xia et al. (2018) proposed a zero-shot transfer thought Capsule Networks (Sabour et al., 2017) and semantic features for detecting the user intent, without labeled data. Moreover, some work addressed the task in a multi-class multi-label setup (Xu and Sarikaya, 2013; Kim et al., 2017; Gangadharaiah and Narayanaswamy, 2019).

2.2.2 Slot Filling

Before the rise of deep learning, sequential models such as maximum entropy Markov model (MEMM) (Toutanvoa and Manning, 2000; McCallum et al., 2000) and conditional random fields Lafferty et al. (2001); Jeong and Lee (2008) were the state-of-the-art choice. Recently, several combinations thereof and different neural network architecture were proposed (Xu and Sarikaya, 2013; Huang et al., 2015; E et al., 2019). Zhu et al. (2020) explored label embeddings from slots filling and different kinds of prior knowledge such as: atomic concepts, slot descriptions, and slot exemplars. Zhang et al. (2020) used time-delayed neural networks achieving state-of-the-art performance. Siddique et al. (2021) investigated zero-shot transfer of the slot filling knowledge between different tasks. However, a steer away from sequential models is observed in favor of self-attentive ones such as the Transformer (Vaswani et al., 2017; Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2020; Lewis et al., 2020). They compose a contextualized representation of both a sentence, and each of its words, through a sequence of intermediate non-linear hidden layers, usually followed by a projection layer, in order to obtain per-token tags. Such models were successfully applied to closely related tasks, e.g., named entity recognition (NER) (Devlin et al., 2019), part-of-speech (POS) tagging (Tsai et al., 2019), etc.

Approaches modeling the intent or the slot as independent of each other suffer from uncertainty propagation due to the absence of shared knowledge between the tasks. In order to overcome this limitation, I learn both tasks using a joint model.

2.2.3 Joint Models

Given the correlation between the intent and word-level slot tags, it is natural to train them jointly. Recent surveys covered different aspects of joint and individual modeling of the slot and the intent (Louvan and Magnini, 2020; Weld et al., 2022).

Xu and Sarikaya (2013) introduced a shared intent and slot hidden state Convolutional Neural Network (CNN) (LeCun et al., 1989), followed by a globally normalized CRF (TriCRF) for sequence tagging. Since then, Recurrent Neural Network have been dominating, e.g., Hakkani-Tür et al. (2016) used bidirectional LSTMs for slot filling and the last hidden state for intent classification, Liu and Lane (2016) introduced shared attention weights between the slot and the intent layer. Goo et al.

(2018) integrated the intent via a gating mechanism into the context vector of LSTM cells used for slot filling.

Qin et al. (2019) used a self-attentive bidirectional LSTM encoder for the input utterances and a dual decoder for the intents and the slots, and they applied both at the token-level. E et al. (2019) introduced a bidirectional interrelated model, using an iterative mechanism to correct the predicted intent and the slot by multiple step refinement. Zhang et al. (2019) tried to exploit the semantic hierarchical relationship between words, slots, and intent via a dynamic routing-by-agreement schema in Capsule Networks (Sabour et al., 2017). Qin et al. (2020) proposed an adaptive graph-interactive framework using BiLSTMs and graph attention networks (GAT, Velickovic et al. (2018)) to model the interaction between intents and slots at the token level. Recently, Qin et al. (2021) introduced a co-interactive Transformer that mixes the slot and the intent information by building a bidirectional connection between them.

Here, I use a pre-trained Transformer, and instead of depending only on the language model's hidden state to learn the interaction between the slot and the intent, I fuse the two tasks together. Namely, I guide the slot filling by the predicted intent, and I use a pooled representation from the task-specific outputs of BERT for intent detection. Moreover, I leverage information from external sources: *(i)* from explicit NER and true case annotations, and *(ii)* from implicit information learned by the language model during its extensive pre-training.

2.3 Question Answering

2.3.1 Machine Reading Comprehension

The growing interest in machine reading comprehension (MRC) has led to the release of various datasets for both extractive (Nguyen et al., 2016; Trischler et al., 2017; Joshi et al., 2017; Rajpurkar et al., 2018; Reddy et al., 2019) and non-extractive (Richardson et al., 2013; Peñas et al., 2014; Lai et al., 2017; Clark et al., 2018; Mihaylov et al., 2018; Sun et al., 2019) comprehension. My work primarily focuses on the non-extractive multiple-choice type, designed by educational experts, since their task is very close to my newly-proposed dataset, and are expected to be well-structured and error-free (Sun et al., 2019).

These datasets brought a variety of models and approaches. The usage of external knowledge has been an interesting topic, e.g., Chen et al. (2017) used Wikipedia knowledge for answering open-domain questions, Pan et al. (2019) applied entity discovery and linking as a source of prior knowledge. Sun et al. (2019) explored different reading strategies such as back and forth reading, highlighting, and self-assessment. Ni et al. (2019) focused on finding essential terms and removing distraction words, followed by reformulation of the question, in order to find better

evidence before sending a query to the MRC system. A simpler approach was presented by Clark et al. (2016), who leveraged information retrieval, corpus statistics, and simple inference over a semi-automatically constructed knowledge base for answering fourth-grade science questions.

Current state-of-the-art approaches in machine reading comprehension are grounded on transfer learning and fine-tuning of language models (Peters et al., 2018; Conneau et al., 2018; Devlin et al., 2019). Yang et al. (2019) presented an open-domain extractive reader based on BERT (Devlin et al., 2019). Radford et al. (2018) used generative pre-training of a Transformer (Vaswani et al., 2017) as a language model, transferring it to downstream tasks such as natural language understanding, reading comprehension, etc.

Finally, Peñas et al. (2012) introduced a small-sized Bulgarian MRC dataset. It was adopted by Simov et al. (2012) for their experiments. They converted the question-answer pairs to declarative sentences, and measured their similarity to the context, transforming both to a bag of linguistic units: lemmata, POS tags, and dependency relations.

2.3.2 Science QA

Work in Science Question Answering emerged in recent years with the development of several challenging datasets. The most notable is ARC (Clark et al., 2018), which is a QA reasoning challenge that contains both *Easy* and *Challenge* questions from 4th to 8th grade examinations in the *Natural Science* domain. As in *Exámus*, the questions in ARC are created by experts, albeit my dataset covers a wide variety of high school (8th-12th grade) subjects including but not limited to, Natural Sciences, Social Sciences, Applied Studies, Arts, Religion, etc.

The early versions of ARC (Clark, 2015; Schoenick et al., 2017) inspired several crowdsourced datasets: Welbl et al. (2017) proposed a scalable approach for crowdsourcing science questions given a set of basic supporting science facts. Dalvi et al. (2019) focused on specific phenomena including understanding science procedural texts, Mihaylov et al. (2018) and Khot et al. (2020) studied multi-step reasoning, given a set of science facts and commonsense knowledge, Tafjord et al. (2019), and Mitra et al. (2019) worked on reasoning about qualitative relationships, and declarative texts, among others. Unlike these English-only datasets, *Exámus* offers questions in 16 languages. Moreover, it contains questions about multiple subjects, which are presumably harder as they were extracted mostly from matriculation examinations (8-12th grade). Finally, *Exámus* contains over 24,000 questions, which is more than three times as many as in ARC.

2.3.3 Multilinguality

Multi- and Cross-lingual Models

Recently, several QA datasets have been created that cover languages other than English, but still focusing on one such language. Gupta et al. (2018) proposed a parallel QA task for English and Hindi, Liu et al. (2019) collected a bilingual cloze-style dataset in Chinese and English. Jing et al. (2019) crowdsourced parallel paragraphs from novels in Chinese and English. A few datasets investigated multiple-choice school QA (Hardalov et al., 2019; Van Nguyena et al., 2020), albeit in a limited domain, and for lower school grades (1st-5th). Other efforts focused on building bi-lingual datasets that are similar in spirit to SQuAD (Rajpurkar et al., 2016) – extractive reading comprehension over open-domain articles. Such datasets are collected by crowdsourcing questions, following a procedure similar to (Rajpurkar et al., 2016), in Russian (Efimov et al., 2020), Korean (Lim et al., 2019), French (d'Hoffschmidt et al., 2020), or by translating existing English QA pairs to Spanish (Carrino et al., 2020).

Recently, some multilingual datasets, were released to the public. MLQA (Lewis et al., 2020), and XQuAD (Artetxe et al., 2020) use translations by professionals and extend the monolingual SQuAD (Rajpurkar et al., 2016) to 7 and 11 languages, respectively, thus forming cross-lingual evaluation benchmarks. Clark et al. (2020) collected an entirely new dataset (TyDi QA) of questions in 11 typologically diverse languages.

The task was to ask a question, and then the shortest span answering it from a list of paragraphs was selected. As these datasets are complementary, rather than making each other obsolete, hereby the recently released XTREME (Hu et al., 2020) benchmark combined them in a joint task. $E\chi\alpha\mu s$ differs from the aforementioned multilingual benchmarks in several aspects. First, I extend the multilingual QA efforts to a different, more challenging domain (Clark et al., 2018). Second, my datasets support more languages. Next, the questions in $E\chi\alpha\mu s$ are written by educational experts rather than non-expert annotators, making the evaluation results comparable to a top-performing student. Finally, my fine-grained evaluation for different subjects, languages, and combinations thereof allows for an in-depth analysis and comparison.

(Zero-Shot) Multilingual Models

Multilingual embeddings helped researchers to achieve new state-of-the-art results on many NLP tasks. While many pre-trained model (Grave et al., 2018; Devlin et al., 2019; Conneau and Lample, 2019) are available, the need for task-specific data in the target language still remains. Training such models is language-independent, and representations for common words remain close in the latent vector space for a single language, albeit unrelated for different languages. A possible approach to

overcome this effect is to learn an alignment function between the spaces (Artetxe and Schwenk, 2019; Joty et al., 2017). Moreover, zero-shot application of fine-tuned multilingual language models (Devlin et al., 2019; Conneau and Lample, 2019) on XNLI (Conneau et al., 2018), a corpus containing sentence pairs annotated with textual entailment and translated into 14 languages, has shown very close results to such by a language-specific model.

Zero-shot transfer and multilingual models had been a hot topic in Neural Machine Translation in the past several years. Johnson et al. (2017) introduced a simple tweak to a standard sequence-to-sequence (Seq2seq, Sutskever et al. (2014)) model by adding a special token to the encoder's input, denoting the target language, thus allowing a zero-shot learning for new language pairs. Recent work in zero-resource translation outlined different strategies for learning to translate without having a parallel corpus between the two target languages. First, a many-to-one approach was adopted by Firat et al. (2016) based on building a corpus from a single language paired with many others, allowing simultaneous training of multiple models, with a shared attention layer. A many-to-many relationship between languages was later used by Aharoni et al. (2019), in an attempt to train a single Transformer model.

Pivot-language approaches can also be used to overcome the lack of parallel corpora for the source–target language pair. Chen et al. (2017) used a student–teacher framework to train an Neural Machine Translation (NMT) model, using a third language as a pivot. A similar idea was applied to MRC by Asai et al. (2018), who translated each question to a pivot language, and then found the correct answer in the target language using soft-alignment attention scores.

2.4 Retrieving Long-Form Explanations

2.4.1 Previously Fact-Checked Claims

While fake news and mis/disinformation detection have been studied extensively (Zubiaga et al., 2016; Li et al., 2016; Zubiaga et al., 2018; Martino et al., 2020; Hardalov et al., 2022), the problem of detecting previously fact-checked claims remains under-explored. Hassan et al. (2017) mentioned the task as a component of their end-to-end fact-checking pipeline, but did not evaluate it in isolation, neither did they study its contribution.

ClaimsKG dataset (Tchechmedjiev et al., 2019) introduced a knowledge graph task, that allows for exploration of the network of claims related to a named entity or keyphrase. While the task is similar it does not give information whether the certain claim was fact-checked or not.

Recently, the task received more attention from the research community. Shaar et al. (2020) collected two datasets, from PolitiFact (political debates) and from

Snopes (tweets), of claim and corresponding fact-checking articles. The CLEF *CheckThat!* lab (Barrón-Cedeno et al., 2020; Shaar et al., 2021) extended these datasets with additional data in English and Arabic. The best-performing systems (Pritzkau, 2021; Mihaylova et al., 2021; Chernyavskiy et al., 2022) used a combination of BM25 retrieval, semantic similarity using sentence embeddings (Reimers and Gurevych, 2019), and reranking. Bouziane et al. (2020) further used external data from fact-checking datasets (Wang, 2017; Thorne et al., 2018; Wadden et al., 2020).

Detecting previously fact-checked claims also raised some attention in the industry. It was introduced as an integral part of Google's Fact Check Explorer.[1] The tool finds results from several well-known fact-checking websites using a standard search functionality.

My work is most similar to that of Vo and Lee (2020), who mined 19K tweets and corresponding fact-checked articles. Unlike them, I focus on textual claims (they were interested in multimodal tweets with images), I collected an order of magnitude more examples, and I proposed a novel approach to learn from such noisy data directly (while they manually checked each example).

2.4.2 Training with Noisy Data

Leveraging large collections of unlabeled data has been at the core of large-scale language models, such as GPT (Radford et al., 2018, 2019), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019). Recently, such language models used noisy retrieved data (Lewis et al., 2020; Guu et al., 2020) or active relabeling and data augmentation (Thakur et al., 2021). Moreover, using distantly supervised data labeling is a crucial part of the recent breakthroughs in few-shot learning (Schick and Schütze, 2021,).

Yet, there has been little work on using noisy data for fact-checking tasks. Vo and Lee (2019) collected tweets containing a link to a fact-checking website, based on which they tried to learn a fact-checking language and to generate automatic answers. You et al. (2019) used similar data from tweets for fact-checking URL recommendations.

Unlike the above work, here I propose an automatic procedure for labeling and self-training specifically designed for the task of detecting previously fact-checked claims.

2.5 Generation Models for Dialogue

The emergence of large conversational corpora such as the Ubuntu Dialog corpus (Lowe et al., 2015), OpenSubtitles (Lison and Tiedemann, 2016), CoQA (Reddy

[1] toolbox.google.com/factcheck/explorer

et al., 2019) and the Microsoft Research Social Media Conversation Corpus (Sordoni et al., 2015)[2] has enabled the use of generative models and end-to-end neural networks in the domain of conversational agents. In particular, sequence-to-sequence models, which were initially proposed for machine translation (Luong et al., 2015; Sutskever et al., 2014; Bahdanau et al., 2015), got adapted to become a standard tool for training end-to-end dialogue systems. Early vanilla Seq2seq models were adopted by Vinyals and Le (2015), who experimented with two datasets: IT helpdesk tickets and Open Subtitles. They further pointed out to the following issues: lack of context modeling for multi-turn dialogs, lack of "personality" for models trained on different sources, and the need for human evaluation of the generated responses. The models got quickly extended to model hierarchical structure (Serban et al., 2016), context (Sordoni et al., 2015), and combination thereof (Sordoni et al., 2015). While models are typically trained on corpora such as Ubuntu, some work (Boyanov et al., 2017) has also used data from Community Question Answering forums (Nakov et al., 2017); this means forming a training pair involving a question and each good answer in the corresponding question-answer thread.

Twitter data is particularly suitable for fitting neural conversational models because of the length restriction, which encourages people to write short, more precise tweets. Thus, it was used in a number of studies. Serban et al. (2016) improved Seq2seq models using a hierarchical structure. Sordoni et al. (2015) worked on modeling the context. Shang et al. (2015) proposed a neural network response generator for short-text conversation, which was trained with a large number of one-round conversations from a micro-blogging service, and could generate grammatically correct and content-wise appropriate responses.

Some interesting approaches for building customer support chatbots were shown in (Cui et al., 2017; Qiu et al., 2017), as a combination of retrieval and neural models. Cui et al. (2017) used information from in-page product descriptions, as well as user-generated content from e-commerce web sites to improve online shopping experience. Their approach incorporated four different components (a fact database, FAQs, opinion-oriented answers, and a neural-based chit-chat generator) into a meta-engine that makes a choice between them. Qiu et al. (2017) proposed an open-domain chatbot engine that integrates results from IR and Seq2seq models, and uses an attentive Seq2seq reranker to choose dynamically between their outputs.

In the domain of customer support, it has been shown that generative models such as Seq2seq and the Transformer perform better then retrieval-based models, but they fail in the case of insufficient training data (Hardalov et al., 2018). Other work has incorporated intent categories and semantic matching into an answer selection model, which uses a knowledge base as its source (Li et al., 2018). In the insurance domain, Feng et al. (2015) proposed a generic deep learning approach

[2] http://research.microsoft.com/convo/

for answer selection, based on Convolutional Neural Network (CNN). In Li et al. (2015) combined Recurrent Neural Network based on Long Short-Term Memory (LSTM) cells and reinforcement learning to learn without the need of prior domain knowledge.

More recently, the Transformer, a model without recurrent connections, was proposed (Vaswani et al., 2017), demonstrating state-of-the-art results for Machine Translation in various experimental scenarios for several language pairs and translation directions, thus emerging as a strong alternative to Seq2seq methods. The fact that it only uses self-attention makes it a lot faster both at training and at inference time, even though its deep architecture requires more calculations than a Seq2seq model, as it enables high degree of parallelism, while maintaining the ability to model word sequences through the mechanism of attention and positional embeddings.

The raise of large pre-trained Transformers has enabled models to generate more concise and coherent sequences. They showed state-of-the-art performance on many benchmarks and tasks, including sequence generation ones. Their success is rooted, on one hand, in the extensive pre-training on a huge amount of diverse textual snippets and on the other, in their scale in terms of learnable parameters. There exist many different approaches for both training the models, and for adapting their architecture. In contrast to encoder-only transformer models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Lan et al., 2020), GPT* models (Radford et al., 2018, 2019; Brown et al., 2020) adopt a decoder-only architecture trained on a generic language modeling objective that allows for word by word sequence generation. Zhang et al. (2020) further specialized the model on conversational data from Reddit. Nevertheless, encoder-decoder transformers trained using text-to-text transfer Raffel et al. (2020) and denoising pre-training Lewis et al. (2020) had shown better performance compared to both encoder- and decoder-only architectures. Moreover, the models can generate sequences, that differ from their inputs, although they are no longer trained as language models. Moreover, based on their success on a single language, these models have been trained using multilingual corpora Liu et al. (2020); Xue et al. (2021). However, an important limitation are the maximum input and output lengths, which remained limited between up to 512 tokens and a couple of thousand tokens. This is often enough for encoding single-turn dialogues, but modern conversational agents must be able to navigate through multi-turn conversation, and thus they face problems when trying to understanding the whole dialogue history, to generate responses based on long document, etc. A recently introduced Transformer variant, namely the Longformer Beltagy et al. (2020), offers a mechanism to ease these limits and increase the maximum sequence lengths 8 times, up to 4,096 tokens, compared to vanilla BERTs Devlin et al. (2019). These models start to find applications in end-to-end chatbot Boyd et al. (2020); Parthasarathi et al. (2021); Su et al. (2022).

2.6 Multi-Source Response Selection

Response selection has been recognized as an important research direction in the domain of customer support chatbots. Ritter et al. (2011) presented a data-driven approach to generating conversational responses to social media posts, based on phrase-based Statistical Machine Translation (SMT) that is conditioned on the dialogue so far. The proposed approach showed better performance compared to classical information retrieval approaches. Ouchi and Tsuboi (2016) worked on addressee and response selection for multi-party conversation. More specifically, they proposed two modeling procedures (static and dynamic) to jointly model the speakers and their utterances in context. Song et al. (2016) proposed a post-ranking procedure that combines utterances from a generative and IR models. In order to obtain new utterance candidates, the model first retrieves candidates from an inverted index, then it re-writes them using a Seq2seq model in order to contextualize the model its input is conditioned on the query as well. Qiu et al. (2017) used an attentive Seq2seq re-ranker to choose dynamically between the outputs of a retrieval-based and a Seq2seq model. Similarly, Cui et al. (2017) combined a fact database, FAQs, opinion-oriented answers, and a neural-based chit-chat generator, by training a meta-engine that chooses between them.

Answer combination is also a key research topic in the related field of information retrieval. For example, Pang et al. (2017) proposed a generic relevance ranker based on deep learning and CNNs, which tries to maintain standard IR search engine characteristics, such as exact matching and query term importance, while enriching the results based on semantics, proximity heuristics, and diversification.

More recently, Curry et al. (2018) introduced a new ensemble architecture of multiple bots, as part of the 2018 Alexa prize challenge Ram et al. (2018). It combines multiple rule-based dialogue systems to support topic-based multi-domain conversations. In particular, the model uses a variety of ontologies and natural language understanding pipelines that extract information from a different web sources such as Reddit, although the final selection is guided by a simple priority bot list. Subramaniam et al. (2018) proposed a novel conversational framework that uses an Orchestrator Bot to understand the user query and to direct them to a domain-specific bot that handles subsequent dialogue. Clarke et al. (2022) focused on improving the capabilities of task-oriented by combining multiple black-box conversational agents. They leveraged existing personal assistants from big tech companies (i.e. Alexa, Google Assistant and Siri) and experimented with two techniques for combining their answers: (*i*) *question agent pairing*, i.e., select the most relevant agent to answer the question using metadata about the agent's capabilities, and (*ii*) *question response pairing*, where, similarly to previous work, the goal is to select the correct agent response.

The response selection task was part of the DSTC-7 (Gunasekara et al., 2019;

Kummerfeld et al., 2019) and the subsequent DSTC-8 (Kim et al., 2019) challenges. Zhou et al. (2018) proposed Deep Attention Matching (DAM), a transformer-based neural network that uses sentence self-attention and cross-attention between the context and the candidate responses to obtain representations of text segments at different granularities in order to find the best matching response for the current context. (Yuan et al., 2019) used fusing with a Multi-hop Selector Network (MSN) to select relevant context utterances and to match them with the response utterance. Tao et al. (2019) took a step further in modeling the utterance-response relation, showing that the depth of interaction affects the effectiveness of the model. (Wang et al., 2020) framed response selection as a dynamic topic tracking task to match the topic between the response and relevant conversation context. Their framework leveraged multi-task learning based on efficient encoding through large pre-trained Transformers and a self-supervised procedure to inject topical information into the models.

Finally, it is worth mentioning that another application of the response selection task is in evaluating dialogue systems (Henderson et al., 2020; Sato et al., 2020; Wang et al., 2020).

2.7 Summary

In this section, I reviewed previous work related to the topic of this thesis. First, I highlighted datasets and holistic approaches for building conversation agents. Next, I continued with the related task of question answering, both from monolingual and cross-lingual perspective. Then, I covered methods for retrieval of long-form explanations, albeit surveying only one possible direction – finding previously fact-check claims. Finally, I explored methods for advanced conversation, i.e., generative conversational models and strategies for combining multiple answers.

Chapter 3

Semantic Parsing of Human-Generated Utterances

This chapter presents a novel method for natural language understanding that models jointly the tasks of intent detection and slot filling. The motivation behind this approach is that the two tasks have a strong connection between them: first, the detected intent can narrow down the possible set of slot tags, as not all tags are applicable to all intents. Moreover, the intent can contextualize the slot tag selection: the same tokens can have different tags depending on the context and the intent, e.g., *Monday, October 12th* can refer to '*departure date*' when the intent is to *book a flight*, however it can also be '*movie release date*' when the client is looking for a movie recommendation. On the other hand, this relation is valid also in the opposite direction as well – between the predicted slots and the intent –, and the predicted slots can serve as clues for the joint model when classifying the intent.

To this end, the main idea is to use a pooling attention layer for intent classification in order to obtain a single representation for the whole sentence formed from all tokens, as their vectors representations encode information about the slots, too. Further, the slot filling task is reinforced with truecasing and word-specific features, that allow the model to distinguish between names such as personal, city, country, state, etc., in addition to the predicted intent distribution from the aforementioned layer. The method outperforms strong neural-based models on two well-known NLU datasets for slot filling and intent detection.

Finally, I present exhaustive analysis of the task-related knowledge in the pre-trained models. This knowledge helps the models to significantly outperform classical NLP models without extensive pre-training, but it often leads to overestimation of the model's performance (Bender et al., 2021; Bowman, 2022).

This chapter is mainly based on

- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2020a. Enriched Pre-trained Transformers for Joint Slot Filling and Intent Detection. *arXiv preprint arXiv:2004.14848*

3.1 Introduction

With the proliferation of portable devices, smart speakers, and the evolution of personal assistants, such as Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana, a need for better natural language understanding (NLU) has emerged. Moreover, many Web platforms and applications that interact with the users depend on the abilities of an internal NLU component, e.g., customer service with social media (Huang et al., 2021), in dialogue systems in general (Zeng et al., 2021), for web queries understanding (Tsur et al., 2016; Ye et al., 2016), and general understanding of natural language interaction (Vedula et al., 2020). The major challenges such systems face are *(i)* finding the intention behind the user's request, and *(ii)* gathering the necessary information to complete it via slot filling, while *(iii)* engaging in a dialogue with the user.

Table 3.1 shows a user request collected from a personal voice assistant. Here, the intent is to *play music* by the artist *Justin Broadrick* from year *2005*. The slot filling task naturally arises as a sequence tagging task. Conventional neural network architectures, such as RNNs or CNNs are appealing approaches to tackle this problem. Various extensions thereof can be found in previous work (Xu and Sarikaya, 2013; Goo et al., 2018; Hakkani-Tür et al., 2016; Liu and Lane, 2016; E et al., 2019; Gangadharaiah and Narayanaswamy, 2019). Moreover, sequence tagging approaches such as maximum entropy Markov model (MEMM) (Toutanvoa and Manning, 2000; McCallum et al., 2000) and conditional random fields (CRF) (Lafferty et al., 2001; Jeong and Lee, 2008; Huang et al., 2015) have been added on top to enforce better modeling of the dependencies between the posteriors for the slot filling task. Recent work has introduced other methods such as hierarchical structured capsule networks (Xia et al., 2018; Zhang et al., 2019), and graph interactive networks (Qin et al., 2020).

In this chapter, I investigate the usefulness of pre-trained models for natural language understanding. My approach is based on BERT (Devlin et al., 2019) and its successor RoBERTa (Liu et al., 2019). That model offers two main advantages over previous ones (Hakkani-Tür et al., 2016; Xu and Sarikaya, 2013; Gangadharaiah and Narayanaswamy, 2019; Liu and Lane, 2016; E et al., 2019; Goo et al., 2018): (*i*) they are based on the Transformer architecture (Vaswani et al., 2017), which allows them to use bi-directional context when encoding the tokens instead of left-to-right (as in RNNs) or limited windows (as in CNNs), and (*ii*) the model is trained on huge

Intent	PlayMusic						
Words	play	music	from	2005	by	justin	broadrick
	↓	↓	↓	↓	↓	↓	↓
Slots	O	O	O	B-year	O	B-artist	I-artist

Table 3.1: Example from the SNIPS dataset with slots encoded in the BIO format. The utterance's intent is *PlayMusic*, and the given slots are *year* and *artist*.

unlabeled text collections, which allows it to leverage relations learned during pre-training, e.g., that *Justin Broadrick* is connected to music or that *San Francisco* is a city.

I further adapt the pre-trained models for the NLU tasks. For the intent, I introduce a pooling attention layer, which uses a weighted sum of the token representations from the last language modeling layer. Moreover, I reinforce the slot representation with the predicted intent distribution, and word features such as predicted word casing, and named entities. To demonstrate its effectiveness, I evaluate it on two publicly available datasets: ATIS (Hemphill et al., 1990) and SNIPS (Coucke et al., 2018)

The contributions of this chapter can be summarized as follows:

- I enrich a pre-trained language model, such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), to jointly solve the tasks of intent classification and slot filling.
- I introduce an additional pooling network from the intent classification task, allowing the model to obtain the hidden representation from the entire sequence.
- I use the predicted user intent as an explicit guide for the slot fitting layer rather than just depending on the language model
- I reinforce the slot learning with features such as named entity and true case annotations.
- I present exhaustive analysis of the task-related knowledge in the pre-trained model, for both datasets.

3.2 Dataset

In my experiments, I use two publicly available datasets, the Airline Travel Information System (ATIS) (Hemphill et al., 1990), and SNIPS (Coucke et al., 2018). The ATIS dataset contains transcripts from audio recordings of flight information requests, while the SNIPS dataset is gathered by a custom intent engine for personal voice assistants. Albeit both are widely used in NLU benchmarks, ATIS is substantially smaller – almost three times in terms of examples, and it contains s times less words. However, it has a richer set of labels, 21 intents and 120 slot categories, as opposed to the 7 intents and 72 slots in SNIPS. Another key difference is the diversity of domains – ATIS has only utterances from the flight domain, while SNIPS covers various subjects, including entertainment, restaurant reservations, weather forecasts, etc. (see Table 3.2) Furthermore, ATIS allows multiple intent labels. As they only form about 2% of the data, I do not extend my model to multi-label

	ATIS	SNIPS
Vocab Size	722	11,241
Average Sentence Length	11.28	9.05
#Intents	21	7
#Slots	120	72
#Training Samples	4,478	13,084
#Dev Samples	500	700
#Test Samples	893	700

Table 3.2: Statistics about the ATIS and SNIPS datasets.

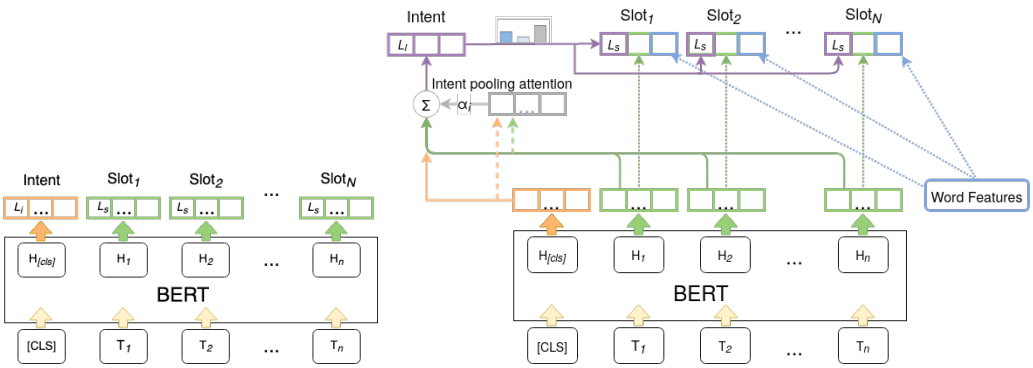
classification. Yet, I add a new intent category for combinations seen in the training dataset, e.g., utterance with intents *flight* and also *airfare*, would be marked as *airfare#flight*. A comparison between the two datasets is shown in Table 3.2.

3.3 Proposed Approach

I propose a joint approach for intent classification and slot filling built on top of a pre-trained language model, i.e., BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019). I further improve the base model in three ways: (*i*) for intent detection, I obtain a pooled representation from the last hidden states for all tokens (Section 3.3.1), (*ii*) I obtain predictions for the word case and named entities for each token (word features), and (*iii*) I feed the predicted intent distribution vector, BERT's last hidden representations, and word features into a slot filling layer (see Section 3.3.2). The complete architecture of the model is shown in Figure 3.1b.

3.3.1 Intent Pooling Attention

Traditionally, BERT and subsequent BERT-style models have used a special token ([CLS]) to denote the beginning of a sequence. In the original paper (Devlin et al., 2019), the authors attach a binary classification loss to it for predicting whether



(a) BERT-Joint. **(b)** Transformer-NLU (mine).

Figure 3.1: Model architectures for joint learning of intent and slot filling: (a) classical joint learning with BERT/RoBERTa, and (b) proposed enhanced version of the model.

two sequences follow each other in the text (next sentence prediction, or NSP). Adding such an objective forces the last residual block to pool a contextualized representation for the whole sentence from the penultimate layer, which should have a more semantic, rather than task-specific meaning. The latter strives to improve downstream sentence-level classification tasks such as entailment, semantic textual similarity, intent detection, etc. However, its effectiveness has been recently debated in the literature (Conneau and Lample, 2019; Joshi et al., 2020; Yang et al., 2019; Lan et al., 2020). It has been even argued that it should be removed (Liu et al., 2019).

Here, the task is to jointly learn the two strongly correlated tasks, i.e., intent detection and slot filling. Hereby, using the pooled representation from the [CLS] token can miss important information about the slots' tags when used as an input for predicting the users' intent. I hypothesize that using the token-level representation obtained from the last layer before the slot projection one can help the model in learning the intent detection task, as these representations contain important task-specific information.

Therefore, I introduce a pooling attention layer to better model the relationship between the task-specific representations for each token and for the intent. I further adopt a global concat attention (Luong et al., 2015) as a throughput mechanism. Namely, I learn an alignment function to predict the attention weights α_{int} for each token. I obtain the latter by multiplying the outputs from the language model $H \in \mathbb{R}^{N \times d_h}$ by a latent weight matrix $W_{int\_e} \in \mathbb{R}^{d_h \times d_h}$, where N is the number of tokens in an example and d_h is the hidden size of the Transformer. This is followed by a non-linear *tanh* activation. In order to obtain importance logit for each token, I multiply the latter by a projection vector $v \in \mathbb{R}^{d_h}$ (shown in Eq. 3.1). I further normalize and scale (Vaswani et al., 2017) to obtain the attention weights.

$$align(H) = v \cdot \tanh(W_{int\_e} \cdot H^T) \tag{3.1}$$

$$\alpha_{int} = softmax(\frac{align(H)}{\sqrt{d_h}}) \tag{3.2}$$

$$h_{int} = tanh(\sum_{i=1}^{N} \alpha_{int}^{i} h_{enc}^{i}) \tag{3.3}$$

$$y_{int} = W_{int} h_{int}^{T} + b_{int} \tag{3.4}$$

Finally, I gather a hidden representation h_{int} as a weighted sum of all attention inputs, and I pass it through a *tanh* activation (see Eq. 3.3). For the final prediction, I use a linear projection on top of h_{int}. I apply dropouts on h_{int}, and on the attention weights (Vaswani et al., 2017).

3.3.2 Slots Modeling

The task of slot filling is closely related to tasks such as part-of-speech (POS) tagging and named entity recognition (NER). Moreover, it can benefit from knowing the interesting entities in the text. Therefore, I reinforce the slot filling with tags found by a named entity recognizer (word features). Next, I combine the intent prediction, the language model's hidden representations, and some extracted word features into a single vector used for token slot attribution. Details about all components are discussed below.

Word Features A major shortcoming of having free-form text as an input is that it tends not to follow basic grammatical principles or style rules. The casing of words can also guide the models while filling the slots, i.e., upper-case words can refer to names or to abbreviations. Also, knowing the proper casing enabled the use of external NERs or other tools that depend on the text quality.

As a first step, I improve the text casing using a *TrueCase*[1] model. The model maps the words into the following classes: *UPPER, LOWER, INIT_UPPER, and O*, where *O* is for mixed-case words such as *McVey*. With the text re-cased, I further extract the named entities with a NER annotator. Named entities are recognized using a combination of three CRF sequence taggers trained on various corpora. Numerical entities are recognized using a rule-based system. Both the truecaser and the NER model are part of the Stanford CoreNLP toolkit (Manning et al., 2014).

Finally, I merge some entities ((job) title, ideology, criminal charge) into a special category *other* as they do not correlate directly to the domains of either dataset. Moreover, I add a custom regex-matching entry for *airport_code*, which are three-letter abbreviations of the airports. The latter is specially designed for the ATIS (Tur et al., 2010) dataset. While, marking the proper terms, some of the codes introduce noise, e.g., the proposition *for* could be marked as an *airport_code* because of *FOR (Aeroporto Internacional Pinto Martins, Fortaleza, CE, Brazil)*. In order to mitigate this effect, I do a lookup in a dictionary of English words, and if a match is found, I trigger the *O* class for the token.

In order to allow the network to learn better feature representations for the named entities and the casing, I pass them through a two-layer feed-forward network. The first layer is shown in Eq. 3.6 followed by a non-linear PReLU activation, where $W_w \in \mathbb{R}^{23\times32}$. The second one is a linear projection f_{words} (Eq. 3.7), where $W_{proj} \in \mathbb{R}^{32\times32}$.

$$s_w^i = W_w[ners; cases] + b_w \tag{3.5}$$

$$h_w^i = max(0, s_w^i) + \alpha * min(0, s_w^i) \tag{3.6}$$

$$f_{words}(ners, cases) = W_{proj} {h_w^i}^T + b_{proj} \tag{3.7}$$

[1] https://stanfordnlp.github.io/CoreNLP/truecase.html

Sub-word Alignment Modern NLP approaches suggest the use of sub-word units (Sennrich et al., 2016; Wu et al., 2016; Kudo and Richardson, 2018), which mitigate the effects of rare words, while preserving the efficiency of a full-word model. Although they are a flexible framework for tokenization, sub-word units require additional bookkeeping for the models in order to maintain the original alignment between words and their labels.

I first split the sentences into the original word-tag pairs, I then disassemble each one into word pieces (or BPE, in the case of RoBERTa). Next, the original slot tag is assigned to the first word piece, while each subsequent one is marked with a special tag (*X*). Still, the word features from the original token are copied to each unit. To align the predicted labels with the input tags, I keep a binary vector for the active positions.

Slot Filling as Token Classification As in Devlin et al. (2019), I treat the slot filling as token classification, and I apply a shared layer on top of each token's representations to predict the tags.

Furthermore, I assemble the feature vector for the i^{th} slot by concatenating together the predicted intent probabilities, the word features, and the contextual representation from the language model. Afterwards, I add a dropout followed by a linear projection to the proper number of slots:

$$y_s^i = W_s[softmax(y_{int}); f_{words}^i; h_{LM}^i] + b_s \tag{3.8}$$

3.3.3 Interaction and Learning

To train the model, I use a joint loss function $\mathcal{L}_{joint}$ for the intent and for the slots. For both tasks, I apply cross-entropy over a softmax activation layer, except in the case of CRF tagging. In those experiments, the slot loss $\mathcal{L}_{slot}$ will be the negative log-likelihood (NLL) loss. Moreover, I introduce a new hyper-parameter γ to balance the objectives of the two tasks (see Eq. 3.9). Finally, I propagate the loss from all the non-masked positions in the sequence, including word pieces, and special tokens ([CLS], <s>, etc.). Note that I do *not* freeze any weights during fine-tuning. More details about the model can be found in Section 3.4.4.

$$\mathcal{L}_{joint} = \gamma * \mathcal{L}_{intent} + (1 - \gamma) * \mathcal{L}_{slot} \tag{3.9}$$

3.4 Experimental Setup

In this section, I describe the evaluation measures, the baselines and the state-of-the-art models I compare to, as well as specific details about my proposed model.

3.4.1 Measures

I evaluate my models using three well-established evaluation metrics. The intent detection performance is measured in terms of accuracy. For the slot filling task, I use F1-score. Finally, the joint model is evaluated using sentence-level accuracy, i.e., proportion of examples in the corpus, whose intent and slots are both correctly predicted. Here, I must note that during evaluation I consider only the predictions for aligned words (I omit special tokens, e.g., [CLS], [SEP], <s>, </s>) and word pieces).

3.4.2 Baselines

For my baseline models, I use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which I fine-tune. In particular, I train a linear layer over the pooled representation of the special [CLS] token to predict the intent. Moreover, I add a shared layer on top of the last hidden representations of the tokens in order to obtain a slot prediction. The model's architecture is shown in Figure 3.1a.

BERT For training the model, I follow the fine-tuning procedure proposed by Devlin et al. (2019). I train a linear layer over the pooled representation of the special [CLS] token to predict the utterance's intent. The latter is optimized during pre-training using the next sentence prediction (NSP) loss to encode the whole sentence. Moreover, I add a shared layer on top of the last hidden representations of the tokens in order to obtain a slot prediction. Both objectives are optimized using a cross-entropy loss.

RoBERTa This model follows the same training procedure as BERT, but drops the NSP task during pre-training. Still, the intent loss is attached to the special start token <s>.

Chen et al. (2019) used BERT with a token classification pipeline to jointly model the slot and the intent, with an additional CRF layer on top.[2] However, they evaluated the slot filling task using per-token F1-score (micro averaging), rather than per-slot entry, as is standard, which in turn artificially inflated their results. As their results are not comparable to the rest, I do not include them in my comparisons.

[2]In terms of micro-average F1 for slot filling, Chen et al. (2019) reported 96.1 on ATIS and 96.27 on SNIPS (per-token). For comparison, for my joint model, these scores are 98.1 and 97.9 (per-token); however, the correct scores for my model are actually 95.7 and 96.3 (per-slot).

3.4.3 State-of-the-Art Models

I further compare my approach to some other benchmark models. Here, I must note that I include models that do not use embeddings from large pre-trained Transformers such as BERT in order to measure the improvements that come solely from the pre-training procedure (see Section 3.5.4):

- Joint Seq. (Hakkani-Tür et al., 2016) uses a RNN to obtain hidden states for each token in the sequence for slot filling, and uses the last state to predict the intent.
- Atten.-Based (Liu and Lane, 2016) treats the slot filling task as a generative one, applying sequence-to-sequence RNN to label the input. Further, an attention weighted sum over the encoder's hidden states is used to detect the intent.
- Slotted-Gated (Goo et al., 2018) introduces a special gated mechanism to an LSTM network, thus reinforcing the slot filling with the hidden representation used for the intent detection.
- Capsule-NLU (Zhang et al., 2019) adopts Capsule Networks to exploit the semantic hierarchy between words, slots, and intents using dynamic routing-by-agreement schema.
- Interrelated (E et al., 2019) uses a Bidirectional LSTM with attentive sub-networks for the slot and the intent modeling, and an interrelated mechanism to establish a direct connection between the two. SF (slot), and ID (intent) prefixes indicate which sub-network to execute first.
- Stack-Propagation (Qin et al., 2019) consists of a self-attentive BiLSTM encoder for the utterance and two decoders, one for the intent-detection task that performs a token-level intent detection, and one for the slot filling task.
- AGIF (Qin et al., 2019) uses Adaptive Graph-Interactive Framework to jointly model intent detection and slot filling with an intent-slot graph interaction layer applied to each token adaptively.

3.4.4 Model Details

I use the PyTorch implementation of BERT from the Transformers library of Wolf et al. (2020) as a base for my models. I fine-tune all models for 50 epochs with hyper-parameters set as follows: batch size of 64 examples, maximum sequence length of 50 word pieces, dropout set to 0.1 (for both attentions and hidden layers), and weight decay of 0.01. For optimization, I use Adam with a learning rate of 8e-05, β_1 0.9, β_2 0.999, ϵ 1e-06, and warm-up proportion of 0.1. Finally, in order to balance between the intent and the slot losses, I set the parameter γ (Eq. 3.9) to 0.6, I test the range 0.4–0.8 with 0.1 increment. All the models use the same

	ATIS			SNIPS		
Model	Intent (Acc)	Sent. (Acc)	Slot (F1)	Intent (Acc)	Sent. (Acc)	Slot (F1)
Joint Seq. (Hakkani-Tür et al., 2016)	92.60	80.70	94.30	96.90	73.20	87.30
Atten.-Based (Liu and Lane, 2016)	91.10	78.90	94.20	96.70	74.10	87.80
Sloted-Gated (Goo et al., 2018)	95.41	83.73	95.42	96.86	76.43	89.27
Capsule-NLU (Zhang et al., 2019)	95.00	83.40	95.20	97.30	80.90	91.80
Interrelated SF-First (E et al., 2019)	97.76	86.79	95.75	97.43	80.57	91.43
Interrelated ID-First (E et al., 2019)	97.09	86.90	95.80	97.29	80.43	92.23
Stack-Propagation (Qin et al., 2019)	96.9	86.5	95.9	98.0	86.9	94.2
AGIF (Qin et al., 2020)	97.1	87.2	96.0	98.1	87.3	94.8
BERT-Joint	97.42	87.57	95.74	98.71	91.57	96.27
RoBERTa-Joint	97.42	87.23	95.32	98.71	90.71	95.85
Transformer-NLU:BERT	**97.87**	**88.69**	**96.25**	**98.86**	91.86	**96.57**
Transformer-NLU:RoBERTa	97.76	87.91	95.65	**98.86**	**92.14**	96.35
Transformer-NLU:BERT w/o Slot Features	97.87	88.35	95.97	98.86	91.57	96.25
Transformer-NLU:BERT w/ CRF	97.42	88.26	96.14	98.57	92.00	96.54

Table 3.3: Intent detection and slot filling results on the SNIPS and the ATIS datasets. Highest results in each category are written in **bold**. My models are shown in *italic*; the non-italic models on top come from the literature. Qin et al. (2019, 2020) report their results with single precision.

pre-processing, post-processing, and the standard for these tasks metrics. In order to tackle the problem with random fluctuations for BERT/RoBERTa, I ran the experiments three times and I used the best-performing model on the development set. I define the latter as the highest sum from all three measures described in Section 3.4.1. All the above-mentioned hyper-parameter values were tuned on the development set, and then used for the final model on the test set. All models were trained on a single K80 GPU instance for around an hour.

3.5 Experiments and Analysis

Here, I discuss the results for my model and I compare them to the state of the art and to BERT baselines. I further present an exhaustive analysis of the model components.

3.5.1 Evaluation Results

Table 3.3 presents quantitative evaluation results in terms of (*i*) intent accuracy, (*ii*) sentence accuracy, and (*iii*) slot F1 (see Section 3.4.1). The first part of the tables refers to previous work, whereas the second part presents my experiments and is separated with a double horizontal line. The evaluation results confirm that my model performs better then the current state-of-the-art.

Metric	Relative Error Reduction	
	ATIS	
Intent (Acc)	4.91%	17.44%
Sent. (Acc)	11.64%	11.43%
Slot (F1)	6.25%	19.87%
	SNIPS	
Intent (Acc)	40.00%	11.63 %
Sent. (Acc)	35.91%	6.76%
Slot (F1)	37.64%	17.35%
Transformer-NLU	vs. SOTA	vs. BERT

Table 3.4: Comparing *Transformer-NLU:BERT* to the two baselines: *(i)* current SOTA for each measure, and *(ii)* conventionally fine-tuned BERT-Joint without the improvements, in terms of relative error reduction (Eq. 3.10).

While, models become more accurate, the absolute difference between two experiments becomes smaller and smaller, thus a better measurement is needed. Hereby, I introduce a fine-grained measure, i.e., *relative error reduction* (RER) percentage, which is defined as the proportion of absolute error reduced by a $model_a$ compared to $model_b$.

$$RER = 1 - \frac{Error_{model_a}}{Error_{model_b}} \tag{3.10}$$

Table 3.4 shows the error reduction by my model compared to the current SOTA, and to a BERT-based baselines (see Section 3.4.2). Since there is no single best model from the SOTA, I take the per-column maximum among all, albeit they are not recorded in a single run. For the ATIS dataset, we see a reduction of 11.64% (1.49 points absolute) for sentence accuracy, and 6.25% (0.25 points absolute) for slot F1, but just 4.91% for intent accuracy (see Table 3.3). Such a small improvement can be due to the quality of the dataset and to its size. For the SNIPS dataset, we see major increase in all measures and over 35% error reduction. In absolute terms, I have 0.76 for intent, 4.84 for sentence, and 1.77 for slots (see Table 3.3). This effects cannot be only attributed to the better model (discussed in the analysis below), but also to the implicit information that BERT learned during its extensive pre-training. This is especially useful in the case of SNIPS, where fair amount of the slots in categories like *SearchCreativeWork, SearchScreeningEvent, AddToPlaylist, PlayMusic* are names of movies, songs, artists, etc.

In addition to the aforementioned results, I also report the Transformer-NLU:BERT's (and BERT's) μ and σ ATIS – Intent 98.0 ± 0.17 (BERT 97.13 ± 0.26), Sentence 88.6 ± 0.23 (BERT 87.8 ± 0), Slot 96.3 ± 0.06 (BERT 96.0 ± 0.14); SNIPS – Intent 98.6 ± 0.14 (BERT 98.42 ± 0), Sentence 92.0 ± 0.17 (BERT 91.8 ± 0.19), Slot 96.2 ± 0.05 (BERT 96.1 ± 0.06). The aforementioned results show that the mean scores of the models in the slot filling task are close, but the variance in Transformer-NLU is lower. Further, I must note that these values are calculated over the best runs from each model re-training, and they are not achieved in a single run.

3.5.2 Transformer-NLU Analysis

I dissect the proposed model by adding or removing prominent components to outline their contributions. The results are shown in the second part of Table 3.3. First, I compare the results of *BERT-Joint* and the enriched model *Transformer-NLU:BERT*. we can see a notable reduction of the intent classification error by 17.44% and 11.63% for the ATIS and the SNIPS dataset, respectively. Furthermore, we see a 19.87% (ATIS) and 17.35% (SNIPS) error reduction in slot's F1, and 11.43% (ATIS) and 11.63% (SNIPS) for sentence accuracy. I also try RoBERTa as a backbone to my model: while I still see the positive effect of the proposed architecture, the overall results are slightly worse. I attribute this to the different set of pre-training data (CommonCrawl vs. Wikipedia). I further focus my analysis on BERT-based models, since they performed better than RoBERTa-based ones.

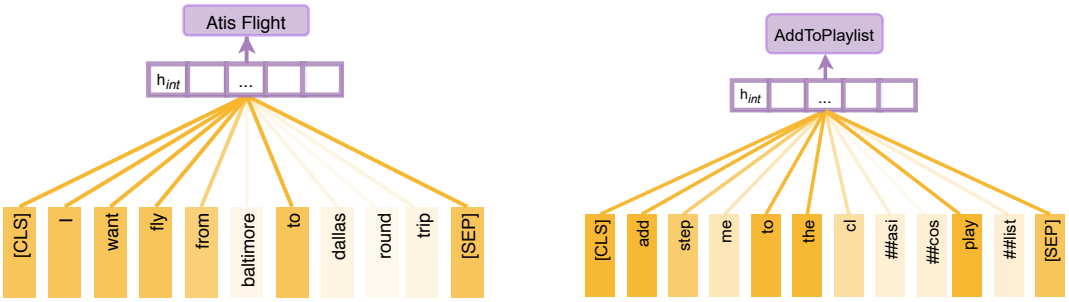
Next, I remove the additional slot features – predicted intent, word casing, and named entities. The results are shown as Transformer-NLU:BERT w/o Slot Features. As expected, the intent accuracy remains unchanged for both datasets, since I retain the pooling attention layer, while the F1-score for the slots decreases. For SNIPS, the model achieved the same score as for *BERT-Joint*, while for ATIS it was within 0.2 points absolute.

Finally, I added a CRF layer on top of the slot network, since it had shown positive effects in earlier studies (Xu and Sarikaya, 2013; Huang et al., 2015; Liu and Lane, 2016; E et al., 2019). I denote the experiment as *Transformer-NLU:BERT w/ CRF*. However, in my case it did not yield the expected improvement. The results for slot filling are close to the highest recorded, while a drastic drop in intent detection accuracy is observed, i.e., -17.44% for ATIS, and -20.28% for SNIPS. I attribute this degradation to the large gradients from the NLL loss. The effect is even stronger in the case of smaller datasets, making the optimization unstable for parameter-rich models such as BERT. I tried to mitigate this issue by increasing the γ hyper-parameter, effectively reducing the contribution of the slot's loss $\mathcal{L}_{slot}$ to the total, which in turn harmed the slot's F1. Moreover, the model does swap interchangeable slots, rather than the *B-* and *I-* prefixes, or slots unrelated to the intent (see the **Error Analysis** below).

3.5.3 Intent Pooling Attention Visualization

Next, I visualize the learned attention weights on Figure 3.2a. It presents a request from the ATIS dataset: *i want fly from baltimore to dallas round trip*. The utterance's intent is marked as *atis_flight*, and we can see that the attention successfully picked the key tokens, i.e., *I*, *want*, *fly*, *from*, and *to*, whereas supplementary words such as names, locations, dates, etc. have less contribution. Moreover, when trained on the ATIS dataset, the layer tends to set the weights in the two extremes — equally high

for important tokens, and towards zero for the rest. I attribute this to the limited domain and vocabulary.



(a) atis_flight (ATIS). **(b)** AddToPlaylist (SNIPS).

Figure 3.2: Intent pooling attention weight for one example per dataset. The thicker the line, the higher the attention weight.

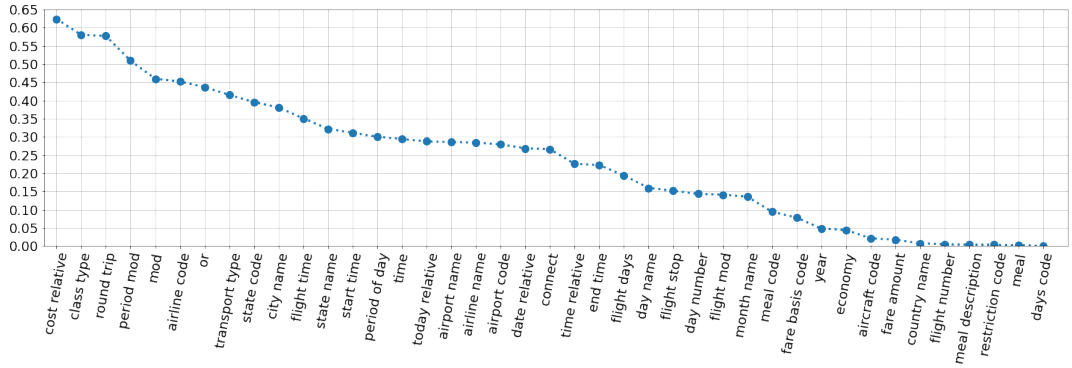
Another example, from the SNIPS dataset, is shown on Figure 3.2b. Here, the intent is to add a song to a playlist (*AddToPlaylist*). In this example, we see a more diverse spread of attention weights. The model again assigns the highest weight to the most relevant tokens *add*, *to*, *the*, and *play*. Also, the model learned that the first wordpiece has the highest contribution, while the subsequent ones are supplementary.

Finally, I let the pooling attention layer consider the special tokens marking the start and the end ([CLS], and [SEP]) of a sequence, since they are expected to learn semantic sentence-level representations from the penultimate layer. The model assigns high attention weights to both.

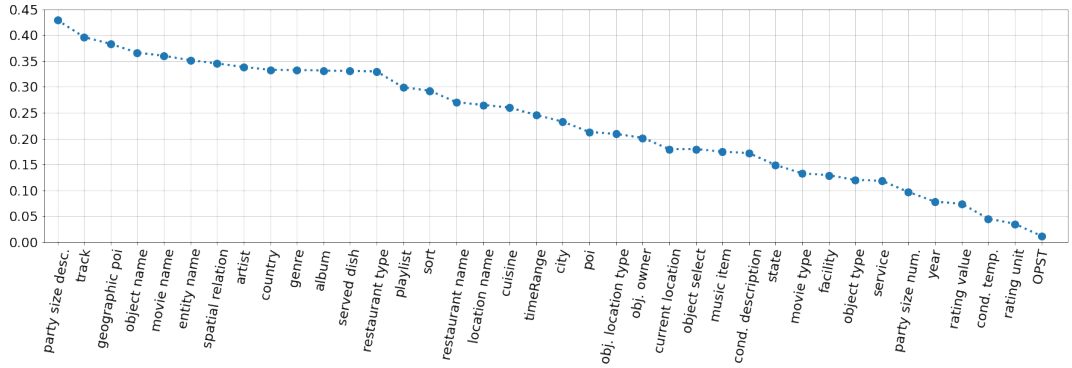
3.5.4 BERT Knowledge Analysis

As I start to understand better BERT-based large-scale pre-trained transformer models (Petroni et al., 2019; Rogers et al., 2020), I also start to observe some interesting phenomena. BERT is trained on Wikipedia articles, which allows it to learn implicit information about the world in addition to learning knowledge about language itself. Here, I evaluate how that former type of knowledge reflects on the two NLU evaluation datasets. As a first step, I extract all the slot phrases from the training sets, i.e., all the words in the slot sequence. Next, I send the latter as a query to Wikipedia[3] and I collect the article titles. Then, I try to match the phrase with an extracted title. In order to reduce the false negatives, I normalize both texts (strip punctuation, replace digits with zeros, lower-case), allow difference of one character between the two, and finally if the title starts with the phrase, I count it as a match (e.g., *Tampa* vs. *Tampa, Florida*). Overall, 66% of the slots in ATIS and 69% in SNIPS matched a Wikipage title.

[3] http://pypi.org/project/wikipedia/



(a) ATIS



(b) SNIPS *OPST stands for *object part of series.* type.

Figure 3.3: Per-class mean reciprocal rank (MRR) for the two datasets used in my study.

Next, I evaluate how much of that information is stored in the model by leveraging the standard masking mechanism used during pre-training. In particular, I split each slot in subwords, and then I replace them one by one sequentially with the special [MASK] token. I then sort the predictions for that position by probability and I take the rank of the true word. Finally, I calculate the mean reciprocal rank (MRR) over all the aforementioned ranks: 0.46 for ATIS, and 0.36 for SNIPS. I must note that the BERT's dictionary contains 32K pieces, and the expected uniform MRR is $\sim$1/16,000. Below, I present two examples to illustrate both high- and low-ranked predictions.

High ranked: *play the album jack takes the floor by tom le [MASK] on netflix*, here the model's top predictions are: [**##hrer**, *##rner, ##mmon, ##hr, ##rman*], and the correct token is ranked with the highest probability.

Low ranked: *play some hong jun [MASK]*, here the model's top guesses are mostly punctuation, and general words such as [*to, ;, ##s, and*]. The correct token *##yang* is at position 3,036, which indicates that this term is challenging.

In SNIPS (see Figure 3.3b), we can see that types such as *track, movie_name, entry_name, artist, album* have very high MRR (0.33–0.40), and ones that require numerical value, or are not part of well-known named entities suchf as object_part_of_series_type (OPST) are the lowest (under 0.1). The same in ATIS (see

Figure 3.3a) for country_name (8e-3), restriction_code (4e-3), meal (4e-3), in contrast to airline_code (0.45), transport_type (0.42), etc. However, ATIS in general does not require such task-specific knowledge, and its MRR is way higher in general, which is reflected by the overall improvement compared to the baseline models.

3.5.5 Error Analysis

Here, I discuss what errors the proposed architecture solves compared to the BERT model, and what types of errors are left unsolved. First, I compare the performance of my method (*Transformer-NLU*) to *BERT-Joint (BERT)*. In the intent detection task, the largest improvement (over BERT) comes from examples with slots, indicative for a given intent. This suggests that the model successfully uses the slot information gathered by the pooling attention layer. For the following groups, this is most prominent: (*i*) examples with multi-label intents, e.g., *atis_airline#atis_flight_no* – *"i need **flight numbers** and **airlines** ..."*, where *BERT* predicted *atis_flight_no*; (*ii*) examples containing distinctive words for another intent class – *"Give me **meal** flights ..."*, *atis_flight → meal (BERT)*, *"I need a **table** ...when it is chiller"*, *GetWeather → BookRestaurant (BERT)*. For all the aforementioned examples, both models filled the slots correctly, but only *Transformer-NLU* captured the correct intent. Moreover, we see a positive effect in detecting *SearchCreativeWork* and *SearchScreeningEvent*, while BERT tends to wrongly fill the slots, and thus swaps the two intents, e.g., *"find **heat wave**"*, or *"find **now and forever**"*. Finally, we see an additional improvement for *AddToPlaylist* and *atis_ground_fare*.

Next, I compare the performance of the two models on the slot filling task. As expected, the newly proposed model performs better, when the curated features capture some interesting phenomena. I observe that, when filling code slots (**airport, meal, airfare**) – *"what does ... code **bh** mean"*, artists, albums, movies, object names – ***dwele, nyoil, turk*** (*artist → entity_name (BERT)*), locations – *"... between milwaukee and **indiana**"*, *state → city (BERT)*; BERT confuses ***mango** (city)* with the fruit (cuisine); *"new york city **area**" O → city (BERT)* and time-related ones – ***afternoon, late night, a.m.***.

Finally, I discuss the errors of *Transformer-NLU* in general. For the ATIS dataset, 50% of the wrong intents come from multi-label cases (35% with two labels, and 15% with three), 31% *atis_flight* – *"how many **flights** does .../**have to/leave** ..."* (→ atis_quantity), 11% *atis_city* – *list la* (→ atis_abbreviation), and the others are mistakes in *atis_aircraft*. For the slots, 50% of the errors come from tags that can have a *fromloc/toloc* prefix, e.g., *city, airport_code, airport_name, etc.*, another 20% are time-related (*arrive_date, return_date*), and filled slots without tag 7%. The errors by the model for the SNIPS datasets are as follows: mislabeled intents *PlayMusic* 11%, *SearchCreativeWork* 22%, *SearchScreeningEvent* 67%, slots – *movie_name* 19%, *object_name* 16%, *playlist* 9%, track 9%, entity_name 5%, *album* 4%, *timeRange* 4%, *served_dish* 2%, filled slots without tag 19%. The model misses 9% (ATIS) and 17%

(SNIPS) of all the slots that should be filled. This is expected since SNIPS' slots have a larger dictionary (11K words), with a large proportion of the slots being names, and often including prepositions, e.g., *"... trailer of **the multiversity**"*.

3.6 Summary

In this chapter, I studied the two main tasks in task-oriented conversational natural language understanding, i.e., intent detection and slot filling. They form an important part (component) of customer service chatbots, serving user requests on the company's website or on different corporate Web and Social Media platforms. That component is responsible for extracting slot–value pairs that are later used by the *dialogue manager* to navigate the agent's next actions.

In particular, I proposed an enriched pre-trained language model to jointly model the two tasks (i.e., intent detection and slot filling), namely, *Transformer-NLU*. I designed a pooling attention layer in order to obtain intent representation beyond just the pooled one from the special start token. Further, I reinforced the slot filling with word-specific features, and the predicted intent distribution. My experiments on two standard datasets showed that Transformer-NLU outperforms other alternatives for all standard measures used to evaluate NLU tasks. I found that using RoBERTa and adding a CRF layer on top of the slot filling network did not help. Finally, the Transformer-NLU:BERT achieved intent accuracy of 97.87 (ATIS) and 98.86 (SNIPS). Or as a relative error reduction – almost 5% for ATIS, and over 40% for SNIPS, compared to the state-of-the-art. In terms of slot's filling F1, my models scored 96.25 (+6.25%) for ATIS, and 96.57 (+37.64%) for SNIPS.

Chapter 4

Curating Answers from External Knowledge Sources

This chapter discusses different approaches for curating answers from external knowledge sources. Here, the focus is on methods that rely on retrieval of contextual information, passages, entire documents, etc. in order to obtain an answer to a user-generated question (or a query).

In Section 4.2, I explore the problem of selecting the most relevant answer from a list of candidates, i.e., multiple-choice question answering. In order to choose the best option, the pipeline should be based on a two-step approach. First, retrieval of contextual passages using the question in combination with each of the candidates as a query, and then predict the most probable option based on the retrieved evidence text. However, rarely the answer to the question is contained directly in the passages, and therefore the models must derive it by reasoning beyond simple word matching.

Nevertheless, a single utterance is not always sufficient to answer the customer's question, especially if they need a step-by-step guide to complete their goal. In Section 4.3, I propose a novel methodology for retrieving previously written documents/articles related to claims made in conversations in Twitter. More precisely, in the domain of conversational agents this can be viewed as redesigning the output that a chatbot produces which is commonly a short sentence, into a long-form answer that can also serve as an explanation of a process or step-by-step guide. More precisely, in this chapter, I formulate the problem as follows: the produced answers are expected to be retrieved fact-checking articles, and the task can be defined as *finding previous fact-checked claims*. The three main challenges explored related to the aforementioned problem in this chapter are: (*i*) data scarcity, as the existing datasets are small in size, less then couple of thousand examples total, (*ii*) finding negative examples, as only correct article–claim pairs are available, and therefore there are no explicit samples from the *negative* class, and (*iii*) learning from noisy (labeled with distant supervision) examples.

This chapter is mainly based on:

- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019a. Beyond English-Only Reading Comprehension: Experiments in Zero-shot Multilingual Transfer for Bulgarian. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '19, pages 447–459, Varna, Bulgaria
- Momchil Hardalov, Anton Chernyavskiy, Ivan Koychev, Dmitry Ilvovsky, and Preslav Nakov. 2022b. CrowdChecked: Detecting Previously Fact-Checked Claims in Social Media. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, AACL-IJCNLP '22, Online

4.1 Introduction

Conversational agents often relay on external sources to answer a user's query. Based on the complexity and the intent of the ask, personal assistants such as Apple's Siri, Amazon's Alexa, Google's Assistant, among others can provide different forms of output based on the information extracted from external sources such as knowledge bases, internal APIs or other services. These answers can be categorized in three groups, based on their lengths: (*i*) short-form – weather forecast, directions from maps, the data from a smart home device, etc., (*ii*) long-form – tutorials, step-by-step guides, passages from a product manual, etc., and (*ii*) a list of search results from the Web, if the question is out of their scope (Chen et al., 2017; Karpukhin et al., 2020). Here, I focus on the former two types, i.e., short- and long-form answers, and leave the general search results for future work.

Regardless of the form of the answer it is not always trivial to find and extract relevant information. Moreover, the performance of the models is higher on in-domain data (Gururangan et al., 2020; Poth et al., 2021), e.g., better represented topics or the examples in the source language of the training data (Conneau and Lample, 2019; Conneau et al., 2020), and drops significantly when the data is out-of-domain (Elsahar and Gallé, 2019). In this chapter, I study the effectiveness of zero-shot transfer from resource-rich language to low-resource one for the task of multiple-choice question answer, i.e., selecting the correct answer from a list of possible candidates, based on retrieved evidence passages. However, it is important to note that the performance of the models is not only limited to their reasoning abilities but it is also highly depend on the quality of the retrieved evidences. This is another research question I explore in the following sections. In particular, I study the the limitations of classical IR models to find sufficient textual passages.

On the other hand, a lot of the questions that users ask, have already been answers in previous conversations (Lewis et al., 2021), and thus it is unnecessary for the dialogue agent to gather the same information again, in order to generate its next turn. This can be especially critical when the expected response is

aggregated from multiple knowledge sources or is a long-form answers, i.e., documents or guides, beyond a few sentences of length. These latter might also be viewed as explanations, as they contain extensive amount of information and usually describe some procedure in order to accomplish a task or argumentation about the question (Kwiatkowski et al., 2019; Barrón-Cedeno et al., 2020; Shaar et al., 2021). Moreover, with recent the development of large pre-trained Transformers and their application for semantic search (Reimers and Gurevych, 2019; Gao et al., 2021; Chuang et al., 2022) enables for better discovery of similar questions and retrieval of previously written answers. In this chapter, I explore the abilities of these models on the task of detecting previously fact-checked claims. In particular, I propose a novel framework based on self-adaptive learning and distant supervision to learn a ranking model that has a very high average precision of matching claims with their fact-checking long-form documents (explanations). Furthermore, as currently most of the publicly available datasets are quite scarce, I propose an entirely unsupervised procedure to collect and label claim–article pairs, in order to improve the model's performance in a low-resource regime.

The contributions of this chapter are as follows:

- Knowledge retrieval and transfer:
  - I introduce a new dataset for reading comprehension in a low-resource language such as Bulgarian. The dataset contains a total of 2,636 multiple-choice questions without contexts from matriculation exams and online quizzes. These questions cover a large variety of science topics in biology, philosophy, geography, and history.
  - I study the effectiveness of zero-shot transfer from English to Bulgarian for the task of multiple-choice reading comprehension, using Multilingual and Slavic BERT (Devlin et al., 2019), fine-tuned on large corpora, such as RACE (Lai et al., 2017).
  - I design a general-purpose pipeline[1] for extracting relevant contexts from an external corpus of unstructured documents using information retrieval.
- Finding previously written long-form answers:
  - I mine a large-scale collection of 330,000 tweets paired with fact-checking articles;
  - I propose two distant supervision strategies to label the dataset;
  - I propose a novel approach to learn from this data using a modified self-adaptive training;

[1]The dataset and the source code are available at http://github.com/mhardalov/bg-reason-BERT

– I demonstrate sizable improvements over the state of the art on a standard test set from the CLEF-CheckThat!21 competition (Shaar et al., 2021).

4.2 Knowledge Retrieval

The ability to answer questions is natural to humans, independently of their native language, and, once learned, it can be easily transferred to another language. After understanding the question, we typically depend on our background knowledge, and on relevant information from external sources.

Machines do not have the reasoning ability of humans, but they are still able to learn concepts. The growing interest in teaching machines to answer questions posed in natural language has led to the introduction of various new datasets for different tasks such as reading comprehension, both extractive, e.g., span-based (Nguyen et al., 2016; Trischler et al., 2017; Joshi et al., 2017; Rajpurkar et al., 2018; Reddy et al., 2019), and non-extractive, e.g., multiple-choice questions (Richardson et al., 2013; Lai et al., 2017; Clark et al., 2018; Mihaylov et al., 2018; Sun et al., 2019). Recent advances in neural network architectures, especially the raise of the Transformer (Vaswani et al., 2017), and better contextualization of language models (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018; Grave et al., 2018; Howard and Ruder, 2018; Radford et al., 2019; Yang et al., 2019; Dai et al., 2019) offered new opportunities to advance the field.

Here, I investigate skill transfer from a high-resource language, i.e., English, to a low-resource one, i.e., Bulgarian, for the task of multiple-choice reading comprehension. Most previous work (Pan et al., 2019; Radford et al., 2018; Tay et al., 2018; Sun et al., 2019) was monolingual, and a relevant context for each question was available a priori. I take the task a step further by exploring the capability of a neural comprehension model in a multilingual setting using external commonsense knowledge. My approach is based on the multilingual cased BERT (Devlin et al., 2019) fine-tuned on the RACE dataset (Lai et al., 2017), which contains over 87,000 English multiple-choice school-level science questions. For evaluation, I build a novel dataset for Bulgarian. I further experiment with pre-training the model over stratified Slavic corpora in Bulgarian, Czech, and Polish Wikipedia articles, and Russian news, as well as with various document retrieval strategies. Finally, I address the resource scarceness in low-resource languages and the absence of question contexts in my dataset by extracting relevant passages from Wikipedia articles.

4.2.1 Model

The model has three components: (*i*) a context retrieval module, which tries to find good explanatory passages for each question-answer pair, from a corpus of non-English documents, as described in Section 4.2.1, (*ii*) a multiple-choice reading

comprehension module pre-trained on English data and then applied to the target language in a zero-shot fashion, i.e., without further training or additional fine-tuning, to a target (non-English) language, as described in Section 4.2.1, and (*iii*) a voting mechanism, described in Section 4.2.1, which combines multiple passages from (*i*) and their scores from (*ii*) in order to obtain a single (most probable) answer for the target question.

Context Retriever

Most public datasets for reading comprehension (Richardson et al., 2013; Lai et al., 2017; Sun et al., 2019; Rajpurkar et al., 2018; Reddy et al., 2019; Mihaylov et al., 2018) contain not only questions with possible answers, but also an evidence passage for each question. This limits the task to question answering over a piece of text, while an open-domain scenario is much more challenging and much more realistic. Moreover, a context in which the answer can be found is not easy to retrieve, sometimes even for a domain expert. Finally, data scarceness in low-resource languages poses further challenges for finding resources and annotators.

In order to enable search for appropriate passages for non-English questions, I created an inverted index from Wikipedia articles using Elasticsearch.[2] I used the original dumps for the entire Wikipage,[3] and I preprocessed the data leaving only plain textual content, e.g., removing links, HTML tags, tables, etc. Moreover, I split the article's body using two strategies: a sliding window and a paragraph-based approach. Each text piece with its corresponding article title was processed by applying word-based tokenization, lowercasing, stop-words removal, stemming (Nakov, 2003; Savoy, 2007), and n-gram extraction. Finally, the matching between a question and a passage was done using cosine similarity and BM25 (Robertson and Zaragoza, 2009).

BERT for Multiple-Choice RC

The recently-proposed BERT (Devlin et al., 2019) framework is applicable to a vast number of NLP tasks. A shared characteristic between all of them is the form of the input sequences: a single sentence or a pair of sentences separated by the [SEP] special token, and a classification token ([CLS]) added at the beginning of each example. In contrast, the input for multiple-choice reading comprehension questions is assembled by three sentence pieces, i.e., context passage, question, and possible answer(s). The model follows a simple strategy of concatenating the option (candidate answer) at the end of a question. Following the notation of Devlin et al. (2019), the input sequence can be written as follows:

[CLS] Passage [SEP] Question + Option [SEP]

[2] http://www.elastic.co/
[3] http://dumps.wikimedia.org/

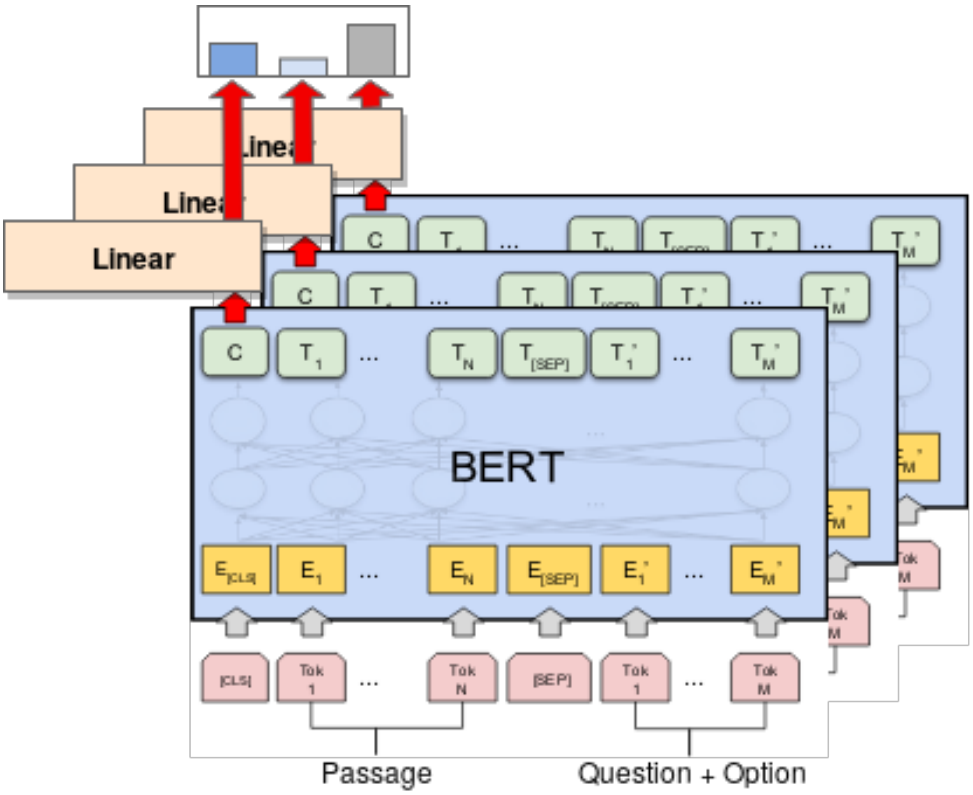


Figure 4.1: BERT for multiple-choice reasoning.

As recommended by Devlin et al. (2019), I introduce a new task-specific parameter vector L, $L \in \mathbb{R}^H$, where H is the hidden size of the model. In order to obtain a score for each passage-question-answer triplet, I take the dot product between L and the final hidden vector for the classification token ([CLS]), thus ending up with N unbounded numbers: one for each option. Finally, I normalize the scores by adding a softmax layer, as shown in Figure 4.1. During fine-tuning, I optimize the model's parameters by maximizing the log-probability of the correct answer.

Answer Selection Strategies

Finding evidence passages that contain information about the correct answer is crucial for reading comprehension systems. The context retriever may be extremely sensitive to the formulation of a question. The latter can be very general, or can contain insignificant rare words, which can bias the search. Thus, instead of using only the first-hit document, we should also evaluate lower-ranked ones. Moreover, knowing the answer candidates can enrich the search query, resulting in improved, more answer-oriented passages. This approach leaves us with a set of contexts that need to be evaluated by the MRC model in order to choose a single correct answer. Prior work suggests several different strategies: Chen et al. (2017) used the raw predicted probability from a recurrent neural network (RNN), Yang et al. (2019) tuned a hyper-parameter to balance between the retriever score and the reading

model's output, while Pan et al. (2019) and Ni et al. (2019) concatenated the results from sentence-based retrieval into a single contextual passage.

In my experiments below, I adopt a simple summing strategy. I evaluate each result from the context retriever against the question and the possible options (see Section 4.2.1 for more details), thus obtaining a list of raw probabilities. I found empirically that explanatory contexts assign higher probability to the related answer, while general or uninformative passages lead to stratification of the probability distribution over the answer options. I formulate this as follows:

$$Pr(a_j|p;q) = \frac{exp(BERT(p, q + a_j))}{\sum_{j'} exp(BERT(p, q + a_{j'}))}, \tag{4.1}$$

where p is a passage, q is a question, A is the set of answer candidates, and $a_j \in A$.

I select the final answer as follows:

$$Ans = \arg\max_{a \in A} \sum_{p \in P} Pr(A|p;q) \tag{4.2}$$

4.2.2 Data

My goal is to build a task for a low-resource language, such as Bulgarian, as close as possible to the multiple-choice reading comprehension setup for high-resource languages such as English. This will allow us to evaluate the limitations of transfer learning in a multilingual setting. One of the largest datasets for this task is RACE (Lai et al., 2017), with a total of 87,866 training questions with four answer candidates for each. Moreover, there are 25,137 contexts mapped to the questions and their correct answers.

While there exist many datasets for reading comprehension, most of them are in English, and there are a very limited number in other languages (Peñas et al., 2012, 2014). Hereby, I collect my own dataset for Bulgarian, resulting in

Domain	#QA-pairs	#Choices	Len Question	Len Options	Vocabulary Size
12th Grade Matriculation Exam					
Biology	437	4	10.4	2.6	2,414 (12,922)
Philosophy	630	4	8.9	2.9	3,636 (20,392)
Geography	612	4	12.8	2.5	3,239 (17,668)
History	542	4	23.7	3.6	5,466 (20,456)
Online History Quizzes					
Bulgarian History	229.0	4	14.0	2.8	2,287 (10,620)
PzHistory	183	3	38.9	2.4	1,261 (7,518)
Overall	2,633	3.9	15.7	2.9	13,329 (56,104)
RACE Train - Mid and High School					
RACE-M	25,421	4.0	9.0	3.9	32,811
RACE-H	62,445	4.0	10.4	5.8	125,120
Overall	87,866	4.0	10.0	5.3	136,629

Table 4.1: Statistics about my Bulgarian dataset compared to the RACE dataset.

(**Biology**) The thick coat of mammals in winter is an example of:
A. physiological adaptation
B. behavioral adaptation
C. genetic adaptation
D. morphological adaptation

(**Philosophy**) According to relativism in ethics:
A. there is only one moral law that is valid for all
B. there is no absolute good and evil
C. people are evil by nature
D. there is only good, and the evil is seeming

(**Geography**) Which of the assertions about the economic specialization of the Southwest region is true?
A. The ratio between industrial and agricultural production is 15:75
B. Lakes of glacial origin in Rila and Pirin are a resource for the development of tourism
C. Agricultural specialization is related to the cultivation of grain and ethereal-oil crops
D. The rail transport is of major importance for intra-regional connections

(**History**) Point out the concept that is missed in the text of the Turnovo Constitution: „Art. 54 All born in Bulgaria, also those born elsewhere by parents Bulgarian _____, count as _____ of the Bulgarian Principality. Art. 78 Initial teaching is free and obligatory for all _____ of the Bulgarian Principality."
A. residents
B. citizents
C. electors
D. voters

(**History Quiz**) Sofroniy Vrachanski started a family that plays a big role in the history of the Bulgarian National Revival. What is its name?
A. Georgievi
B. Tapchileshtovi
C. Bogoridi
D. Palauzovi

Table 4.2: Example questions, one per subject, from the Bulgarian dataset. The correct answer is marked in green.

2,633 multiple-choice questions, without contexts, from different subjects: biology (16.6%), philosophy (23.93%), geography (23.24%), and history (36.23%). Table 4.2 shows an example question with candidate answers chosen to represent best each category. I use green to mark the correct answer, and bold for the question category. For convenience all the examples are translated to English.

Table 4.1 shows the distribution of questions per subject category, the length (in words) for both the questions and the options (candidate answers), and the vocabulary richness, measured in terms of unique words. The first part of the table presents statistics about the dataset, while the second part is a comparison to RACE (Lai et al., 2017).

I divided the Bulgarian questions into two groups based on the question's source. The first group (*12th Grade Matriculation Exam*) was collected from twelfth grade matriculation exams created by the Ministry of Education of Bulgaria in the period 2008–2019. Each exam contains thirty multiple-choice questions with four possible answers per question. The second set of questions (*Online History Quizzes*)

are history-related and are collected from online quizzes. While they are not created by educators, the questions are still challenging and well formulated. Furthermore, I manually filtered out questions with non-textual content (i.e., pictures, paintings, drawings, etc.), ordering questions (i.e., order the historical events), and questions involving calculations (i.e., how much X we need to add to Y to arrive at Z).

Table 4.1 shows that history questions in general contain more words (14.0–38.9 on average), compared to other subjects (8.9–12.8 on average). A tangible difference in length compared to other subjects is seen for *12th grade History* and *PzHistory*, due to the large number of quotes, and document pieces contained in questions from these two groups. Also, the average question length is 15.7, which is longer compared to the RACE dataset with 10.0. On the other hand, the option lengths per subject category in my dataset follow a narrower distribution. They fall in the interval between 2.5 and 2.9 words on average, expect for *12th grade History*, with 3.6 words. Here, I note a significant difference compared to the option lengths in RACE, which tend to be 2.4 words longer on average – 5.3 for RACE vs. 2.9 for ours.

Finally, I examine the vocabulary richness of the two datasets. The total number of unique words is shown in the last column of Table 4.1 (Vocab Size). For my dataset, there are two numbers per row: the first one shows statistics based on the question–answer pairs only, while the second one, enclosed in parentheses, measures the vocabulary size including the extracted passages by the Context Retriever. The latter number is a magnitude estimate rather then a concrete number, since its upper limit is the number of words in Wikipedia, and it can vary for different retrieval strategies.

4.2.3 Experiments and Evaluation

BERT Fine-Tuning

I divide the fine-tuning into two groups of models (*i*) Multilingual BERT, and (*ii*) Slavic BERT. Table 4.3 below presents the results in the multiple-choice comprehension task on the dev dataset from RACE (Lai et al., 2017).

Multilingual BERT As my initial model, I use $\text{BERT}_{\text{base}}$, Multilingual Cased which is pre-trained on 104 languages, and has 12-layers, 768-hidden units per layer, 12-heads, and a total of 110M parameters. I further fine-tune the model on RACE (Lai et al., 2017) for 3 epochs saving a checkpoint after each epoch. I use a batch size of 8, a max sequence size of 320, and a learning rate of 1e-5.

#Epoch	RACE-M	RACE-H	Overall
Multilingual BERT			
1	64.21	53.66	56.73
2	68.80	57.58	60.84
3	69.15	58.43	61.55
Slavic BERT			
2	53.55	44.48	47.12
3	57.38	46.88	49.94

Table 4.3: Accuracy measured on the dev RACE dataset after each training epoch.

Slavic BERT The Slavic model[4] was built using transfer learning from the Multilingual BERT model to four Slavic languages: Bulgarian, Czech, Polish, and Russian. In particular, the Multilingual BERT model was fine-tuned on a stratified dataset of Russian news and Wikipedia articles for the other languages. I use this pre-trained Slavic BERT model, and I apply the same learning procedure as for *Multilingual BERT*.

Wikipedia Retrieval and Indexing

Here, I discuss the retrieval setup (see Section 4.2.1 for more details). I use the Bulgarian dump of Wikipedia from 2019-04-20, with a total of 251,507 articles. I index each article title and body in plain text, which I call a *passage*. I further apply additional processing for each field:

- *ngram*: word-based 1–3 grams;
- *bg*: lowercased, stop-words removed (from Lucene), and stemmed (Savoy, 2007);
- *none*: bag-of-words index.

I ended up using a subset of four fields from all the possible analyzer-field combinations, namely *title.bg*, *passage*, *passage.bg*, and *passage.ngram*. I applied Bulgarian analysis on the *title* field only as it tends to be short and descriptive, and thus very sensitive to noise from stop-words, which is in contrast to questions that are formed mostly of stop-words, e.g., *what*, *where*, *when*, *how*.

For indexing the Wikipedia articles, I adopt two strategies: sliding window and paragraph. In the window-based strategy, I define two types of splits: small, containing 80-100 words, and large, of around 300 words. In order to obtain indexing chunks, I define a window of size K, and a stride equal to one forth of K. Hence, each $\frac{K}{4}$ characters, which is the size of the stride, are contained into four different documents. The paragraph-based strategy divides the article by splitting it using one or more successive newline characters ([\n]+) as a delimiter. I avoid indexing

[4] http://github.com/deepmipt/Slavic-BERT-NER

entire documents due to their extensive length, which can be far beyond the maximum length that BERT can take as an input, i.e., 320 word pieces (see Section 4.2.3 for the more details). Note that extra steps are needed in order to extract a proper passage from the text. Moreover, the amount of facts in the Wikipedia articles that are unrelated to the questions give rise to false positives since the question is short and term-unspecific.

Finally, I use a list of top-N hits for each candidate answer. Thus, I have to execute an additional query for each question + option combination, which may result in duplicated passages, thus introducing an implicit bias towards the candidates they support. In order to mitigate this effect, during the answer selection phase (see Section 4.2.1), I remove all duplicate entries, keeping a single instance.

Experimental Results

Here, I discuss the accuracy of each model on the original English MRC task, followed by experiments in zero-shot transfer to Bulgarian.

English Pre-training for Multiple-Choice MRC. Table 4.3 presents the change in accuracy on the original English comprehension task, depending on the number of training epochs. In the table, "BERT" refers to the Multilingual BERT model, while "Slavic" stands for BERT with Slavic pre-training. I further fine-tune the models on the RACE dataset. Next, I report their performance in terms of accuracy, following the notation from (Lai et al., 2017). Note that the questions in RACE-H are more complex than those in RACE-M. The latter has more word matching questions and fewer reasoning questions. The final column in the table, *Overall*, shows the accuracy calculated over all questions in the RACE testset. I train both setups for three epochs and I report their performance after each epoch. We can see a positive correlation between the number of epochs and the model's accuracy. We further see that the Slavic BERT performs far worse on both RACE-M and RACE-H, which suggests that the change of weights of the model towards Slavic languages has led to catastrophic forgetting of the learned English syntax and semantics. Thus, it should be expected that the adaptation to Slavic languages would yield decrease in performance for English. What matters though is whether this helps when testing on Bulgarian, which I explore next.

Zero-Shot Transfer. Here, I assess the performance of the model when applied to Bulgarian multiple-choice reading comprehension. Table 4.4 presents an ablation study for various components. Each line denotes the type of the model, and the addition (+) or the removal (–) of a characteristic from the setup in the previous line. The first line shows the performance of a baseline model that chooses an option uniformly at random from the list of candidate answers for the target question. The

Setting	**Accuracy**
Random	24.89
Train for 3 epochs	–
+ window & title.bg & pass.ngram	29.62
+ passage.bg & passage	39.35
– title.bg	39.69
+ passage.bg^2	40.26
+ title.bg^2	40.30
+ bigger window	36.54
+ paragraph split	42.23
+ Slavic pre-training	33.27
Train for 1 epoch best	40.26
Train for 2 epochs best	41.89

Table 4.4: Accuracy on the Bulgarian testset: ablation study when sequentially adding/removing different model components.

following rows show the results for experiments conducted with a model trained for three epochs on RACE (Lai et al., 2017).

The basic model uses the following setup: Wikipedia pages indexed using a small sliding window (400 characters, and stride of 100 characters), and context retrieval over two fields: Bulgarian analyzed title (*text.bg*), and word *n*-grams over the passage (*passage.ngram*). This setup yields 29.62% accuracy, and it improves over the random baseline by 4.73% absolute. We can think of it as a non-random baseline for further experiments. Next, I add two more fields to the IR query: passage represented as a bag of words (named *passage*), and Bulgarian analyzed (*passage.bg*), which improves the accuracy by additional 10%, arriving at 39.35%. The following experiment shows that removing the *title.bg* field does not change the overall accuracy, which makes it an insignificant field for searching. Further, I add double weight on *passage.bg*, (shown as ^2), which yields 1% absolute improvement.

From the experiments described above, I found the best combination of query fields to be *title.bulgarian*^2, *passage.ngram*, *passage*, *passage.bulgarian*^2, where the *title* has a minor contribution, and can be sacrificed for ease of computations and storage. Fixing the best query fields, allowed me to evaluate other indexing strategies, i.e., bigger window (size 1,600, stride 400) with accuracy 36.54%, and paragraph splitting, with which I achieved the highest accuracy of 42.23%. This is an improvement of almost 2.0% absolute over the small sliding window, and 5.7% over the large one.

Next, I examined the impact of the Slavic BERT. Surprisingly, it yielded 9% absolute drop in accuracy compared to the multi-lingual BERT. This suggests that the latter already has enough knowledge about Bulgarian, and thus it does not need further adaptation to Slavic languages.

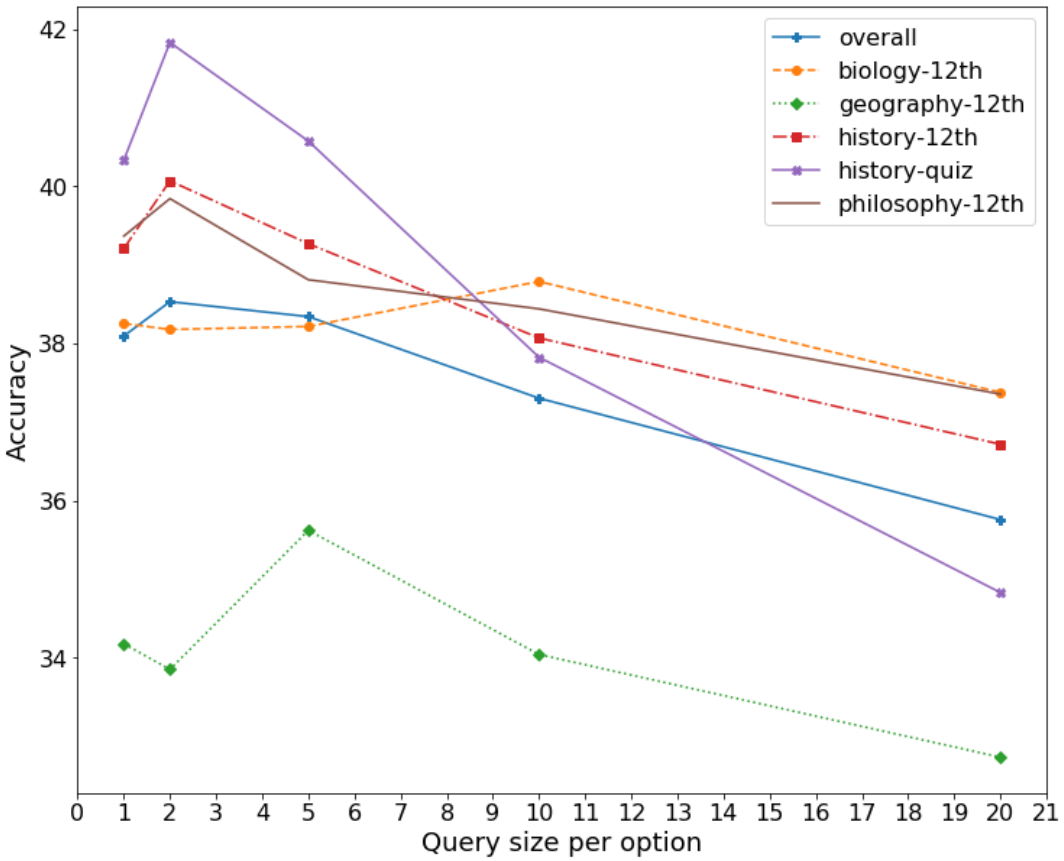


Figure 4.2: Accuracy per question category based on the number of query results per answer option.

Further, I study the impact of the number of fine-tuning epochs on the model's performance. I observe an increase in accuracy as the number of epochs grows, which is in line with previously reported results for English tasks. While this correlation is not as strong as for the original RACE task (see Table 4.3 for comparison), I still observe 1.6% and 0.34% absolute increase in accuracy for epochs 2 and 3, respectively, compared to epoch 1. Note that I do not go beyond three epochs, as previous work has suggested that 2-3 fine-tuning epochs are enough (Devlin et al., 2019), and after that, there is a risk of catastrophic forgetting of what was learned at pre-training time (note that I have already seen such forgetting with the Slavic BERT above).

I further study the impact of the size of the results list returned by the retriever on the accuracy for the different categories. Figure 4.2 shows the average accuracy for a given query size S_q over all performed experiments, where $S_q \in \{1, 2, 5, 10, 20\}$. We can see in Figure 4.2 that longer query result lists (i.e., containing more than 10 results) per answer option worsen the accuracy for all categories, except for *biology*, where we see a small peak at length 10, while still the best overall results for this category is achieved for a result list of length 5. A single well-formed maximum at length 2 is visible for *history* and *philosophy*. With these

two categories being the biggest ones, the cap at the same number of queries for the overall accuracy is not a surprise. The per-category results for the experiments are discussed in more detail in the next Section '*Per-Category Results*'.

We can see that the highest accuracy is observed for *history*, particularly for online quizzes, which are not designed by educators and are more of a word-matching nature rather then a reasoning one (see Table 4.2). Finally, *geography* appears to be the hardest category with only 38.73% accuracy: 3.5% absolute difference compared to the second-worst category. The performance for this subject is also affected differently by changes in query result length: the peak is at lengths 5 and 10, while there is a drop for length 2. A further study of the model's behavior can be found in Section 4.2.4.

Per-Category Results Table 4.5 gives an overview, including per-category breakdown, of my parameter tuning experiments. I present the results for some interesting experiments rather then for a full grid search. The first row shows a random baseline for each category. In the following rows, I compare different types of indexing: first, I show the results for a small sliding window (400-character window, and 100-character stride), followed by a big window (1,600-character window, and 400-character stride), and finally for paragraph indexing. I use the same notation as in Section 4.2.3. The last group in the table (*Paragraph*) shows the best-performing model, where I mark in bold the highest accuracy for each category. For completeness, I also show the accuracy when using the *Slavic BERT* model for prediction, which yields a 10% drop on average compared to using the *Multilingual BERT*, for each of the categories.

4.2.4 Case Study

In Table 4.6, I present the retrieved evidence passages for the example questions in Table 4.2: I omit the answers, and I only show the questions and the contexts. Each example is separated by a double horizontal line, where the first row is the question starting with "*Q:*", and the following rows contain passages returned by the retriever. For each context, I normalize the raw scores from the comprehension model using Eq. 4.1 to obtain a probability distribution. I then select an answer using arg max, according to Eq. 4.2. In the table, I indicate the correctness of each predicted answer using one of the following symbols before the question:

- ✓ The question is answered correctly.
- ✗ An incorrect answer has the highest score.
- ? Two or more answers have the highest score.

I show the top retrieved result in order to illustrate the model scores over different evidence passages and the quality of the articles. The queries are formed by concatenating the question with an answer option, even though this can lead to

#docs	Overall	biology-12th	philosophy-12th	geography-12th	history-12th	history-quiz
Random						
0	24.89	26.09	24.44	24.18	25.87	24.03
Window Small						
title.bulgarian, passage.bulgarian						
1	39.95	40.27	40.63	34.97	42.99	41.99
2	40.22	40.27	40.63	35.95	42.62	42.72
5	40.22	38.90	40.63	38.07	41.51	42.48
10	38.66	40.50	39.84	35.46	39.30	38.83
20	36.84	37.53	39.05	33.82	38.75	34.71
title.bulgarian, passage.ngram						
1	28.94	29.06	32.06	27.29	27.49	28.40
2	29.09	29.06	33.33	25.00	28.78	29.13
5	29.05	27.46	32.06	26.63	30.63	27.67
10	29.62	29.06	32.54	26.96	30.07	29.13
20	29.43	31.81	32.70	26.63	28.60	27.18
title.bulgarian, passage.ngram, passage, passage.bulgarian						
1	38.32	38.22	40.00	34.48	39.48	40.05
2	39.08	37.07	40.32	34.48	40.59	44.17
5	39.35	40.96	39.84	34.64	41.33	41.26
10	38.63	40.50	40.63	33.50	40.41	38.83
20	36.54	38.67	37.94	31.37	37.45	38.59
passage.ngram, passage, passage.bulgarian^2						
1	39.69	40.27	40.63	35.13	42.07	41.26
2	40.26	39.82	40.95	35.95	42.62	42.96
5	39.57	39.59	39.37	37.25	40.96	41.50
10	38.70	41.19	39.52	35.78	39.30	38.35
20	37.14	39.36	37.78	35.29	38.38	34.95
title.bulgarian^2, passage.ngram, passage, passage.bulgarian^2						
1	39.84	40.27	40.79	35.13	42.25	41.75
2	40.30	40.27	40.63	36.11	42.80	42.72
5	40.26	39.13	40.63	38.40	41.14	42.48
10	38.74	40.50	39.68	35.62	39.48	39.08
20	37.07	37.76	39.05	34.64	38.56	34.95
Window Big						
title.bulgarian^2, passage.ngram, passage, passage.bulgarian^2						
1	31.22	28.38	33.97	29.41	30.81	33.25
2	33.12	31.58	37.46	31.21	33.95	29.85
5	36.04	35.70	38.10	33.82	37.82	34.22
10	36.54	37.30	36.03	33.99	39.30	36.65
20	35.62	34.55	39.68	31.05	38.38	33.74
Paragraph						
title.bulgarian^2, passage.ngram, passage, passage.bulgarian^2						
1	41.82	41.42	42.06	38.07	40.96	48.54
2	**42.23**	42.56	**43.17**	35.62	**42.99**	**49.27**
5	41.59	**43.25**	40.32	**38.73**	40.04	48.06
10	39.46	40.96	38.41	36.93	39.85	42.72
20	37.52	39.13	37.62	34.64	38.56	38.59
Slavic BERT						
1	33.19	30.89	33.17	28.76	32.29	43.45
2	33.27	31.58	31.90	31.21	35.24	37.62
5	31.14	30.21	30.16	29.25	31.00	36.65
10	30.42	29.29	29.68	29.74	31.92	31.80
20	29.66	28.60	29.37	28.43	32.10	29.85

Table 4.5: Evaluation results for the Bulgarian multiple-choice reading comprehension task: comparison of various indexing and query strategies.

Context	Pr_A	Pr_B	Pr_C	Pr_D
✓ Q: The thick coat of mammals in winter is an example of: 1) The hair cover is a rare and rough bristle. In winter, soft and dense hair develops between them. Color ranges from dark brown to gray, individually and geographically diverse	0.19	0.19	0.15	0.47
✗ Q: According to relativism in ethics: 1) Moral relativism	0.45	0.24	0.10	0.21
2) In ethics, relativism is opposed to absolutism. Whilst absolutism asserts the belief that there are universal ethical standards that are inflexible and absolute, relativism claims that ethical norms vary and differ from age to age and in different cultures and situations. It can also be called epistemological relativism - a denial of absolute standards of truth evaluation.	0.28	0.41	0.09	0.22
✓ Q: Which of the assertions about the economic specialization of the Southwest region is true? 1) Geographic and soil-climatic conditions are blessed for the development and cultivation of oil-bearing rose and other essential oil crops.	0.12	0.52	0.28	0.08
2) Kirov has an airport of regional importance. Kirov is connected with rail transport with the cities of the Transsiberian highway (Moscow and Vladivostok).	0.14	0.27	0.06	0.53
3) Dulovo has always been and remains the center of an agricultural area, famous for its grain production. The industrial sectors that still find their way into the city's economy are primarily related to the primary processing of agricultural produce. There is also the seamless production that evolved into small businesses with relatively limited economic significance.	0.25	0.05	0.67	0.03
4) In the glacial valleys and cirques and around the lakes in the highlands of Rila and Pirin, there are marshes and narrow-range glaciers (overlaps).	0.10	0.72	0.08	0.10
? Q: Point out the concept that is missed in the text of the Turnovo Constitution: ... 1) _____	0.26	0.26	0.26	0.22
✓ Q: Sofroniy Vrachanski sets up a genre that plays a big role in the history of the Bulgarian Revival. What is his name? 1) Bogoridi is a Bulgarian Chorbadji genus from Kotel. Its founder is Bishop Sofronius Vrachanski (1739-1813). His descendants are:	0.06	0.16	0.68	0.10

Table 4.6: Retrieved unique top-1 contexts for the example questions in Table 4.2. The passages are retrieved using queries formed by concatenating a question with an answer option.

duplicate results since some answers can be quite similar or the question's terms could dominate the similarity score. The questions in Table 4.6 are from five different categories: biology, philosophy, geography, history, and online quizzes. Each of them has its own specifics and gives me an opportunity to illustrate a different model behavior.

The first question is from the biology domain, and we can see that the text is very general, and so is the retrieved context. The latter talks about *hair* rather than *coat*, and the correct answer (D) *morphological adaptation* is not present in the retrieved text. On the other hand, all the terms are only connected to it, and hence the model assigns high probability to this answer option.

For the second question, from the philosophy domain, there are two related contexts found. The first one is quite short, noisy, and it does not give much information in general. The second paragraph manages to extract the definition of *relativism* and to give good supporting evidence for the correct answer, namely that *there is no absolute good and evil* (B). As a result, this option is assigned high probability. Nevertheless, the incorrect answer *here is only one moral law that is valid for all* (A) is assigned an even higher probability and it wins the voting.

In the third example, from the domain of geography, we see a large number of possible contexts, due to the long and descriptive answers. We can make two key observations: (*i*) the query is drawn in very different directions by the answers, and (*ii*) there is no context for *Southwestern region*, and thus, in the second option, the result is for Russia, not for Bulgaria. The latter passage pushes the probability mass to an option that talks about transportation (D), which is incorrect. Fortunately, the forth context has an almost full term overlap with the correct answer (B), and thus gets very high probability assigned to it: 72%.

The fourth question, from the history domain, asks to point out a missing concept, but the query is dominated by the question, and especially by underscores, leading to a single hit, counting only symbols, without any words. As expected, the model assigned uniform probability to all classes.

The last question, a history quiz, is a factoid one, and it lacks a reasoning component, unlike the previous examples. The query returned a single direct match. The retrieved passage contains the correct answer exactly: option *Bogoridi* (C). Thereby, the comprehension model assigns to it a very high probability of 68%.

4.3 Answer Retrieval from a Pool of Explanations

The massive spread of disinformation online, especially in social media, was counter-acted by major efforts to limit the impact of false information not only by journalists and fact-checking organizations but also by governments, private companies, researchers, and ordinary Internet users. Such efforts include, but are

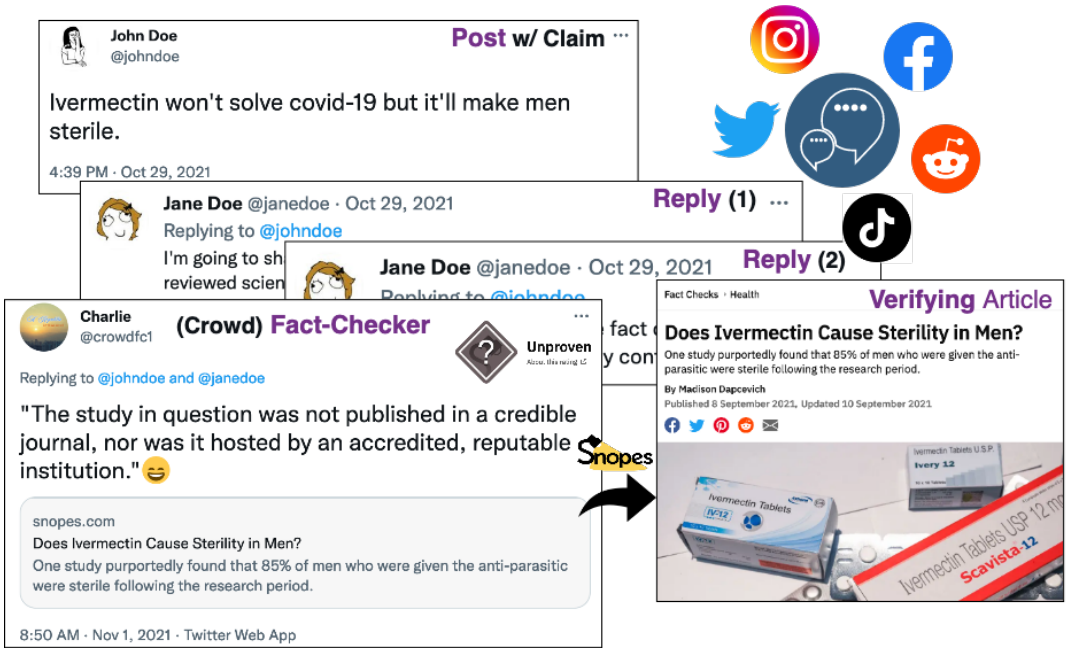


Figure 4.3: Crowd fact-checking thread on Twitter. The first tweet (**Post w/ claim**) makes the claim that *Ivermectin causes sterility in men*, which then receives **replies**. A **(crowd) fact-checker** replies with a link to a **verifying article** from a fact-checking website. I pair the *article* with the *tweet that made this claim* (the first post ✓), as it is irrelevant (✗) to the other *replies*.

not limited to building systems for automatic fact-checking (Thorne and Vlachos, 2018; Guo et al., 2022), rumor debunking (Zubiaga et al., 2016; Derczynski et al., 2017), fake news detection (Ferreira, 2016; Pomerleau and Rao, 2017), and media profiling (Baly et al., 2020; Stefanov et al., 2020), among others.

I study the following problem of detecting previously fact-checked claims: *Given a user comment, detect whether the claim it makes was previously fact-checked with respect to a collection of verified claims and their corresponding articles* (see Table 4.7). This task is an integral part of an end-to-end fact-checking pipeline (Hassan et al., 2017), and also an important task on its own right as people often repeat the same claim (Barrón-Cedeno et al., 2020; Vo and Lee, 2020; Shaar et al., 2021). Research on this problem is limited by data scarceness, with datasets typically having about a 1,000 tweet–verifying article pairs (Barrón-Cedeno et al., 2020; Shaar et al., 2020, 2021), with the notable exception of Vo and Lee (2020), which contains 19K claims about images matched against 3K fact-checking articles.

I propose to bridge this gap using crowd fact-checking to create a large collection of tweet–verifying article pairs, which I then label (if the pair is correctly matched) automatically using distant supervision. An example is shown in Figure 4.3.

User Post w/ Claim: Sen. Mitch McConnell: "As recently as October, now-President Biden said you can't legislate by executive action unless you are a dictator. Well, in one week, he signed more than 30 unilateral actions." [URL] — Forbes (@Forbes) January 28, 2021		
Verified Claims and their Corresponding Articles		
(1)	When he was still a candidate for the presidency in October 2020, U.S. President Joe Biden said, "You can't legislate by executive order unless you're a dictator." http://snopes.com/fact-check/biden-executive-order-dictator/	✓
(2)	U.S. Sen. Mitch McConnell said he would not participate in 2020 election debates that include female moderators. http://snopes.com/fact-check/mitch-mcconnell-debate-female/	✗

Table 4.7: Illustrative examples for the task of detecting previously fact-checked claims. The **post contains a claim** (related to *legislation and dictatorship*), the **Verified Claims** are part of a search collection of previous fact-checks. In row (*1*), the fact-check is a correct match for the claim made in the tweet (✓), whereas in (2), the claim still discusses *Sen. Mitch McConnell*, but it is a different claim (✗), and thus it forms an incorrect pair.

4.3.1 My Newly Collected Dataset: CrowdChecked

Dataset Collection

I use Snopes as my target fact-checking website, due to its popularity among both Internet users and researchers (Popat, Kashyap and Mukherjee, Subhabrata and Strötgen, Jannik and Weikum, Gerhard, 2016; Hanselowski et al., 2019; Augenstein et al., 2019; Tchechmedjiev et al., 2019). I further use Twitter as the source for collecting user messages, which could contain claims and fact-checks of these claims.

My data collection setup is similar to the one in Vo and Lee (2019). First, I form a query to select tweets that contain a link to a fact-check from Snopes (*url:snopes.com/fact-check/*), which is either a reply or a quote tweet, and not a retweet.[5] An example result from the query is shown in Figure 4.3, where the tweet *from the crowd fact-checker* contains a link to a fact-checking article. I then assess its appropriateness to the claim (if any) made in the first tweet (the root of the conversation) and the last reply in order to obtain tweet–verified article pairs. I analyze in more detail the conversational structure of these threads in Section 4.3.1, '*Tweet Collection (Conversation Structure)*'.

I then collect all tweets matching the query from October 2017 till October 2021, obtaining a total of 482,736 unique hits. I further collect 148,503 reply tweets and 204,250 conversation (root) tweets.[6] Finally, I filter out malformed pairs, i.e., tweets linking to themselves, empty tweets, non-English ones, such with no resolved URLs in the Twitter object ('*entities*'), with broken links to the fact-checking website, and

[5] I exclude retweets, as they do contain no comments, but rather share previous tweets.

[6] The sum of the unique replies and of the conversation tweets is not equal to the number of fact-checking tweets, as more than one tweet might reply to the same comment.

all tweets in the *CheckThat '21* dataset. After cleaning the dataset, I ended up with 332,660 unique tweet–article pairs (shown in first row in Table 4.11), 316,564 unique tweets, and 10,340 fact-checking articles from Snopes they could point to. More detail about the fact-checking articles collection and statistics are given in Section 4.3.1 and on Figure 4.5.

Tweet Collection (Conversation Structure)

It is important to note that this *'fact-checking'* tweet can be part of a multiple-turn conversational thread, therefore taking the post that it replies to (previous turn), does not always express a claim which the current tweet targets. In order to better understand that phenomena, I perform manual analysis of conversation thread. The conversational threads are organized in a similar way shown Figure 4.3, i.e., the root is the first comment, then there can be a long discussion, followed by a fact-checking comment (the one with the Snopes link). In my analysis I identify four patterns: (*i*) current tweet verifies a claim in the the tweet it replies to, (*ii*) the tweet verifies the root of the conversation, (*iii*) the tweet does not verify any claim in the chain (a common scenario), (*iv*) in very few cases the fact-check targets a claim expressed not in the root or the closest tweet. This analysis suggests that for the task of detecting previously fact-checked claims, it is sufficient to collect the triplet of the fact-checking tweet, root of the conversation (*conversation*), and the tweet that the target tweet is replying to (reply).

Comparison to Existing Datasets

Next, I compare my dataset to a closely related dataset from the CLEF-2021 CheckThat '21 on Detecting Previously Fact-Checked Claims in Tweets (Shaar et al., 2021), to which I will refer as *CheckThat '21* in the rest of the chapter. There exist other related datasets that are smaller (Barrón-Cedeno et al., 2020), come from a different domain (Shaar et al., 2021), are not in English (Elsayed et al., 2019), or are multimodal (Vo and Lee, 2020).

Table 4.8 compares *CrowdChecked* to *CheckThat '21* in terms of number of examples, length of the tweets, and vocabulary size. Before I calculated these statistics, I lowercased the text and I removed all URLs, Twitter handlers, English stop words, and punctuation. We can see in Table 4.8 that *CrowdChecked* contains two orders of magnitude more examples, slightly shorter tweets (but the maximum length stays approximately the same, which can be explained by the word limit of Twitter), and has a vocabulary size that is an order of magnitude larger. Note, however, that many examples in *CrowdChecked* are incorrect matches (see Section 4.3.1), and thus I use distant supervision to label them (see Section 4.3.1), with the resulting dataset sizes of matching pairs shown in Table 4.11. Here, I want to emphasize that there is absolutely no overlap between CrowdChecked and CheckThat '21 in terms of tweets/claims.

Dataset	**Tweets**‡	**Words**			**Vocab**
	|Unique|	Mean	50%	Max	|Unique|
CrowdChecked (Mine)	316,564	12.2	11	60	114,727
CheckThat '21	1,399	17.5	16	62	9,007

Table 4.8: Statistics about out dataset vs.CheckThat '21. ‡The number of unique tweets is lower compared to the total number of tweet–article pairs, as one tweet can be fact-checked by multiple articles.

In terms of topics, the claims in both my dataset and *CheckThat '21* are quite diverse, including fact-checks for a broad set of topics related, but not limited to politics (e.g., the Capitol Hill riots, U.S. elections), pop culture (e.g., famous performers and actors such as Drake and Leonardo di Caprio), brands (e.g., McDonald's and Disney), and COVID-19, among many others. Illustrative examples of the claim/topic diversity can be found in Tables 4.7 and A.3. Moreover, the collection of Snopes articles contains almost 14K different fact-checks on an even wider range of topics, which further diversifies the set of tweet–article pairs.

More detail about the process of collecting the fact-checking articles is given in Section 4.3.1. Finally, I compare the set of Snopes fact-checking articles referenced by the crowd fact-checkers to the ones included in the CheckThat '21 competition.

Data Labeling (Distant Supervision)

To label the examples, I experiment with two distant supervision approaches: (*i*) based on the Jaccard similarity between the tweet and its fact-checking article, and (*ii*) based on the predictions of a model trained on CheckThat '21.

Jaccard Similarity In this approach, I first pre-process the texts by converting them to lowercase, removing all URLs and replacing all numbers with a single zero. Then, I tokenize the texts using the NLTK's *Twitter tokenizer* (Loper and Bird, 2002), and I strip all handles and user mentions. The final preprocessing step is to filter out all stop words[7] and punctuation (including quotes and special symbols) and to stem (Porter, 1980) all tokens.

In order to obtain a numerical score for each tweet–article pair, I calculate the *Jaccard similarity* (jac) between the normalized tweet text and each of the *title* and the *subtitle* from the Snopes article (i.e., the intersection over the union of the unique tokens). Both fields present a summary of the fact-checked claim, and thus should include more compressed information. Finally, I average these two similarity values to obtain a more robust score. Statistics are shown in Table 4.9.

Semi-Supervision Here, I train a Sentence-BERT (Reimers and Gurevych, 2019) model, as described in Section 4.3.2, using the manually annotated data from

[7] I use the predefined list of English stop words in the NLTK library.

Range (Jaccard)	**Examples** (%)	**Correct Pairs** Reply (%)	**Correct Pairs** Conv. (%)
[0.0;0.1)	62.57	5.88	0.00
[0.1;0.2)	18.98	36.36	14.29
[0.2;0.3)	10.21	46.67	50.00
[0.3;0.4)	4.17	76.47	78.57
[0.4;0.5)	2.33	92.86	92.86
[0.5;0.6)	1.08	94.12	94.12
[0.6;0.7)	0.43	80.00	80.00
[0.7;0.8)	0.11	92.31	92.31
[0.8;0.9)	0.05	91.67	92.86
[0.9;1.0]	0.02	100.00	100.00

Table 4.9: Proportion of examples in different bins based on average Jaccard similarity between the tweet ↔ the title/subtitle. Manual annotations of *correct pairs* (i.e., tweet–article pairs, where the article fact-checks the claim in the tweet).

Range (Cosine)	**Examples** (%)	**Correct Pairs** (%)
[-0.4;0.1)	37.83	0.00
[0.1;0.2)	16.50	6.67
[0.2;0.3)	12.28	41.46
[0.3;0.4)	10.12	36.36
[0.4;0.5)	8.58	63.16
[0.5;0.6)	6.69	70.00
[0.6;0.7)	4.47	84.21
[0.7;0.8)	2.48	96.15
[0.8;0.9)	0.97	93.10
[0.9;1.0]	0.08	100.00

Table 4.10: Proportion of examples in different bins based on cosine similarity using Sentence-BERT trained on *CheckThat '21*. Manual annotations of *correct pairs*.

CheckThat '21. The model shows strong performance on the testing set of CheckThat '21 (see Table 4.12), and thus I expect it to have good precision at detecting matching fact-checked pairs. In particular, I calculate the *cosine similarity* between the embeddings of the fact-checked tweet and the fields from the Snopes article. Statistics about the scores are shown in Table 4.10.

Feasibility Evaluation

To evaluate the feasibility of the obtained labels, I performed manual annotation, aiming to estimate the number of *correct pairs* (i.e., tweet–article pairs, where the article fact-checks the claim in the tweet). My prior observations of the data suggested that unbiased sampling from the pool of tweets was not suitable, as it would include mostly pairs that have very few overlapping words, which is often an indicator that the texts are not related. Thus, I sample the candidates for annotation

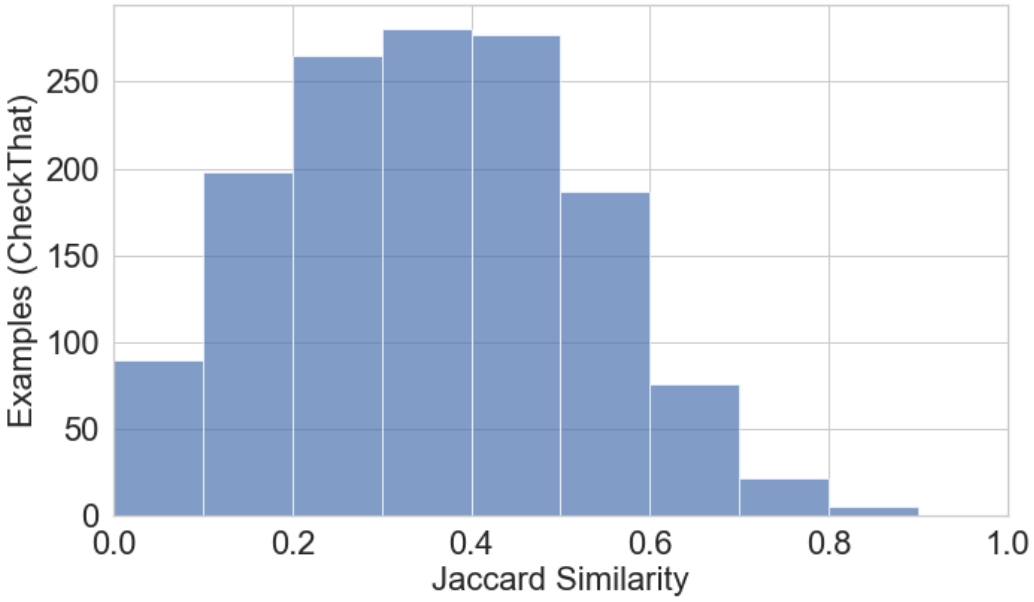


Figure 4.4: Distribution of the Jaccard similarity scores. The score is an average of the *sim(tweet, title)* and *sim(tweet, subtitle)*.

based on their Jaccard similarity, i.e., I divided the range of possible values [0;1] into 10 equally sized bins and I sampled 15 examples from each bin, resulting into 150 conversation–reply–tweet triples. Afterwards, the appropriateness of each reply-article and conversation-article pair is annotated by three annotators independently. The annotators had a *good level* of inter-annotator agreement: 0.75 in terms of Fleiss Kappa (Fleiss, 1971) (see Section A.1.2).

Tables 4.9 and 4.10 show the resulting estimates of *correct pairs* for both Jaccard and cosine-based labeling. In the case of Jaccard, we can see that the expected number of correct examples is very high (over 90%) in the range of *[0.4–1.0]*, and then it drastically decreases, going to almost zero when the similarity is less than 0.1. Similarly, for the cosine score, we can see high number of matches in the top 4 bins (*[0.6–1.0]*), albeit the number of matches remains relatively high in the following interval of *[0.2–0.6)* between 36% and 63%, and again gets close to zero for the lower-score bins. Next, I analyze the distribution of the Jaccard scores in CheckThat '21 in more detail.

Finally, I analyze the distribution of the Jaccard scores in the CheckThat '21, shown in Figure 4.4. The distribution is different compared to the one observed in the my newly collected dataset, as it peaks at around 0.4, and is slightly shifted towards lower similarity values, suggesting the examples included are not easily solvable with basic lexical features (Shaar et al., 2021), which I also observe in my experiments (see Section 4.3.3).

Fact-checking Articles Collection

In order to obtain a collection of fact-checking articles for each tweet, I first formed a list of unique URLs shared in the fact-checking tweets from the crowd fact-checkers. Next, from each URL I downloaded the HTML of the whole page and extracted the meta information using CSS selectors and RegEx rules. In particular, I follow previous work (Barrón-Cedeno et al., 2020; Shaar et al., 2021) and collect: *title*, the title of the page, *subtitle*, short description of the fact-check, *claim*, the claim of interest, *subtitle*, short description of the fact-check, *date*, the date on the article was published, *author*, the author of the article. I do not parse the content of the article and factual label, as the credibility of the claim is not related to the objective of this task, i.e., the goal is to find a fact-checking article, but not to verify it.

As a result I collected 10,340 articles that are published in the period between 1995–2021. The per-year distribution is shown in Table 4.5 (in brown). The majority of the articles are from the period after 2015, with a peak at the ones from 2020/2021. I attribute this on the increased media literacy and on the nature of the Twitter dynamics (Zubiaga, 2018).

Fact-Check Articles Comparison Finally, I compare the set of Snopes fact-checking articles referenced by the crowd fact-checkers to the ones included in the CheckThat '21 competition. We can see that the tweets in CrowdChecked refer to around 3.5K less articles (namely 10,340), compared to CheckThat '21, which consists of 13,835 articles. A total of 8,898 articles are present in both datasets. Since the CheckThat '21 is collected earlier, it includes less articles from recent years compared to CrowdChecked, and peaks at 2016/2017. Nevertheless, for CheckThat '21, the number of Snopes articles included in a claim–article pair is far less compared to my dataset (even after filtering out the unrelated pairs), as it is capped at the number of tweets included in that dataset (which is 1.4K).

4.3.2 Method

General Scheme As a base for my models, I use Sentence-BERT (SBERT). It uses a Siamese network trained with a Transformer (Vaswani et al., 2017) encoder to obtain sentence-level embeddings. I keep the base architecture proposed by Reimers and Gurevych (2019), but I use additional features, training tricks, and losses described in the next sections. The input is a pair of a tweet and fact-checking article, which I encode as follows:

- User Tweet: [CLS] *Tweet Text* [SEP]
- Verifying article: [CLS] *Title* [SEP] *Subtitle* [SEP] *Verified Claim* [SEP]

I train the models using the multiple negatives ranking (MNR) loss (Henderson et al., 2017) (see Eq. 4.3), instead of the standard cross-entropy (CE) loss, as the datasets contain only positive (i.e., matching) pairs. Moreover, I propose a new

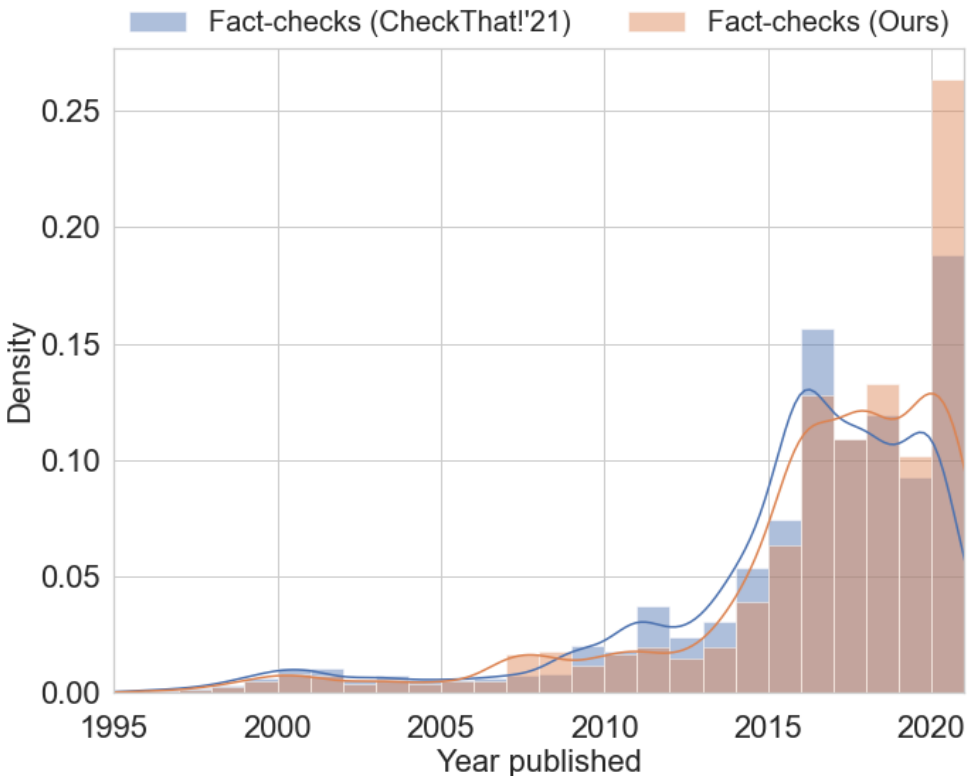


Figure 4.5: Histogram of the year of publication of the Snopes articles included in *CrowdChecked* (my dataset) vs. those in *CheckThat '21*.

variant of the MNR loss that accounts for the noise in the dataset, as described in detail in Section 4.3.2.

Enriched Scheme In the enriched scheme of the model, I adopt the pipeline proposed in the best-performing system from the CheckThat '21 competition (Chernyavskiy et al., 2021). Their method consists of independent components for assessing lexical (TF.IDF-based) and semantic (SBERT-based) similarities. The SBERT models use the same architecture and input format as described in the *'General Scheme'* above. However, Chernyavskiy et al. (2021) use an ensemble of models, i.e., instead of calculating a single similarity between the tweet and the joint title/subtitle/verified claim, the similarities between the tweet and the claim, the joint title/claim, and the three together are obtained from three models, one using on TF.IDF and one using SBERT, for each combination. These similarities are combined via a re-ranking model (see Section 4.3.2). In my experiments, the TF.IDF and the model ensembles are included only in the models with re-ranking.

Shuffling and Temperature I adopt a temperature parameter (τ) in the MNR loss. I also make it trainable in order to stabilize the training process as suggested in (Chernyavskiy et al., 2022). This forces the loss to focus on the most complex and important examples in the batch. Moreover, this effect is amplified after each epoch by an additional data shuffling that composes batches from several groups

of the most similar examples. This shuffling, in turn, increases the temperature significance. The nearest neighbors forming the groups are found using the model predictions. More detail about the training and the models themselves can be found in (Chernyavskiy et al., 2021).

Training with Noisy Data

Self-Adaptive Training To account for possible noise in the distantly supervised data, I propose a new method based on a self-adaptive training (Huang et al., 2020), which was introduced for classification tasks and the CE loss; however it needs to be modified in order be used with the MNR loss. I iteratively refurbish the labels y using the predictions of the current model starting after an epoch of choice, which is a hyper-parameter:

$$y^r \leftarrow \alpha \cdot y^r + (1-\alpha) \cdot \hat{y},$$

where y^r is the current refurbished label ($y_r = y$ initially), $\hat{y}$ is the model prediction, and α is a momentum hyper-parameter (I set α to 0.9).

Since the MNR loss operates with positive pairs only (it does not operate with labels), to implement this approach, I had to modify the loss function. Let $\{c_i, v_i\}_{1..m}$ be the batch of input pairs, where m is the batch size, $C, V \in \mathbb{R}^{m \times h}$ are the matrices of embeddings for the tweets and for the fact-checking articles (h is the embeddings' hidden size), and C, V are normalized to the unit hyper-sphere (I use cosine similarity), then:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m} {y^r}_i \Big(\frac{c_i^T v_i}{\tau} - \log \sum_{j=1}^{m} \exp(\frac{c_i^T v_j}{\tau}) \Big) \tag{4.3}$$

If I set $y_i^r = 1$, then Eq. 4.3 resembles the MNR loss definition. The parameter τ is the temperature, discussed in Section 4.3.2 *Shuffling and Temperature*.

Weighting In the self-adaptive training approach, Huang et al. (2020) introduce weights $w_i = \max_{j \in \{1,..,L\}} t_{i,j}$, where t_i is the corrected one-hot encoded target vector in a classification task with L classes. The goal is to ensure that noisy labels will have a lower influence on the training process compared to correct labels. Instead of a classification task with one-hot target vectors $t_{i,j}$, here we have real targets y_i^r. Therefore, I take these probabilities as weights: $w_i = y_i^r$. After applying both modifications with the addition of labels and weights, the impact of each training example is proportional to the square of the corrected label, i.e., in Eq. 4.3 y_i^r is now squared.

Re-ranking

Re-ranking has shown major improvements for detecting previously fact-checked claims (Shaar et al., 2020, 2021; Mihaylova et al., 2021; Chernyavskiy et al., 2021), and thus I include it as part of my model. In particular, I adopt the re-ranking

Dataset	Data Split	Threshold	Tweet-Article Pairs
CrowdChecked (My Dataset)	Train	-	332,660
	Train *Jaccard*	0.30	27,387
		0.40	12,555
		0.50	4,953
	Train *Cosine*	0.50	48,845
		0.60	26,588
		0.70	11,734
		0.80	3,496
CheckThat '21	Train	-	999
	Dev	-	199
	Test	-	202

Table 4.11: Statistics about my collected datasets in terms of tweet–verifying article pairs.

procedure from Chernyavskiy et al. (2021). It uses a LambdaMART (Wu et al., 2010) model. The inputs are the reciprocal ranks (position in the ranked list of claims) and the predicted relevance scores (2 factors) based on the scores of the TF.IDF and SBERT models (2 models), between the tweet and the claim, claim+title, and claim+title+subtitle (3 combinations), for a total of twelve features in the ensemble and four in the single model.

4.3.3 Experiments

In this section, I describe my experimental setup and I present my experimental results. The training procedure and the hyper-parameters are in Appendix A.1.1, and the baselines are in Section 4.3.3.

Experimental Setup

Datasets Table 4.11 shows statistics about the data split sizes for CrowdChecked and CheckThat '21. I use these splits in my experiments, albeit sometimes mixed together.

The first group (CrowdChecked) is the data splits obtained from distant supervision. As the positive pairs are annotated with distant supervision and not by humans, I include them as part of the training set. Each shown split is obtained using a different similarity measure (Jaccard or Cosine) or threshold. From the total number of 332K collected tweet–article pairs in CrowdChecked, I end up with subsets of sizes between 3.5K and 49K examples.

The second group describes the CheckThat '21 dataset. I preserve the original training, development, and testing splits. In each of my experiments, I validate and

test on the corresponding subsets from the CheckThat '21, while the training set can be a mix with CrowdChecked.

Baselines and State-of-the-Art

- **Retrieval** (Shaar et al., 2021) uses an information retrieval model based on BM25 (Robertson and Zaragoza, 2009) that ranks the list of fact-checking articles based on the relevance score between its *{'claim', 'title'}* and the tweet's text.
- **Sentence-BERT** is a bi-encoder model based on Sentence-BERT fine-tuned for detecting previously fact-checked claims using MNR loss. The details are in Section 4.3.2, *General Scheme*.
- **DIPS** (Mihaylova et al., 2021) adopts a Sentence-BERT model that computes the cosine similarity for each pair of an input tweet and a verified claim (article). The final ranking is made by passing a sorted list of cosine similarities to a fully-connected neural network.
- **NLytics** (Pritzkau, 2021) uses a RoBERTa-based model optimized as a regression function obtaining a direct ranking for each tweet-article pair.
- **Aschern** (Chernyavskiy et al., 2021) combines TF.IDF with a Sentence-BERT (ensemble with three models of each type). The final ranking is obtained from a re-ranking LambdaMART model.

Metrics For my evaluation, I adopt the ranking measures used in the CheckThat '21 competition. In particular, I calculate the mean reciprocal rank (MRR) mean average precision (MAP@K) and Precision@K, for $K \in \{1, 3, 5, 10\}$. All the models are optimized for MAP@5, as was in the CLEF-2021 CheckThat! lab subtask 2A.

Experimental Results

Below, I present experiments that (*i*) aim to analyze the impact of training with the distantly supervised data from CrowdChecked, and (*i*) to further improve the state-of-the-art (SOTA) results using modeling techniques to better leverage the noisy data points (see Section 4.3.2). In all my experiments, I evaluate the model on the development and on the testing sets from CheckThat '21 (see Table 4.11), and I train on a mix with *CrowdChecked*. The reported results for each experiment (for each metric) are averaged over three runs using different seeds.

Threshold Selection Analysis My goal here is to evaluate the impact of using distantly supervised data from CrowdChecked. In particular, I train an SBERT baseline, as described in Section 4.3.2, using four different training datasets: (*i*) the training data from CheckThat '21, (*ii*) training data from *CrowdChecked*, (*iii*) pre-training

Model	MRR	P@1	MAP@5
Baselines (CheckThat '21)			
Retrieval (Shaar et al., 2021)	76.1	70.3	74.9
SBERT (CheckThat '21)	79.96	74.59	79.20
CrowdChecked (My Dataset)			
SBERT (jac > 0.30)	81.50	76.40	80.84
SBERT (cos > 0.50)	81.58	75.91	81.05
(Pre-train) CrowdChecked, (Fine-tune) CheckThat '21			
SBERT (jac > 0.30, Seq)	**83.76**	**78.88**	**83.11**
SBERT (cos > 0.50, Seq)	82.26	77.06	81.41
(Mix) CrowdChecked and CheckThat '21			
SBERT (jac > 0.30, Mix)	83.04	78.55	82.30
SBERT (cos > 0.50, Mix)	82.12	76.57	81.38

Table 4.12: Evaluation on the CheckThat '21 testing set. In parenthesis is name of the training split, i.e., *Jac*card or *Cos*ine selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two. The highest results are in **bold**.

on data from *CrowdChecked* and then fine-tuning on CheckThat '21, (*iv*) mixing the data from both datasets.

Table 4.12 shows the results grouped based on training data used. In each group, I include the two best-performing models. We see that all SBERT models outperform the Retrieval baseline by 4–8 points absolute MAP@5. Interestingly, training only on distantly supervised data is enough to outperform the SBERT trained on the CheckThat '21 by more than 1.5 MAP@5 points. Moreover, the performance of both data labeling strategies (i.e., Jaccard and Cosine) is relatively close, suggesting comparable amount of noise in the two datasets.

Next, I train on combined data from the two datasets. Unsurprisingly, both mixing the data and training on the two datasets sequentially (CrowdChecked $\rightarrow$ CheckThat '21) yields additional improvement compared to training on a single dataset. I observe the best result when the model is first pre-trained on the *(jac > 0.3)* subset of CrowdChecked, and then fine-tuned on CheckThat '21. This combination gains 2 points absolute in all metrics, compared to *SBERT (CrowdChecked)* and 4 points compared to *SBERT (CheckThat '21)*. Nevertheless, I must note that pre-training with the *Cosine similarly (cos > 0.50)* did not yield such sizable improvements as the ones when using Jaccard. I attribute this, on one hand, to the higher expected noise in the data according to the manual annotations (see Section 4.3.1), and on the other hand, to these examples being annotated by a similar model, and thus presumably easy for it.

Further, I analyze the impact of choosing different thresholds for the distant supervision approaches. Figure 4.6 shows the change of MAP@5 for each data labeling strategy. On the left part of the figure, in the interval [0.3–0.5], are shown

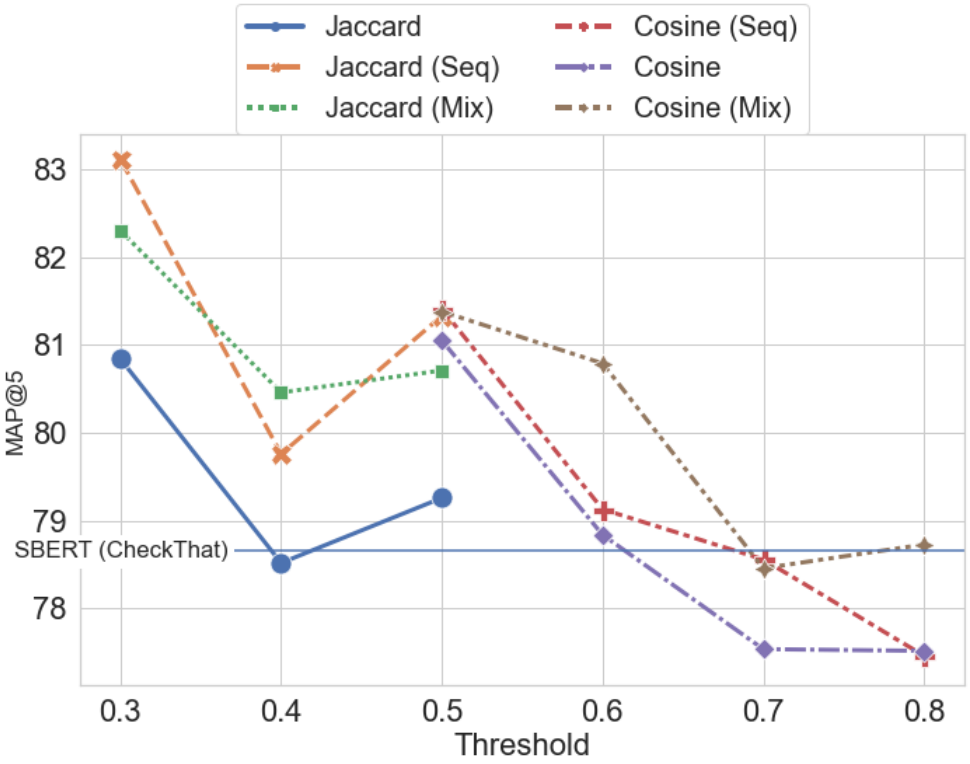


Figure 4.6: MAP@5 for different thresholds and distant supervision approaches. The *Jaccard* and the *Cosine* models are trained only on CrowdChecked, while (*Seq*) and (*Mix*) were trained also on CheckThat '21.

the results of the Jaccard-based data labeling strategy, and on the right ([0.5–0.8]) – the Cosine strategy. Once again, the models trained on the data selected using Jaccard similarity perform similarly or better as the *SBERT (CheckThat '21)* model (blue solid line). On the other hand, the Cosine-based selection outperforms the baseline only in small thresholds ≤ 0.6. These observations are in favor of the hypothesis that the highly ranked pairs from the fine-tuned SBERT model are easy examples, and do not bring much signal to the model over the CheckThat '21 data, whereas the Jaccard ranked ones significantly improve the model's performance. Nonetheless, we see similar performance when training with data from the lowest two thresholds for the two similarities (without data mixing), which suggests that these subsets have similar characteristics.

Adding more distantly supervised data is beneficial for the model, regardless of the strategy. The only exception is the drop in performance when I decrease the Jaccard threshold from 0.5 to 0.4. I attribute this to the quality of the data in that bracket, as the examples with lower similarity are expected to add more noise, however the results improve drastically at the next threshold (adding x2 more examples). The latter suggests that the model was able to generalize better from the new data. There is no such drop in the Cosine strategy. I explain this with expectation that noise increases proportionally to the decrease in model confidence.

Model	**MAP@5**	
	Dev	Test
DIPS (Mihaylova et al., 2021)	93.6	78.7
NLytics (Pritzkau, 2021)	-	79.9
Aschern (Chernyavskiy et al., 2021)	94.2	88.2
SBERT (jac > 0.30, Mix)	90.0	82.3
+ shuffling & trainable temp.	92.4	82.6
+ self-adaptive training (Eq. 4.3)	92.6	83.6
+ loss weights	*92.7*	*84.3*
+ TF.IDF + Re-ranking	93.1	89.7
+ TF.IDF + Re-ranking (ens.)	**94.8**	**90.3**

Table 4.13: Results on CheckThat '21 (dev and test). I compare my model and its components (added sequentially) to the state of the art. The best results are in **bold**.

Finally, I report the performance of each model both on the development and on the testing datasets in Section 4.3.3, Tables 4.14 and 4.15.

Modeling Noisy Data I explore the effects of the proposed changes to the SBERT training approach: (*i*) shuffling and training temperature, (*ii*) data-related modification of the MNR loss for self-adaptive training with weights. I use the *(jac > 0.30, mix)* approach in my experiments, as the baseline SBERT models achieved the highest scores on the dev set (Table 4.14). In Table 4.13, I ablate each of these modifications by adding them iteratively to the baseline SBERT model.

First, we can see that adding a special shuffling procedure and a trainable temperature (τ) improves the MAP@5 by 2 points on the dev set and 0.3 on the test set. Next, we see a sizable improvement of 1 point MAP@5 on the test set, when using the self-adaptive training with MNR loss. Moreover, an additional 0.7 points comes from adding weights to the loss, arriving at 84.3 MAP@5. These weights allow the model to give higher importance to the less noisy data during the training process. Here, I must note that for these two ablations the improvements on the development set are diminishing. I attribute this to its small size (199 examples) and the high values of MAP@5. Finally, note that my model, without using re-ranking, outperforms all state-of-the-art models, except Aschern, by more than 4.5 points on the testing dataset.

On the last two rows of Table 4.13, I present the results of my model that includes all proposed components, in combination with TF.IDF features and the LambdaMART re-ranking, described in Section 4.3.2. Here, I must note that the model is trained on a part of the CheckThat '21 training pool (80%) – the other part is used to train the re-ranking model. The full setup boosts the model's MAP@5 up to *89.7* when using a single model of the TF.IDF and SBERT (using the title/subtitle/claim as inputs, same as SBERT). With the ensemble architecture

Model	MRR	P@1	MAP@5
Baselines (CheckThat '21)			
Retrieval (Shaar et al., 2021)	76.1	70.3	74.9
SBERT (CheckThat '21)	87.97	84.92	87.45
CrowdChecked (My Dataset)			
SBERT (cos > 0.50)	88.20	85.76	87.80
SBERT (cos > 0.60)	87.21	84.25	86.69
SBERT (cos > 0.70)	86.18	83.08	85.76
SBERT (cos > 0.80)	83.57	80.40	82.93
SBERT (jac > 0.30)	88.01	85.09	87.61
SBERT (jac > 0.40)	87.26	84.76	86.80
SBERT (jac > 0.50)	86.53	83.42	86.13
(Pre-train) CrowdChecked, (Fine-tune) CheckThat '21			
SBERT (cos > 0.50, Seq)	89.92	87.60	89.49
SBERT (cos > 0.60, Seq)	89.56	87.27	89.20
SBERT (cos > 0.70, Seq)	88.70	85.59	88.36
SBERT (cos > 0.80, Seq)	88.42	85.26	88.03
SBERT (jac > 0.30, Seq)	90.21	87.44	89.69
SBERT (jac > 0.40, Seq)	89.64	86.77	89.25
SBERT (jac > 0.50, Seq)	89.44	86.26	89.03
(Mix) CrowdChecked and CheckThat '21			
SBERT (cos > 0.50, Mix)	89.47	86.77	88.99
SBERT (cos > 0.60, Mix)	88.54	85.76	87.98
SBERT (cos > 0.70, Mix)	87.71	84.92	87.18
SBERT (cos > 0.80, Mix)	88.40	85.26	87.97
SBERT (jac > 0.30, Mix)	90.41	87.94	90.00
SBERT (jac > 0.40, Mix)	89.82	86.60	89.48
SBERT (jac > 0.50, Mix)	88.71	85.26	88.31

Table 4.14: Evaluation on the CheckThat '21 **development** set. In parenthesis is name the training split, i.e., Jaccard (*jac*) or Cosine (*cos*) data selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two datasets.

(re-ranking based on the scores of three TF.IDF and three SBERT models), I reach my best results of *90.3* on the test set (adding 1.7 MAP@5 on dev, and 0.6 on test), outperforming the previous state-of-the-art approach (*Aschern*, 88.2) by 2 points MAP@5, and more than 11 compared to the second best model (*NLytics*, 79.9). This improvement corresponds to the observed gain over the SBERT model without re-ranking. Nevertheless, the change in the strength of the factors in LambdaMART is less. The TF-IDF models still have high importance for re-ranking – a total of 41% compared to 42.8% reported in Chernyavskiy et al. (2021). Here, I have a decrease mainly due to an increase of the importance of the reciprocal rank factor from 18.8% to 20.2% of the SBERT model that selects candidates. The strength of other factors remains almost unchanged.

Results on the Development Set Here, I present the expanded results for my experiments described in Section 4.3.3. Tables 4.14 and 4.15 include the results for

Model	MRR	Precision @1	Precision @3	Precision @5	Precision @10	Precision @20	MAP @1	MAP @3	MAP @5	MAP @10	MAP @20
Baselines (CheckThat '21)											
Retrieval (Shaar et al., 2021)	76.1	70.3	26.2	16.4	8.8	4.6	70.3	74.1	74.9	75.7	75.9
SBERT (CheckThat '21)	79.96	74.59	27.89	17.19	8.96	4.61	74.59	78.66	79.20	79.66	79.83
CrowdChecked (My Dataset)											
SBERT (cos > 0.50)	81.58	75.91	28.60	17.76	9.04	4.67	75.91	80.36	81.05	81.27	81.48
SBERT (cos > 0.60)	79.71	74.75	27.39	16.96	8.86	4.59	74.75	78.25	78.84	79.38	79.61
SBERT (cos > 0.70)	78.27	72.28	27.61	17.10	8.89	4.53	72.28	76.95	77.54	78.01	78.12
SBERT (cos > 0.80)	78.39	72.94	27.34	16.83	8.81	4.55	72.94	77.04	77.52	78.08	78.28
SBERT (jac > 30)	81.50	76.40	28.49	17.43	8.94	4.65	76.40	80.45	80.84	81.14	81.38
SBERT (jac > 40)	79.45	74.42	27.34	16.93	8.89	4.65	74.42	77.92	78.52	79.08	79.33
SBERT (jac > 50)	79.96	74.75	27.89	17.29	8.94	4.60	74.75	78.63	79.26	79.63	79.81
(Pre-train) CrowdChecked, (Fine-tune) CheckThat '21											
SBERT (cos > 0.50, Seq)	82.26	77.06	28.27	17.62	9.26	4.76	77.06	80.64	81.41	81.99	82.18
SBERT (cos > 0.60, Seq)	80.13	75.41	27.45	17.00	8.94	4.65	75.41	78.55	79.13	79.76	79.99
SBERT (cos > 0.70, Seq)	79.27	73.43	27.72	17.33	8.94	4.58	73.43	77.78	78.56	78.94	79.09
SBERT (cos > 0.80, Seq)	78.32	72.77	27.17	16.93	8.89	4.58	72.77	76.71	77.41	77.98	78.15
SBERT (jac > 0.30, Seq)	83.76	78.88	28.93	17.82	9.21	4.71	78.88	82.59	83.11	83.49	83.63
SBERT (jac > 0.40, Seq)	80.69	75.25	27.83	17.33	9.09	4.69	75.25	79.04	79.76	80.34	80.57
SBERT (jac > 0.50, Seq)	81.99	76.90	28.16	17.76	9.13	4.69	76.90	80.34	81.33	81.70	81.88
(Mix) CrowdChecked and CheckThat '21											
SBERT (cos > 0.50, Mix)	82.12	76.57	28.55	17.59	9.13	4.68	76.57	80.86	81.38	81.82	82.00
SBERT (cos > 0.60, Mix)	81.45	76.40	28.27	17.43	8.96	4.61	76.40	80.25	80.79	81.14	81.31
SBERT (cos > 0.70, Mix)	79.08	73.10	27.83	17.33	8.89	4.57	73.10	77.72	78.46	78.77	78.95
SBERT (cos > 0.80, Mix)	79.73	74.75	27.56	17.00	9.06	4.62	74.75	78.22	78.73	79.46	79.59
SBERT (jac > 0.30, Mix)	83.04	78.55	28.66	17.52	9.11	4.69	78.55	81.93	82.30	82.75	82.94
SBERT (jac > 0.40, Mix)	81.18	74.59	28.55	17.72	9.14	4.74	74.59	79.79	80.46	80.85	81.10
SBERT (jac > 0.50, Mix)	81.56	76.73	28.22	17.36	9.03	4.71	76.73	80.23	80.71	81.19	81.45

Table 4.15: Evaluation on the CheckThat '21 **testing** set. In parenthesis is name the training split, i.e., Jaccard (*jac*) or Cosine (*cos*) data selection strategy, *(Seq)* first training on CrowdChecked and then on CheckThat '21, *(Mix)* mixing the data from the two datasets.

the *threshold selection analysis* experiments on the development set, and testing set, respectively. In Table 4.15 corresponds to Table 4.12 in the main paper, and includes all metrics and for all thresholds (shown in Figure 4.6). Next, the results from my *Modeling Noisy Data* experiments are in Table 4.16, which corresponds to Table 4.13 in the main paper. In all tables I use the same notation and grouping as in their corresponding table.

4.3.4 Discussion

My proposed distant supervision data selection strategies show promising results, achieving SOTA results on the CheckThat '21. Nonetheless, I am not able to identify all matching pairs in the list of candidates in CrowdChecked. Hereby, I try to estimate their expected number using the statistics from the manual annotations,[8] shown in Tables 4.9,4.10.

In particular, I estimate it by multiplying the fraction of correct pairs in each similarity bin by the number of examples in this bin. Based on cosine similarity, I

[8]Due to the small number of annotated examples the variance in the estimates is large.

Model	MRR	Precision @1	Precision @3	Precision @5	Precision @10	MAP @1	MAP @3	MAP @5	MAP @10
DIPS (Mihaylova et al., 2021)	79.5	72.8	28.2	17.7	9.2	72.8	77.8	78.7	79.1
NLytics (Pritzkau, 2021)	80.7	73.8	28.9	17.9	9.3	73.8	79.2	79.9	80.4
Aschern (Chernyavskiy et al., 2021)	88.4	86.1	30.0	18.2	9.2	86.1	88.0	88.3	88.4
SBERT (jac > 0.30, Mix)	83.0	78.6	28.7	17.5	9.1	78.6	81.9	82.3	82.8
+ shuffling & trainable temp.	83.2	77.7	29.1	17.8	9.1	77.7	82.2	82.6	82.9
+ self-adaptive training (Eq. 4.3)	84.2	78.7	29.3	18.1	9.3	78.7	83.0	83.6	83.9
+ loss weights	84.8	79.7	29.5	18.2	9.3	79.7	83.7	84.3	84.6
+ TF.IDF + Re-ranking	89.9	86.1	30.9	18.9	9.6	86.1	89.2	89.7	89.8
+ TF.IDF + Re-ranking (ens.)	90.6	87.6	30.7	18.8	9.5	87.6	89.9	90.3	90.4

Table 4.16: Results on the CheckThat '21 **testing** set. I compare my model and its components (added sequentially) to state-of-the-art approaches.

estimate that out of the 332,600 pairs, the matching pairs are approximately 90,170 (27.11%). Further, based on the Jaccard distribution, I estimate that 14.79% of all tweet-conversation (root of the conversation), and 22.23% tweet–reply (the tweet before the current in the conversation) pairs are expected to match, or nearly 61,500 examples, assuming that the number of conversations and replies is equal.[9]

My experiments show that the models can effectively account for the noise in the training data. Both the self-adaptive training and the additional weighing in the loss function (described in Section 4.3.2), gain 1 additional point MAP@5 each. These results suggest that further investigation of the potential of learning from noisy labels (Han et al., 2018; Wang et al., 2019; Song et al., 2020,; Zhou and Chen, 2021) and utilizing all examples in CrowdChecked, can improve the results even more. Moreover, I argue that incorporating the negative examples (non-matching pairs) from CrowdChecked in the training objective can be beneficial for the models (Lu et al., 2021; Thakur et al., 2021).

4.4 Summary

In this chapter, I studied two directions for curating answers from external knowledge sources, namely: (*i*) zero-shot transfer from a rich- to a low-resource language for answer selection from a list of candidates based on a set of retrieved evidence contexts from an external knowledge base, and (*ii*) answer retrieval from a pool of explanations, i.e., previously written long-form answers such as documents or articles.

First, I studied the task of multiple-choice reading comprehension for low-resource languages, using a newly collected Bulgarian corpus with 2,633 questions from matriculation exams for twelfth grade in history and biology, and online exams in history without explanatory contexts. In particular, I designed an end-to-end

[9] In practice, there are more replies than conversations.

approach, on top of a multilingual BERT model (Devlin et al., 2019), which I fine-tuned on large-scale English reading comprehension corpora, and open-domain commonsense knowledge sources (Wikipedia). My main experiments evaluated the model when applied to Bulgarian in a zero-shot fashion. The experimental results found additional pre-training on the English RACE corpus to be very helpful, while pre-training on Slavic languages to be harmful, possibly due to catastrophic forgetting. Paragraph splitting, *n*-grams, stop-word removal, and stemming further helped the context retriever to find better evidence passages, and the overall model to achieve accuracy of up to 42.23%, which is well above the baselines of 24.89% and 29.62%.

Next, I presented CrowdChecked, a large-scale dataset for detecting previously fact-checked claims, with more than 330,000 pairs of tweets and corresponding fact-checking articles posted by crowd fact-checkers. I further investigated two techniques for labeling the tweet–article pairs using distance supervision, based on Jaccard similarity and the predictions from a neural network model resulting in training sets of 3.5K–50K examples. I also proposed an approach for training from noisy data using self-adaptive learning and additional weights in the loss function. Furthermore, I exhibit the utility of my data, which yielded sizable performance gains of four points in terms MRR, P@1, and MAP@5 over strong baselines trained on manually annotated data (Shaar et al., 2021). Finally, I demonstrated improvements over the state of the art on the CheckThat '21 dataset by two points, achieving MAP@5 of 90.3, when using the proposed dataset and pipeline.

Chapter 5

Advanced Conversation

This chapter explores advanced conversational methods that go beyond single language and individual models. In Section 5.3, I discuss end-to-end generative models. In contrast to the models discussed in previous chapters, these methods should allow the agent to handle the dialogue and to produce new answers that are unseen so far in the conversation, without depending on external sources or NLU components.

In Section 5.4, I propose a novel approach for selecting the next utterance in the conversation from a set of candidates obtained from multiple sources, e.g., generated using sequence-to-sequence models or retrieved from a knowledge base. I evaluate the proposed approaches using a large-scale dataset collected from a real-world customer support conversations in social media (Twitter) between companies and their peers. The dataset is described in detail in Section 5.2.

Finally, in Section 5.5 I study methods that go beyond single language and zero-shot learning. In particular, I introduce a new dataset for multiple-choice question answering covering sixteen language from eight language families. Moreover, I use this dataset to evaluate the capabilities of recent state-of-the-art multilingual models for cross-lingual transfer. This section develops on and extends further some of the ideas presented in Chapter 4, Section 4.2.

This chapter is mainly based on

- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards Automated Customer Support. In *Proceedings of the 18th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, AIMSA '18, pages 48–59, Varna, Bulgaria
- Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019b. Machine Reading Comprehension for Answer Re-Ranking in Customer Support Chatbots. *Information*, 10(3)
- Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020b. EXAMS: A Multi-subject High School Examinations Dataset for Cross-lingual and Multilingual Question Answering.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 5427–5444, Online

5.1 Introduction

Task-oriented dialogue agents are highly effective in serving a user request such as 'reserve a table' or 'book a flight' they follow a finite-state dialog flow, with pre-defined structure of the conversation, and iteratively fill a set of pre-defined slots until the agent has all the needed information to complete the task (Gao et al., 2019). However, often the agents produce utterances that are based on a limited number of discrete templates that leads to less natural experience compered to human-to-human conversation(Følstad and Skjuve, 2019). Moreover, many agents cannot not effectively engage into a long, open-domain conversation, forcing their users to either respond passively or lead the dialogue constantly (Hardy et al., 2021). On one hand, this is due to the limited number of topics that they cover (Mrkšić et al., 2017; Hung et al., 2022), on the other they lack of personality which can lead to inconsistent answers and writing style. All of these factors increase the probability of dialogue breakdowns (Higashinaka et al., 2016).

Recent advances in neural networks for NLP and the rise of the large pre-trained Transformers, especially models that are trained specifically to generate sequences (Vaswani et al., 2017; Radford et al., 2018, 2019; Raffel et al., 2020; Lewis et al., 2020) are a step towards end-to-end conversational agents. In theory, these models should be able to perform multi-turn open-domain dialogues without the need of pre-defined scenario or querying external knowledge sources (Zhang et al., 2020; Roller et al., 2021; Xu et al., 2022). Moreover, they can further learn from the conversations they engage in with the users, in addition to injected feedback from domain experts (e.g., customer support agents) (Li et al., 2017; Hancock et al., 2019; Shuster et al., 2022). Nonetheless, here I must note that this also raises a lot of ethical and practical concerns, as a chatbots can be biased to produce inappropriate and harmful dialogue acts, not only by exploits from malicious actors (Vincent, 2016; Hancock et al., 2019; Vanderlyn et al., 2021),but also from the induced biases learned during their extensive pre-training (Buolamwini and Gebru, 2018; Bender et al., 2021).

While end-to-end models tend to produce fluent and fairly consistent responses, another major limitation that prevent their practical applications is that that they suffer from the risk of hallucination (Dziri et al., 2021; Shuster et al., 2021; Dziri et al., 2022), i.e., producing factually invalid statements. That said, involving other strategies (sources) to obtain the next turn can mitigate their limitations and improve the overall experience (Ouchi and Tsuboi, 2016; Qiu et al., 2017; Cui et al., 2017; Clarke et al., 2022), e.g., retrieving texts from previous conversations or template-generated utterances are factual and more concise but less engaging

compared to end-to-end models. Therefore, in this chapter I explore methods for multi-source response selection in the domain of customer support conversations in Social Media.

Finally, more and more companies, independent of their size and market, provide services for a global audience, and thus offering localization and customer support in more than language. In this chapter, I explore the abilities of state of the art multilingual models for cross-lingual transfer (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020; Wang et al., 2020; Liu et al., 2020; Xue et al., 2021; Soltan et al., 2022). However, training monolingual models for each language is often infeasible but also limits the conversations that the models are exposed to. Recent Transformer-based multilingual models had shown an impressive performance on zero-shot transfer between languages. Furthermore, a strong indication of the importance of the task is the recent development multilingual dataset for conversational agents (Razumovskaia et al., 2022; Ding et al., 2022; Hung et al., 2022; FitzGerald et al., 2022).

The contributions of this chapter are as follows:

- End-to-End dialogue agents:
  - I study end-to-end automating customer support on Twitter using two types of models: (*i*) retrieval-based (IR with BM25), and (*ii*) based on generative neural networks .
  - I provide new data splits of a customer support dataset, based on the timestamp of the post in order to simulate a real-world scenario.
  - I explore two types of unsupervised evaluation measures that does not need additional human judgments: (*i*) word-overlap (BLEU@2 and ROUGE-L), and (*ii*) semantics (Embedding Average, Greedy Matching, and Vector Extrema).
  - I show that neural-based models outperform retrieval-based ones in all evaluation metrics.
- Multi-source response selection:
  - I propose a novel framework for re-ranking response candidates for conversational agents based on techniques from the domain of machine reading comprehension.
  - I design a new negative sampling procedure and incorporate it into a state-of-the-art question answering model (QANet).
  - The proposed re-ranking model shows sizable improvements over single models on the customer support dataset by selecting the most relevant from their answers.

- Multilingual and cross-lingual modeling:
  - I advance the task of science QA with multilingual and cross-lingual evaluations.
  - I collect a new challenging dataset $E\chi\alpha\mu s$ from multilingual high school examinations, which offers several advantages over existing datasets: (*i*) it covers various domains, (*ii*) it is nearly three times larger than pre-existing Science QA datasets, (*iii*) it extends multilingual QA tasks to more languages, (*iv*) the questions are written by experts, rather than translated or crowdsourced, (*v*) the questions are harder since they are from matriculation exams rather than 4-8th grade.
  - I use fine-grained evaluation – per subject and per language – which yields more precise comparison between models.
  - I perform extensive experiments and analysis using top-performing multilingual models (mBERT, XLM-R), and I show that $E\chi\alpha\mu s$ offers several challenges that such models would need to overcome in the future, including multi-lingual and cross-lingual knowledge retrieval, aggregation, and reasoning, among others.

5.2 Dataset for Customer Support Conversations

Overall, data and resources that could be used to train a customer support chatbot are very scarce, as companies keep conversations locked on their own closet, proprietary support systems. This is due to customer privacy concerns and to companies not wanting to make public their know-how and the common issues about their products and services. An extensive 2015 survey on available dialog corpora by Serban et al. (2018) found no good publicly available dataset for real-world customer support.

This situation has changed as a new open dataset, named *Customer Support on Twitter*, was made available on Kaggle.[1] It is a large corpus of recent tweets and replies, which is designed to support innovation in natural language understanding and conversational models, and to help study modern customer support practices and impact. The dataset contains 3M tweets from 20 big companies such as Amazon, Apple, Uber, Delta, and Spotify, among others. See Figure 5.1 for detail.

As customer support topics from different organizations are generally unrelated to each other, I focus only on tweets related to Apple support, which represents the second largest number of tweets in the corpus. This allows us to stay focused on a small range of topics that are related to a single company, a situation closer to a real-world scenario. I filtered all utterances that redirect the user

[1] https://www.kaggle.com/thoughtvector/customer-support-on-twitter

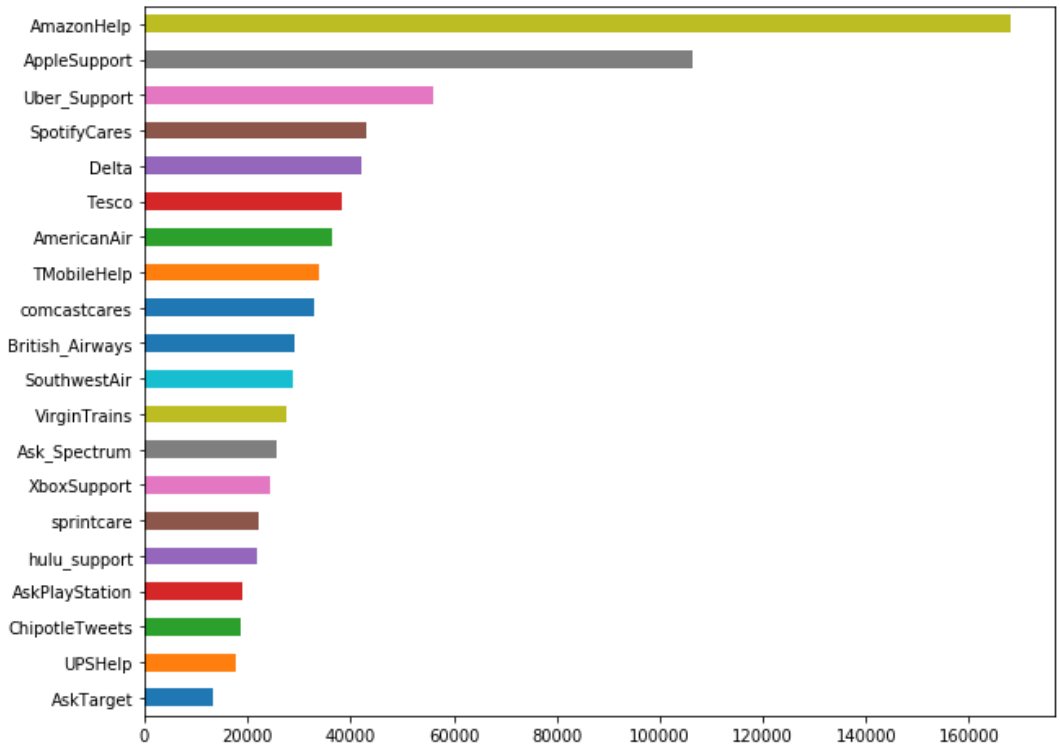


Figure 5.1: Number of user tweets with replies from customer support per company.

to another communication channel, e.g., direct messages, which are not informative for the model and only bring noise. Moreover, since answers evolve over time, I divided the dataset into a training and a testing part, keeping earlier posts for training and the latest ones for testing. I further excluded from the training set all conversations that are older then sixty days. For evaluation, I used dialogues from the last five days in the dataset, to simulate a real-world scenario for customer support. I ended up with a dataset of 49,626 dialog tuples divided in 45,582 for training and 4,044 for testing.

Tables 5.1 and 5.2 show some statistics about the dataset. In Table 5.1 we can see that the average number of turns per dialog is under three, which means that most of the dialogues finish after one answer from customer support. Table 5.2 shows the distribution of words in the user questions vs. the customer support answers. We can see that answers tend to be slightly longer, which is natural as replies by customer support must be extensive and helpful.

5.3 End-to-End Generative Agent

The rapid proliferation of mobile and portable devices has enabled a number of new products and services. Yet, it has also laid stress on customer support as users now also expect 24x7 availability of information about their orders, or answers to

Overall	
# words (in total)	26,140
Min # turns per dialog	2.00
Max # turns per dialog	106.00
Avg. # turns per dialog	2.6
Avg. # words in question	20.00
Avg. # words in answer	25.88
# dialogs tuples	49,626
Training set: # of dialogs	45,582
Testing set: # of dialogs	4,044

Table 5.1: Overall statistics about the dataset.

	Questions	**Answers**
Avg. # words	21.31	25.88
Min # words	1.00	3.00
25%	13.00	20.00
50%	20.00	23.00
75%	27.00	29.00
Max # words	136.00	70.00

Table 5.2: Statistics about the dataset.

basic questions such as 'Why is my Internet connection dead?' and 'What time is the next train from Sofia to Varna?'

Customer support has always been important to companies. Traditionally, it was offered primarily over the phone, but recently a number of alternative communication channels have emerged such as e-mail, social networks, forums/message boards, live chat, self-serve knowledge base, etc. As a result, it has become increasingly expensive for companies to maintain quality customer support services over a growing number of channels. First, they must find people that have both good language and communication skills. Second, each new employee must go through several training sessions before being able to operate in the target channel, which is inefficient and time-consuming. And finally, it is difficult to have employees available for customer support 24x7. Chatbots are especially fit for the task as they are automatic: fully or partially. Moreover, from a technological viewpoint, they are feasible as the domain they need to operate in is narrow. As a result, chit-chat is reduced to a minimum, and chatbots serve primarily as question-answering devices. Moreover, it is possible to train them on real-world chat logs. Here, I experiment with such logs from customer support on Twitter, and I compare two types of chatbots: (*i*) based on information retrieval (IR), and (*ii*) on neural question answering. I further explore semantic similarity measures since generic ones such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005), which come from machine translation or text summarization, are not well suited

for chatbots.

5.3.1 Method

Preprocessing

Since Twitter has its own specifics of writing in terms of both length[2] and style, standard text tokenization is generally not suitable for tweets. Therefore, I used a specialized Twitter tokenizer (Manning et al., 2014) to preprocess the data. Then, I further cleaned the data by replacing the shorthand entries, e.g., *'ll*, *'d*, *'re*, *'ve*, with the most likely literary form, e.g., *will*, *would*, *are*, *have*. I also replaced slang words, e.g., *'bout* and *'til*, with the standard words, e.g., *about* and *until*. Similarly, I replaced URLs with the special word <*url*>, all user mentions with <*user*>, and all hashtags with<*hashtag*>.

Moreover, I tried to mitigate the effect of missing context in long conversations by concatenating all previous turns to the current question. Finally, since Seq2seq models cannot be trained efficiently with a large vocabulary, I chose the top N words when building the model (see Section 5.3.2 for more details), and I replaced the instances of the remaining words with a special symbol <*unk*>.[3]

Information Retrieval

The IR approach can be defined as follows: given a user question q' and a list of pairs of previously asked questions and their answers $(Q, A) = \{(q_j, a_j)|j = 1, \ldots, n\}$, find the most similar question q_i in the training dataset that a user has previously asked and return the answer a_i that customer support has given to q_i. The similarity between q' and q_i can be calculated in various ways, but most commonly this is done using the cosine between the corresponding TF.IDF-weighted vectors.

$$a' = \arg\max_{(q_j, a_j)} sim(q', q_j) \tag{5.1}$$

Sequence-to-Sequence

My encoder uses a bidirectional recurrent neural network RNN based on LSTM Hochreiter and Schmidhuber (1997). It encodes the input sequence $x = (x_1, \ldots, x_n)$ and calculates a forward sequence of hidden states $(\overrightarrow{h_1}, \ldots, \overrightarrow{h_m})$ and also a backward sequence $(\overleftarrow{h_1}, \ldots, \overleftarrow{h_m})$. The decoder is a unidirectional LSTM-based RNN, and it predicts the output sequence $y = (y_1, \ldots, y_n)$. Each y_i is predicted using the recurrent state s_i, the previous predicted word y_{i-1}, and a context vector c_i.

[2] By design, tweets have been strictly limited to 140 characters; this constrain has been relaxed to 280 characters in 2017.

[3] In future work, I plan to try byte-pair encoding instead (Sennrich et al., 2016).

The latter is computed using an attention mechanism as a weighted sum over the encoder's output $(\overrightarrow{h_j}, \overleftarrow{h_j})$, as proposed by Bahdanau et al. (2015).

Transformer

The Transformer model was proposed by Vaswani et al. (2017), and it has shown very strong performance for machine translation, e.g., it achieved state-of-the-art results on WMT2014 data for English-German and English-French translation. Similarly to the Seq2seq model, the Transformer has an encoder and a decoder. The encoder is a stack of identical layers, based on multi-head self-attention and a simple position-wise fully connected network. The decoder is similar, but in addition to the two sub-layers in the encoder, it introduces a third sub-layer, which performs multi-head attention over the encoders' stack outputs. The main advantage of the Transformer model is that it can be trained significantly faster, as compared to recurrent or convolutional neural networks.

5.3.2 Experiments

I performed three experiments using the models described in Section 5.3.1. Below, each model is abbreviated by its architecture name from 5.3.2.

IR is based on ElasticSearch[4] (ES), as it provides out-of-the-box implementation of all the components I need. I fed the pre-processed training data into an index with English analyzer enabled, whitespace- and punctuation-based tokenization, and word 3-grams. For retrieval, I used the default BM25 algorithm (Robertson and Zaragoza, 2009), which is an improved version of TF.IDF. For all training questions and for all testing queries, I appended the previous turns in the dialog as context. Given a user question from the testing set, I returned the customer support answer for the top-ranked result from ES.

Seq2seq contains one bi-directional LSTM layer with 512 hidden units per direction (a total of 1,024). The decoder has two unidirectional layers connected directly to the bidirectional one in the encoder. The network takes as input words encoded as 200-dimensional embeddings. It is a combination of pre-trained GloVe (Pennington et al., 2014) vectors learned from 27B Twitter posts[5] for the known words, and a positional embedding layer, learned as model parameters, for the unknown words. The embedding layers for the encoder and for the decoder are not shared, and are learned separately. This separation is due to the fact that the words used in utterances by customers are very different compared to posts by the support. In my experiments, I used the top 8,192 words sorted by frequency for both the embedding and the output. Based on the statistics presented in Section 4.2.2, I chose to use 60 words (time-steps) for both the encoder and the decoder. I avoid overfitting

[4] https://www.elastic.co/products/elasticsearch
[5] https://nlp.stanford.edu/projects/glove/

by applying dropout (Srivastava et al., 2014) with keep probability of 0.8 after each recurrent layer. For the optimizer, I used Adam (Kingma and Ba, 2015) with a base value of 1.00×10^{-3} and an exponential decay of 0.99 per epoch.

Transformer is based on two identical layers for the encoder and for the decoder, with four heads for the self-attention. The dimensionality of the input and of the output is $d_{model} = 256$, and the inner dimensionality is $d_{inner} = 512$. The input consists of queries with keys of dimension $d_k = 64$ and values of dimension $d_v = 64$. The input and the output embedding are learned separately with sinusoidal positional encoding. The dropout is set to 0.9 keep probability. For the optimization, I use Adam with varying learning rate based on Eq. 5.2. The hyper-parameter choice was guided by the experiments described by the authors in the original Transformer paper (Vaswani et al., 2017).

$$lrate = d_{model}^{-0.5} \cdot \min\left(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5}\right) \tag{5.2}$$

Evaluation Measures

How to evaluate a chatbot is an open research question. As the problem is related to machine translation (MT) and text summarization (TS), which are nowadays also addressed using Seq2seq models, researchers have been using MT and TS evaluation measures such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), which focus primarily on word overlap and measure the similarity between the chatbot's response and the gold customer support answer to the user question. However, it has been argued (Liu et al., 2016; Lowe et al., 2017) that such word-overlapping measures are not very suitable for evaluating chatbots. Thus, I adopt three additional measures, which are more semantic in nature.[6]

The *embedding average* constructs a vector for a piece of text by taking the average of the word embeddings of its constituent words. Then, the vectors for the chatbot response and for the gold human one are compared using cosine similarity.

The *greedy matching* was introduced in the context of intelligent tutoring systems (Rus and Lintean, 2012). It matches each word in the chatbot output to the most similar word in the gold human response, where the similarity is measured as the cosine between the corresponding word embeddings, multiplied by a weighting term, which I set to 1, as shown in equation (5.3). Since this measure is asymmetric, I calculate it a second time, with arguments swapped, and then I take the average as shown in equation 5.4.

$$greedy(u_1, u_2) = \frac{\sum_{v \in u_1} weight(v) * \max_{w \in u_2} cos(v, w)}{\sum_{v \in u_1} weight(v)} \tag{5.3}$$

[6]Note that I do not use measures trained on the same data as advised by Liu et al. (2016).

	Word Overlap Measures	
	BLEU@2	ROUGE-L
IR - BM25	13.73	22.35
Seq2seq	**15.10**	**26.60**
Transformer	12.43	25.33

Table 5.3: Results based on word-overlap measures.

	Semantic Evaluation Measures		
	Embedding Average	Greedy Matching	Vector Extrema
IR - BM25	76.53	29.72	37.99
Seq2seq	**77.11**	**30.81**	**40.23**
Transformer	75.35	30.08	39.40

Table 5.4: Results based on semantic measures.

$$simGreedy(u_1, u_2) = \frac{greedy(u_1, u_2) + greedy(u_2, u_1)}{2} \tag{5.4}$$

The *vector extrema* was proposed by Forgues et al. (2014) for dialogue systems. Instead of averaging the word embeddings of the words in a piece of text, it takes the coordinate-wise maximum (or minimum), as shown in Eq. 5.5. Finally, the resulting vectors for the chatbot output and for the gold human one are compared using cosine.

$$extrema(u_i) = \begin{cases} \max u_i, & if \max u_i \geq |\min u_i| \\ \min u_i, & \text{otherwise} \end{cases} \tag{5.5}$$

Results

Table 5.3 shows the results for the three models I compare (IR, Seq2seq, and Transformer) when using word overlap measures such as BLEU@2, which uses unigrams and bigrams only, and ROUGE-L (Lin and Och, 2004), which uses longest common subsequence (LCS).

Table 5.4 shows the results for the same three systems, but using the above-described semantic evaluation measures, namely Embedding Average (with cosine similarity), Greedy Matching, and Vector Extrema (with cosine similarity). For all three measures, I used Google's pre-trained word2vec embeddings because they are not learned during training, which helps avoid bias, as has been suggested in (Liu et al., 2016; Lowe et al., 2017).

Question & Answer	**Response**
(1) my phones speaker just broke down on me thanks <user> do better <user> we are here to help. which iphone and ios version are you using? are you having a sound quality issue or is there no sound at all?	**IR:** <user> we can help with your device . do you notice the slowness when performing a particular activity ? **Seq2seq:** <user> that is not the experience we want you to have . which iphone model do you own ? **Transformer:** <user> thanks for reaching out to us . we are always happy to help . can you tell us which version of ios 11 you are using ?
(2) i am trying to type it not fuckin i.t, your product is absolute garbage. figure it out. <user> <user> <user> lets take a closer look at this to get your device working right. which model do you have and that is the exact os version number installed? have you tried any steps so far and when did this start happening?	**IR:** <user> thank you for reaching out . we want to help . to start , lets go to settings > general > about and see which version of ios your iphone is on please . **Seq2seq:** <user> we would be happy to look into that is going on . can you tell us more? **Transformer:** <user> heres what you can do to work around the issue until it is fixed in a future software update : <url>
(3) capaz que si actualizo ios me teléfono se convierta en ladrillo :(....ya no confío en sus productos <user> <user> we offer support via twitter in english. get help in spanish here: <url> or join <url>	**All models:** <user> we offer support via twitter in english get help in spanish here <url> or join <url>
(4) <user> excuse me , i am got a refund confirmation email but nothing happened . do i have to do anythings ? → <user> thanks for reaching out to us about your refund ! to clarify , was this a refund for a purchase in the itunes store ? or was this a product purchase ? <user> thanks for reply . i am purchased music in itunes store . <user> thanks. our itunes store support team would have the best information for you on that. you can contact them here: <url>	**IR:** <user> alright . our itunes support team will be able to look at this with you . please reach out to them here : <url> **Seq2seq:** <user> we are investigating and will update our system status page as more info becomes available <url> **Transformer:** <user> thanks for reaching out . we would recommend leaving that request on our feedback page : <url>

Table 5.5: Chatbot responses. The first column shows the original question and the gold customer support answer, while the second column shows responses by the models.

5.3.3 Discussion

The evaluation results show that *Seq2seq* performed best with respect to all five evaluation measures. For the group of semantic measures, it outperformed the other systems in terms of Embedding Average by +0.58, in terms of Greedy Matching by +0.73, and in terms of Vector Extrema by +0.83 (points absolute). Moreover, SeqSeq was also clearly the best model in terms of word-overlap evaluation measures, scoring 15.10 on BLEU@2 (+1.37 ahead of the second), and 26.60 on ROUGE-L (+1.27 compared to the second best system). The *Transformer* model was ranked second by three of the evaluation measures: Greedy Matching, Vector Extrema, and ROUGE-L. This was unexpected given the state-of-the-art results it achieved for neural machine translation. Higher Greedy Matching and Vector Extrema scores show that the Transformer was able to capture the semantics of the gold answer. Moreover, lower Embedding Average and BLEU@2 scores suggest that it chose different vocabulary or used different word order. This is confirmed by lower ROUGE-L, which is based on longest common subsequence.

Finally, the retrieval (*ir*) model achieved the second-best results in terms of BLEU@2 and Embedding Average, but it was the worst according to the other three evaluation measures. This shows the superiority of the generative neural models over simple retrieval.

Table 5.5 shows some example responses generated by the three models. In the first example *(1)*, the IR model is off and retrieves an answer that addresses a different customer problem. The Seq2seq model is on the right track, because it asks the user about his device. The Transformer suggests a similar utterance, but it makes an assumption about the phone's operating system, which was not stated in the user's question. In the second example *(2)*, all models propose very different ways of action to the user, compared to the original answer, and they all seem plausible in this context; yet, the Transformer is a bit off. The next example *(3)* illustrates the ability of the three models to distinguish between different languages, and point the user in the right direction. The last example *(4)* is a typical example when neural models fail. The particular question–answer tuple is hard to answer as there are very few similar examples in the training data. Thus, what the neural models generate ends up being off-topic. In contrast, the retrieval approach was able to overcome this and to propose a very good answer.

5.4 Multi-Source Response Selection

The growing popularity of smart devices, personal assistants, and online customer support systems has driven the research community to develop various new methodologies for automatic question answering and chatbots. In the domain of conversational agents, two general types of systems have become dominant: (i) retrieval-based, and (ii) generative. While the former produce clear and

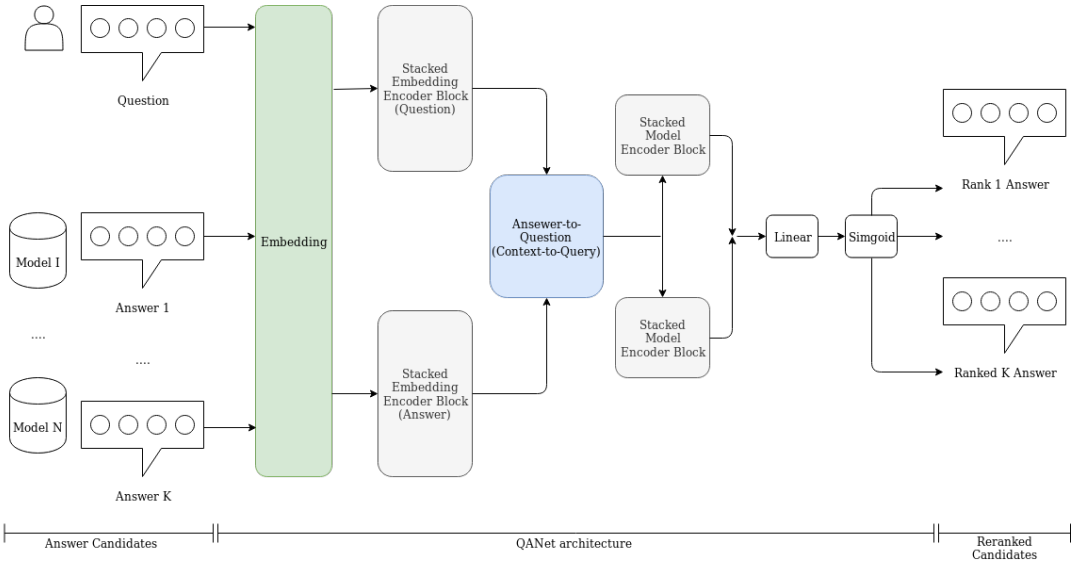


Figure 5.2: My answer re-ranking framework, based on the QANet architecture.

smooth output, the latter bring flexibility and the ability to generate new unseen answers.

In my thesis, I focus on finding the most suitable answer for a question, where each candidate can be produced by a different system, e.g., knowledge-based, rule-based, deep neural network, retrieval, etc. In particular, I propose a re-ranking framework based on machine reading comprehension (Seo et al., 2017; Chen et al., 2017; Yu et al., 2018) for question–answer pairs. Moreover, instead of selecting the top candidate from the re-ranker's output, I use probabilistic sampling that aims to diversify the agent's language and to up-vote popular answers from different input models. I train my model using negative sampling based on question–answer pairs from the Twitter Customer Support Dataset.

In my experimental setup, I adopt a real-world application scenario, where I train on historical logs for some period of time, and then I test on logs for subsequent days. I evaluate the model using both semantic similarity measures, as well as word-overlap ones such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), which come from machine translation and text summarization.

5.4.1 Re-Ranking Model

My re-ranking framework uses a classifier based on QANet (Yu et al., 2018), a state-of-the-art architecture for machine reading comprehension, to evaluate whether a given answer is a good fit for the target question. It then uses the posterior probabilities of the classifier to re-rank the candidate answers, as shown in Figure 5.2.

Negative Sampling

My goal is to distinguish "good" vs. "bad" answers, but the original dataset only contains valid, i.e., "good" question–answer pairs. Thus, I use *negative sampling* (Mikolov et al., 2013), where I replace the original answer to the target question with a random answer from the training dataset. I further compare the word-based cosine similarity between the original and the sampled answer, and, in some rare cases, I turn a "bad" answer into "good" one if it is too similar to the original "good" answer.

QANet Architecture

Machine reading comprehension aims to answer a question by looking to extract a string from a given text context. Here, I use that model to measure the appropriateness of a given question–answer pair.

The first layer of the network is a standard an embedding layer, which transforms words into low-dimensional dense vectors. Afterwards, a two-layer highway network (Srivastava et al., 2015) is added on top of the embedding representations. This allows the network to regulate the information flow using a gated mechanism. The output of this layer is of dimensionality $\#words \times d$, where $\#words$ is the number of words in the encoded sentence (Note that it differs for the question vs. the answer. See Section 5.4.2 for more detail.) and d is the input/output dimensionality of the model for all Transformer layers, which is required by the architecture.

I experiment with two types of input embeddings. First, I use 200-dimensional GloVe (Pennington et al., 2014) vectors trained on 27 billion Twitter posts. I compare their performance to ELMo (Peters et al., 2018), a recently proposed way to train contextualized word representations. In ELMo, these word vectors are learned activation functions of the internal states of a deep bi-directional language model. The latter is built upon a single (embedding) layer, followed by two LSTM (Hochreiter and Schmidhuber, 1997) layers, which are fed the words from a target sentence in a forward and a backward direction, accordingly. I obtain the final embedding by taking a weighted average over all three layers as suggested in (Peters et al., 2018).

The embedding encoder layer is based on a convolution, followed by self-attention (Vaswani et al., 2017) and a feed-forward network. I use a kernel size of seven, d filters, and four convolutional layers within a block. The output of the layer is $f(layernorm(x)) + x$, where $layernorm$ is the layer normalization operation (Ba et al., 2016). The output again is mapped to $\#words \times d$ by a 1D convolution. The input and the embedding layers are learned separately for the question and the answer.

The attention layer is a standard module for machine reading comprehension models. I call it *answer-to-question* ($A2Q$) and *question-to-answer* ($Q2A$) attention,

which are also known as *context-query* and *query-context*, respectively. Let us denote the output of the encoder for the question as Q and for the answer as A. In order to obtain the attention, the model first computes a matrix S with similarities between each two words for the question and the answer, then the values are normalized using softmax. The similarity function is defined as follows: $f(a,q) = W_0[a;q;a \odot q]$.

I adopt the notation $\overline{S} = softmax(S)$, which is a softmax normalization over the rows of S, and $\overline{\overline{S}} = softmax(S^\intercal)$ is a normalization over the columns. Then, the two attention matrices are computed as $A2Q = \overline{S} \cdot Q^\intercal$, and $Q2A = \overline{S} \cdot \overline{\overline{S}}^\intercal \cdot C^\intercal$.

The attention layer is followed by a model layer, which takes as input the concatenation of $[a; a2q; a \odot a2q; a \odot q2a]$, where I use small letters to denote rows from the original matrices. For the output layer, I learn two different representations by passing the output of the model layer to two residual blocks, applying dropout (Srivastava et al., 2014) only to the inputs of the first one. I predict the output as $P(a|q) = \sigma(W_o[M_0; M_1])$. The weights are learned by minimizing a binary cross-entropy loss.

Answer Selection

I experimented with two answer selection strategies: (i) max, and (ii) proportional sampling after softmax normalization. The former strategy is standard and it selects the answer with the highest score, while the latter one returns a random answer with probability proportional to the score returned by the softmax, aiming at increasing the variability of the answers.

For both strategies, I use a linear projection applied on the output of the last residual model block, which is shows as "linear block" in Figure 5.2. I can generalize the latter as follows: $o(q, a_k) = W_o[M]$, where M is the concatenation of the outputs of one or more residual model blocks.

I present the formulation of the two strategies, as I introduce the following notation: Ans is the selected utterance by the agent; $o(q, a_k)$ is the output of the model before applying the sigmoid function; q is the original question by the user; A is the set of possible answers that I want to re-rank. Equation (5.6) shows the selection process in the max case.

$$Ans = \arg\max_{a \in A}(o(q,a)) \tag{5.6}$$

I empirically found that the answer selection based on the *max* strategy does not always perform well. As my experimental results in Tables 5.7 and 5.8 show, I can gain notable improvement by using proportional sampling after softmax normalization, instead of always selecting the answer with the highest probability. In

my experiments, I model *Ans* as a random variable that follows a categorical distribution over $K = |A|$ events (candidate answers). For each of the question–answer pairs (q, a), I define the probability p that a is a good answer to q using softmax as shown in Eq. 5.7 and 5.8. Finally, I draw a random sample from Eq. 5.8 to obtain the best matching answer.

$$p|q, A \sim \text{softmax}(o(q, a_1), \cdots, o(q, a_K)) \tag{5.7}$$

$$Ans|p \sim \text{Cat}(K, p) \tag{5.8}$$

5.4.2 Experiments

Preprocessing

Since Twitter has its own specifics of writing in terms of both length (by design, tweets have been strictly limited to 140 characters; this constraint has been relaxed to 280 characters in 2017) and style, standard text tokenization is generally not suitable for tweets. Therefore, I used a specialized Twitter tokenizer (Manning et al., 2014) to preprocess the data. Then, I further replaced shorthand entries such as *'ll*, *'d*, *'re*, *'ve*, with the most corresponding literary form, e.g., *will*, *would*, *are*, *have*. I also replaced shortened slang words, e.g., *'bout* and *'til*, with the standard words, e.g., *about* and *until*. Similarly, I replaced URLs with the special word <*url*>, all user mentions with <*user*>, and all hashtags with <*hashtag*>.

Due to the nature of writing in Twitter and the free form of the conversation, some of the utterances contain emoticons and emojis. They are handled automatically by the Twitter tokenizer and treated as a single token. I keep them in their original form, as they can be very useful for detecting emotions and sarcasm, which pose serious challenges for natural language understanding.

Based on the statistics presented in Section 4.2.2, I chose to trim the length of the questions and of the answers to 60 and 70 words, respectively.

Training Setup

For training, I use the Adam (Kingma and Ba, 2015) optimizer with decaying learning rate, as implemented in TensorFlow (Abadi et al., 2016). I start with the following values: learning rate η = 5e-4, exponential decay rate for the 1st and the 2nd momentum β_1 = 0.9 and β_2 = 0.999, and constant for prevention of division by zero ϵ = 1e-7. Then, I decay the learning after each epoch by a factor of 0.99. I also apply dropout with a probability of 0.1, and L2 weight decay on all trainable variables with λ =3e-7. I train each model for 42K steps with a batch size of 64. I found these values by running a grid search on a dev set (extracted as a fraction of the training data) and using the values suggested in (Yu et al., 2018), where applicable.

Individual Models

Following Section 5.3, here I experiment with three individual models: (i) information retrieval-based, (ii) sequence-to-sequence and (iii) the Transformer.

For IR, I use ElasticSearch with English analyzer enabled, whitespace- and punctuation-based tokenization, and word 3-grams. I further use the default BM25 algorithm (Robertson and Zaragoza, 2009), which is an improved version of TF.IDF. For all training questions and for all testing queries, I append the previous turns in the dialog as context.

For Seq2seq, I use a bi-directional LSTM network with 512 hidden units per direction. The decoder has two uni-directional layers connected directly to the bi-directional layer in the encoder. The network takes as input words encoded as 200-dimensional embeddings. It is a combination of pre-trained GloVe (Pennington et al., 2014) vectors for the known words, and a positional embedding layer, learned as model parameters, for the unknown words. The embedding layers for the encoder and for the decoder are not shared, and are learned separately. This separation is due to the words used in utterances by the customers being very different from the posts by the customer support.

For the Transformer, I use two identical layers for the encoder and for the decoder, with four heads for the self-attention. The dimensionality of the input and of the output is $d_{model} = 256$, and the inner dimensionality is $d_{inner} = 512$. The input consists of queries with keys of dimensionality $d_k = 64$ and values of the same dimensionality $d_v = 64$. The input and the output embedding are learned separately with sinusoidal positional encoding.

Evaluation Measures

In order to allow for fair comparison with single models (see Section 5.3), I use the same evaluation measures described in details in Section 5.3.2.

5.4.3 Evaluation Results

Below, I first discuss my auxiliary classification task, where the objective is to predict which question–answer pair is "good", and then I move to the main task of answer re-ranking.

Auxiliary Task: Question–Answer Appropriateness Classification

Table 5.6 shows the results for the auxiliary task of question–answer appropriateness classification. The first column is the name of the model. It is followed by three columns showing the type of embedding used, the size of the hidden layer, and the number of heads (see Section 5.4.1). The last column reports the accuracy.

Model	Embedding Type	d_model	Heads	Accuracy
Majority class	–	–	–	50.52
QANet	GloVe	64	4	80.58
		64	8	82.83
		128	8	83.42
QANet	ELMo (token level)	64	4	82.92
		64	8	83.88
		128	8	83.48
QANet	ELMo (sentence level)	64	8	84.09
		128	8	**85.45**

Table 5.6: Auxiliary task: question–answer appropriateness classification results.

Since the dataset is balanced (I generate about 50% positive, and about 50% negative examples), accuracy is a suitable evaluation measure for this task. The top row of the table shows the performance for a majority class baseline. The following lines show the results for my full QANet-based model when using different kinds of embeddings. We can see that contextualized sentence-level embeddings are preferable to using simple word embeddings as in GloVe or token-level ELMo embeddings. Moreover, while token-level ELMo outperforms GloVe when the size of the network is small, there is no much difference when the number of parameters grows ($d_{model} = 128$, $\#Heads = 8$).

Answer Selection/Generation: Individual Models

Table 5.7 reports the performance of the individual models: information retrieval (IR), sequence-to-sequence (Seq2seq), and the Transformer (see Section 5.4.2 for more details about these models). The same experimental setup is used for the experiments described in Section 5.3. The table is organized as follows: The first column contains the name of the model used to obtain the best answer. The second and the third columns report the word overlap measures: (i) BLEU@2, which uses uni-gram and bi-gram matches between the hypothesis and the reference sentence, and (ii) ROUGE-L (Lin and Och, 2004), which uses LCS. The last three columns are for the semantic similarity measures: (i) Embedding Average (Emb Avg) with cosine similarity, (ii) Greedy Matching (Greedy Match), and (iii) Vector Extrema (Vec Extr) with cosine similarity. In the three latter measures, I used the standard pre-trained word2vec embeddings because they are not learned during training, which helps avoid bias, as has been suggested in (Liu et al., 2016; Lowe et al., 2017).

We can see in Table 5.7 that the Seq2seq model outperforms IR by a margin on all five evaluation measures, which is consistent with previous results in the literature. What is surprising, however, is the relatively poor performance for the Transformer, which trails behind the Seq2seq model on all evaluation measures. I hypothesize that this is due to the Transformer having to learn more parameters as

	Word Overlap		Semantic Similarity		
Model	**BLEU@2**	**ROUGE_L**	**Emb Avg**	**Greedy Match**	**Vec Extr**
Transformer	12.43	25.33	75.35	30.08	39.40
IR-BM25	13.73	22.35	76.53	29.72	37.99
Seq2seq	15.10	26.60	77.11	30.81	40.23
QANet on IR *(Individual)*	14.92 ± 0.13	23.30 ± 0.12	77.47 ± 0.06	30.40 ± 0.06	39.63 ± 0.06

Table 5.7: Main task: performance of the individual models. Single model results results are reported in Section 5.3.2, Tables 5.3 and 5.4

it operates with higher-dimensional word embeddings. Overall, the Transformer is arguably slightly better than the IR model, outperforming it on three of the five evaluation measures.

The last row of Table 5.7 is not an individual model; it is my re-ranker applied to the top answers returned by the IR model. In particular, I use *QANet with Sentence level ELMo* ($d_{model} = 128$, $\#Heads = 8$). I took the top-5 answer candidates (the value of 5 was found using cross-validation on the training dataset) from the IR model, and I selected the best answer based on the re-ranker's scores. I can see that re-ranking yields improvements for all evaluation measures: +1.18 on BLEU@2, +0.93 on ROUGE_L, +1.12 on Embedding Average, +0.67 on Greedy Matching, and +1.64 in Vector Extrema. These results show that I can get sizable performance gains when re-ranking the top-K predictions of a single model; below I will combine multiple models.

Main Task: Multi-Source Answer Re-Ranking

Next, I combine the top-K answers from different models: IR and Seq2seq. I did not include the Transformer in the mix as its output is generative and similar to that of the Seq2seq model; moreover, as we have seen in Table 5.7 above, it performs worse than Seq2seq on the dataset. I set $K = 2$ for the baseline, *Random Top Answer*, which selects a random answer from the union of the top K answers by the models involved in the re-ranking. For the remaining re-ranking experiments, I use $K = 5$. I found these values using cross-validation on the training dataset, trying 1–5.

The results are shown in Table 5.8, where different representations are separated by a horizontal line. The first row of each group contains the name of the model. Then, on the even rows (second, forth, etc.), I show the results from a greedy answer selection strategy, while on the odd rows are the results from an exploration strategy (softmax sampling). Since softmax sampling and random selection are stochastic in nature, I include a 95% confidence interval for them.

We can see in Table 5.8 that *QANet with sentence-level ELMo* ($d_{model} = 128$, $\#Heads = 8$) performs best in terms of BLEU@2, ROUGE_L, and Greedy Matching.

Model	Word Overlap		Semantic Similarity		
	BLEU@2	**ROUGE_L**	**Emb Avg**	**Greedy Match**	**Vec Extr**
Random Top Answer	14.52 ± 0.12	23.41 ± 0.12	77.21 ± 0.06	30.24 ± 0.07	38.25 ± 0.20
QANet+GloVe					
d=64, h=4	15.18	24.13	78.38	31.14	**40.85**
Softmax	15.81 ± 0.09	24.53 ± 0.05	78.32 ± 0.08	31.10 ± 0.03	40.51 ± 0.12
d=64, h=8	15.41	23.62	78.48	30.97	40.81
Softmax	15.90 ± 0.06	24.39 ± 0.03	78.38 ± 0.04	31.11 ± 0.02	40.66 ± 0.06
d = 128, h = 8	15.94	24.59	78.29	31.19	40.63
Softmax	16.04 ± 0.08	24.71 ± 0.06	78.36 ± 0.07	31.20 ± 0.07	40.70 ± 0.05
QANet+ELMo (Token)					
d = 64, h = 4	15.23	23.48	78.25	30.77	40.22
Softmax	15.77 ± 0.15	24.44 ± 0.09	78.27 ± 0.03	31.06 ± 0.05	40.46 ± 0.11
d = 64, h = 8	15.30	23.41	**78.54**	30.97	40.19
Softmax	15.86 ± 0.07	24.40 ± 0.06	78.36 ± 0.08	31.11 ± 0.04	40.49 ± 0.05
d = 128, h = 8	15.24	23.59	78.34	30.90	40.19
Softmax	15.89 ± 0.08	24.55 ± 0.10	78.33 ± 0.06	31.11 ± 0.05	40.40 ± 0.05
QANet+ELMo (Sentence)					
d = 64, h = 8	15.48	23.88	78.44	30.96	40.33
Softmax	16.00 ± 0.14	24.50 ± 0.33	78.34 ± 0.10	31.13 ± 0.08	40.56 ± 0.09
d = 128, h = 8	15.64	24.13	78.52	31.14	40.63
Softmax	$\mathbf{16.05 \pm 0.06}$	$\mathbf{24.81 \pm 0.08}$	78.40 ± 0.07	$\mathbf{31.20 \pm 0.06}$	40.58 ± 0.03

Table 5.8: Main task: re-ranking the top $K = 5$ answers returned by the IR and the Seq2seq models.

Note also the correlation between higher results on the auxiliary task (see Table 5.6) and improvement in terms of word-overlap measures, where I find the largest difference between individual and re-ranked models (+1.5 points absolute over the baseline, and +0.95 over Seq2seq in terms of BLEU@2). In terms of semantic similarity, I note the highest increase for Embedding Average (+1.3 over the baseline, and +1.4 over Seq2seq), and a smaller one for Greedy Matching (+1.0 over the baseline, and +0.4 over Seq2seq), and Vector Extrema (+2.6 over the baseline, and +0.6 over Seq2seq).

Overall, the re-ranked models are superior as evaluated on word-matching measures, which is supported by the improvement of BLEU@2 and Embedding Average. The smaller improvement for Greedy Matching and Vector Extrema can be explained by the training procedure for the re-ranking model, which is based on word comparison. However, these two measures focus on keyword similarity between the target and the proposed answers, and generative models are better at this. This is supported by comparing the combined model to IR-BM25, where I see sizable improvements of +1.5 and +2.0 in terms of Greedy Matching and Vector Extrema, respectively.

We can further see in Table 5.8 that using a stochastic approach to select the best answer yields additional improvements. This strategy accounts for the predicted appropriateness score for each candidate, thus, enriching the model in two

ways. First, implicit voting is used, as duplicate answer candidates are not removed, resulting in higher selection probability of popular answers from different input modules. Second, albeit two answers may have a very different structure, they still can be similar in meaning, leading to very similar scores and promoting only the first one. This behavior can be mitigated by choosing the winner proportionally to its ranking, thus, also introducing diversity in the chatbot's language. This hypothesis is supported by the results in Table 5.8: compare each model to the corresponding one with *softmax* selection.

5.5 Multi- and Cross-Linguality

Research on science question answering has attracted a lot of attention in recent years (Clark, 2015; Schoenick et al., 2017; Clark et al., 2019). Such questions are challenging as they require domain and common sense knowledge (Clark et al., 2018), as well as complex reasoning and different forms of inference over a variety of knowledge sources (Khashabi et al., 2016, 2018). Indeed, a combination of these was required to achieve noticeable performance gains (Clark et al., 2016). This inevitably made research in school-level science question answering (QA) hard for languages other than English due to the scarceness of resources (Clark et al., 2014; Khot et al., 2017, 2018; Bhakthavatsalam et al., 2020).

There has been a recent mini-revolution in QA, as well as in the field of natural language processing (NLP) in general, due to the invention of the Transformer (Vaswani et al., 2017), and the subsequent rise of large-scale pre-trained models (Peters et al., 2018; Radford et al., 2018, 2019; Devlin et al., 2019; Lan et al., 2020; Yang et al., 2019; Liu et al., 2019; Raffel et al., 2020). Nowadays, fine-tuning

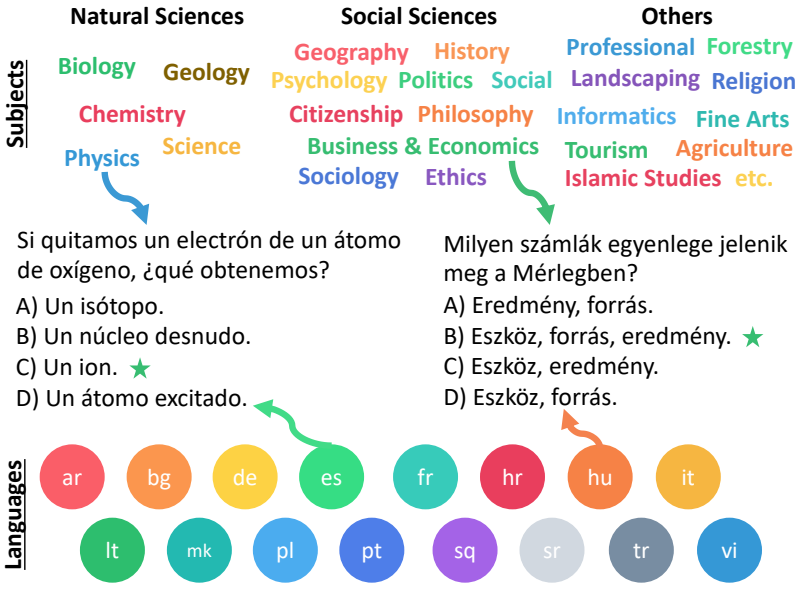


Figure 5.3: Properties and examples from *Εχαμς*.

such models on task-specific data has become an essential element of any top-scoring QA system. Yet, for science QA, training on datasets from a different domain (Sun et al., 2019; Khashabi et al., 2020) and carefully selected background knowledge (Banerjee et al., 2019; Ni et al., 2019) could improve such models further.

The success of large-scale pre-trained models and the development of their multilingual versions (Devlin et al., 2019; Conneau et al., 2020) gives hopes for supposedly better performance in multilingual question answering. Therefore, several new datasets have been released for multilingual reading comprehension and open-domain question answering in the Wikipedia domain (Liu et al., 2019; Lewis et al., 2020; Artetxe et al., 2020; Clark et al., 2020).

Here, I present *Eχαμs*, a new dataset and benchmark for multilingual and cross-lingual evaluation of models and methods for answering diverse school science questions (see Figure 5.3).

I release the code, pre-trained models and data for research purposes.[7]

5.5.1 *Eχαμs* Dataset

I introduce *Eχαμs*, a new benchmark dataset for multilingual and cross-lingual question answering from high school examinations. In this section, I present the properties of the dataset, and I give details about the process of data collection, preparation and normalization, as well as information about the data splits, and the parallel questions.

Lang	Family	#Subjects	Question Len	Choice Len	#Choices	#Questions	Vocab
Albanian	Albanian	8	15.0	5.0	4.0	1,505	11,572
Arabic	Semitic	5	10.3	3.4	4.0	562	5,189
Bulgarian	Balto-Slavic	6	13.0	3.3	4.0	2,937	15,127
Croatian	Balto-Slavic	14	14.7	4.1	3.9	2,879	20,689
French	Romance	3	18.4	10.5	3.5	318	2,576
German	Germanic	5	18.3	9.1	3.5	577	4,664
Hungarian	Finno-Ugric	10	11.6	5.9	3.9	2,267	15,045
Italian	Romance	12	20.0	5.6	3.9	1,256	9,050
Lithuanian	Balto-Slavic	2	9.7	4.7	4.0	593	5,394
Macedonian	Balto-Slavic	8	13.4	4.5	4.0	2,075	13,114
Polish	Balto-Slavic	1	13.7	4.3	4.0	1,971	18,990
Portuguese	Romance	4	19.9	8.6	4.0	924	6,811
Serbian	Balto-Slavic	14	15.4	4.3	3.9	1,637	15,509
Spanish	Romance	2	23.0	10.2	3.2	235	2,130
Turkish	Turkic	8	19.5	4.6	4.4	1,964	22,069
Vietnamese	Austroasian	6	37.0	6.4	4.0	2,443	6,076
#Langs 16	#Families 8	24	17.19	5.08	3.96	24,143	158,942

Table 5.9: Statistics about *Eχαμs*. The average length of the question (*Question Len*) and the choices (*Choice Len*) are measured in number of tokens, and the vocabulary size (*Vocab*) is measured in number of words.

[7]The *Eχαμs* dataset and code are publicly available at http://github.com/mhardalov/exams-qa

Dataset Statistics

I collected *Eχαμs* from official state exams prepared by the ministries of education of various countries. These exams are taken by students graduating from high school, and often require knowledge learned through the entire course. The questions cover a large variety of subjects and material based on the country's education system. Moreover, I do not focus only on major school subjects such as Biology, Chemistry, Geography, History, and Physics, but I also cover highly-specialized ones such as Agriculture, Geology, Informatics, as well as some applied and profiled studies. These characteristics make the questions in the dataset of very high variety, and not easily solvable, due to the need for highly specialized knowledge. Next, I discuss the cross-lingual and the multilingual properties of my dataset.

Parallel Questions Some countries allow students to take official examinations in several languages. Such parallel examinations also exist in my dataset. In particular, there are 9,857 parallel question pairs spread across seven languages as shown in Table 5.10. The parallel pairs are coming from Croatia (Croatian, Serbian, Italian, Hungarian), Hungary (Hungarian, German, French, Spanish, Croatian, Serbian, Italian), and North Macedonia (Macedonian, Albanian, Turkish).

	de	**es**	**fr**	**hr**	**hu**	**it**	**mk**	**sq**	**sr**
de	-								
es	199	-							
fr	253	120	-						
hr	189	134	109	-					
hu	456	159	274	236	-				
it	30	9	15	1,214	99	-			
mk	0	0	0	0	0	0	-		
sq	0	0	0	0	0	0	1,403	-	
sr	40	25	20	1,564	104	1,002	0	0	-
tr	0	0	0	0	0	0	1,222	981	0

Table 5.10: Parallel questions for different language pairs.

Multilinguality The dataset includes a total of 24,143 questions in 16 languages from eight language families. Each question is a 3-way to 5-way (3.96 on average) multiple-choice question with a single correct answer. Table 5.9 shows a breakdown for each language, where the number of subjects, questions, and the vocabulary size are shown as absolute numbers, while the question length, the choice length, and the number of choices are averaged. All statistics about the questions and the answer options are measured in terms of words. We see that I have a rich vocabulary with almost 160,000 unique words. Interestingly, there are ∼9,500 shared words between at least one pair of languages in the dataset, excluding numbers and punctuation. As expected, the overlapping words are mostly between closely related languages (bg-mk, bg-sr, es-it, es-pt, hr-sr, mk-sr). Other common shared words are subject-specific words such as person names (e.g., *Abraham*, *Karl*, *Ivan*),

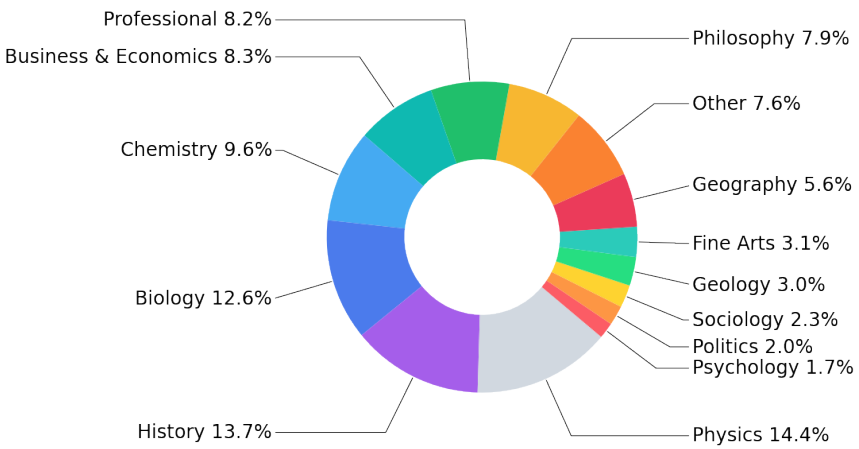


Figure 5.4: Relative sizes of the subjects. Those that cover less than 1.5% of the examples are in *Other*.

chemical compounds (e.g., *NaOH*, *HCl*), units (e.g., *m/s*, *g/mol*), etc. Then, there are cognates with the exact same spelling (homographs) even between unrelated languages, mostly words of Latin or Greek origin, e.g., *temperatura* (temperature) and *forma* (form). Finally, there are also *false friends*, whose meaning differs across languages, e.g., *para* can mean *for* (es/pt) vs. *money* (mk/tr/sq) vs. *couple* (pl); similarly, *ser* can mean *be* (es/pt) vs. *cheese* (pl) vs. *after* (vi).

Subjects and Categories

Each education system has its own specifics, resulting in some differences in curricula, topics, and even naming of the subjects. That being said, the original, non-normalized categories in the dataset are more than 40 for exams from just a few countries. Given the sparse nature of the subjects, I use a two-level taxonomy in order to categorize them into logically connected groups. The lower-level is a subject, and the higher level is a major group. I normalized the subject using a two-step algorithm: first, I put each subject (with its original naming) in a separate category, then, if the subject was general enough, e.g., Biology, History, etc., or there were no similar ones, I retained the category; otherwise, I merged all similar subjects together in a unifying category, e.g., Economics Basics, and Economics & Marketing. I repeated the aforementioned steps until there were no suitable merge candidates. As a result, I ended up with a total of 24 subjects (see Section 5.5.1 for more details), which I further grouped into three major categories, based on the main branches of science: **Natural Science** – *"the study of natural phenomena"*, **Social Sciences** – *"the study of human behavior and societies"*, **Other** – *Applied Studies, Arts, Religion, etc.* (see Figure 5.3).[8]

The distribution of the major categories is *Natural Sciences* (40.0%) and *Social Sciences* (44.0%) and 16.0% for *Others* (these are the actual numbers, not approximate). The remaining questions are labeled as *Other* as they are not suitable for

[8] https://en.wikipedia.org/wiki/Branches_of_science

the two main categories. Figure 5.4 presents the relative sizes of the subjects in the dataset.

Collection and Preparation

Here, I describe the process of collecting and preparing the data, as it is not trivial and it could be applied to other languages and examinations. First, I identified potential online sources of publicly available school exams starting from the *Matriculation Examination* page in Wikipedia.[9]

For all languages in the dataset, the first step in the process of data collection was to download the PDF files per year, per subject, and per language (when parallel languages were available in the same source). I converted the PDF files to text and I used only those that were well-formatted and followed the document structure.

Then, I used Regular Expressions (RegEx) to parse the questions, their corresponding choices and the correct answer choice. In order to ensure that all the questions are answerable using textual input only, I removed questions that contained visual information. I did that using a manually curated list of words such as *map*, *table*, *picture*, *graph*, etc., in the corresponding language. Next, I performed data cleaning to ensure the quality of the generated dataset, by manually reviewing each question and its choices and ensuring that all options, text, and symbols (e.g., μ, $\rightarrow$, α, $\leftarrow$) were displayed correctly. As a result, I filtered out about 17% of the questions (the percentage varies based on the source, the language, and the subject). Finally, in order to remove frequency bias such as "most answers are B)", I shuffled each question's choices.

Data Splits

In my experiments, I aim at evaluating the multilingual and the cross-lingual question answering capabilities of different models. Therefore, I split the data in order to support both evaluation strategies: *Multilingual* and *Cross-lingual*.

Multilingual In this setup, I want to train and to evaluate a given model with multiple languages, and thus I need multilingual *training*, *validation* and *test* sets. In order to ensure that I include as many of the languages as possible, I first split the questions independently for each language L into Train_L, Dev_L, Test_L with 37.5%, 12.5%, 50% of the examples, respectively.[10] I then unite all language-specific subsets into the multilingual sets Train_{Mul}, Dev_{Mul}, Test_{Mul}, and I used them for training, development, and testing.

[9] https://en.wikipedia.org/wiki/Matriculation_examination

[10] For languages with fewer than 900 examples, I only have Test_L.

	Multilingual			Cross-lingual	
Language	**Train**	**Dev**	**Test**	**Train**	**Dev**
Albanian	565	185	755	1,194	311
Arabic	-	-	562	-	-
Bulgarian	1,100	365	1,472	2,344	593
Croatian	1,003	335	1,541	2,341	538
French	-	-	318	-	-
German	-	-	577	-	-
Hungarian	707	263	1,297	1,731	536
Italian	464	156	636	1,010	246
Lithuanian	-	-	593	-	-
Macedonian	778	265	1,032	1,665	410
Polish	739	246	986	1,577	394
Portuguese	346	115	463	740	184
Serbian	596	197	844	1,323	314
Spanish	-	-	235	-	-
Turkish	747	240	977	1,571	393
Vietnamese	916	305	1,222	1,955	488
Combined	7,961	2,672	13,510	-	-

Table 5.11: Number of examples in the data splits based on the experimental setup.

Since I have parallel data for several languages (discussed in Section 5.5.1), in this setup, I ensure that the same parallel questions are only found in either training, development or testing, so that I do not leak the answer from training via some other language. In order to do that, I sample the questions with the assumptions and the ratios mentioned above, stratified per subject in the given language. The number of examples per language and the total number of multilingual sets are shown in the first three columns of Table 5.11.[11]

Cross-Lingual In this setting, I want to explore the capability of a model to transfer its knowledge from a single source language L_{src} to a new unseen target language L_{tgt}. In order to ensure that I have a larger training set, I train the model on 80% of L_{src}, I validate on 20% of the same language, and I test on a subset of L_{tgt}.[12] The last three columns of Table 5.11 show the number of examples used for training and validation with the corresponding language.

Reasoning and Knowledge Types

In order to give a better understanding of the reasoning, and the knowledge types in $E\chi\alpha\mu s$, I sampled and annotated 250 questions, all of which are from the multilingual Dev. For each question, I provided English translations as not all annotators

[11]Sometimes, grouping parallel questions in the same split slightly violates the splitting ratios.

[12]To ensure that the cross-lingual evaluation is comparable to the multilingual one, I use the same subset of questions from language L_{tgt} that are used in Test_{Mul}

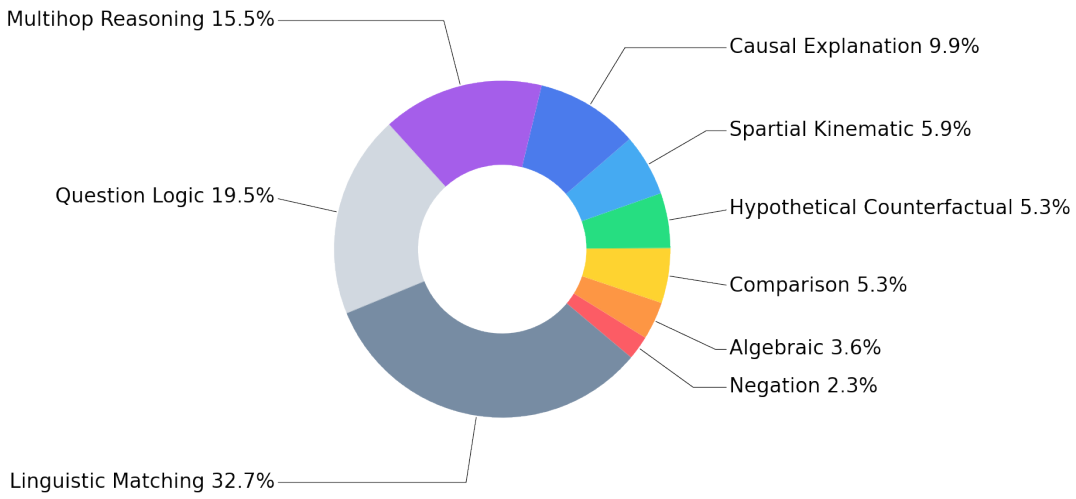


Figure 5.5: Relative sizes of reasoning types in *Εχαμς*.

were native speakers of the questions' language. I followed the procedure and re-used the annotation types presented in earlier work (Clark et al., 2018; Boratko et al., 2018). However, as they were designed mainly for Nature Science questions, I extended them with two new annotation types: *"Domain Facts and Knowledge"* and *"Negation"*. I define these types as:

Domain Facts and Knowledge (Knowledge) This skill requires specific expertise in properties and facts in a given domain, e.g., physical properties, characteristics of a chemical element.

Example from Philosophy (*Portugal*):

> *Which of the following is an example of a priori knowledge?*
> A) I know my name.
> B) I know how old I am.
> C) *I know that no brother is an only child.* ✓
> D) I know some parents are not married.

Negation (Reasoning) is a direct statement of negation, and it is often combined with other reasoning types such as linguistic matching.

Example from Fine Arts (*North Macedonia*):

> *Which of the following works of art does **not** belong to the fine arts?*
> A) Graphics.
> B) *Poem.* ✓
> C) Design.
> D) Sculpture.

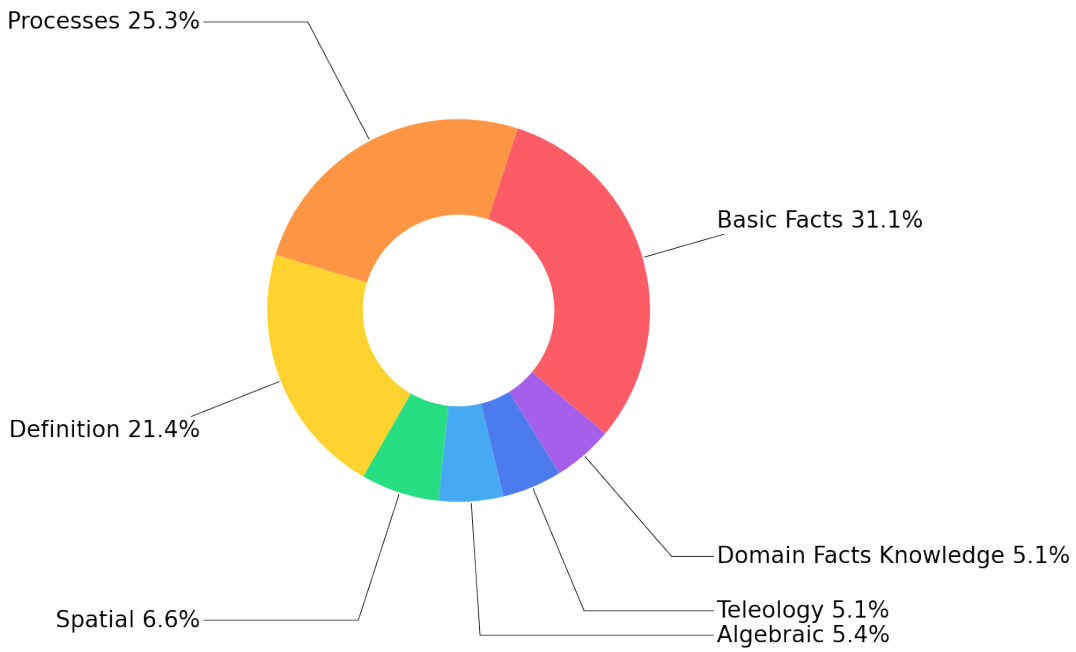


Figure 5.6: Relative size of the *Εχαμς* knowledge types.

The relative sizes of the knowledge and the reasoning types are shown in Figures 5.5 and 5.6. Here, I must note that the sizes are approximate rather than exact, since the annotations are subjective and the distribution may vary.

Subject Analysis

The *Natural Science* group contains five subjects. The corresponding question length is 16.4 characters and 3.9 answers on average. Some of the subjects are well-known and widely studied, such as *Physics*, *Biology* and *Chemistry*. They appear in at least 10 out of the 16 languages, covering 7 out of 8 language families. However, *Geology* is less common and is present for only 4 languages. Finally, *Science* is an isolated subject for Arabic. This group contributes a total of 9,962 questions in the entire dataset, as shown in Table 5.12. The major groups in the table are divided with a horizontal line for convenience.

The second subject group covers *Social Sciences*. *Geography*, *History*, *Philosophy*, *Psychology* and *Ethics* are more common, and thus are included in seven languages on average (see Table 5.12). The subject group's average question length is 18.5 characters. The only sizable deviation being for Citizenship, as most of the questions in this subject explain some social situation in detail.

The last and smallest of the three subject groups is *Others*. It combines subjects that cannot be categorized as exactly science-related (either social or natural). Those subjects are often specific for a particular country or culture and are fairly diverse. As expected, they are present for less languages (just two).

Group	Subject	Language	Grade	Q Len	Ch Len	#Ch	#Q	Vocab
Natural Science	Biology	ar, bg, hr, hu, it, sr, sq, mk, tr, pt, vi	H	18.2	4.6	4.0	3,042	24,603
Natural Science	Chemistry	bg, hr, it, sr, de, hu, sq, mk, tr, vi	H	17.3	4.6	4.2	2,315	14,420
Natural Science	Geology	hr, it, sr, lt, pt	H	12.9	5.6	4.0	720	7,251
Natural Science	Physics	ar, bg, hr, it, sr, fr, de, hu, es, sq, mk, tr, vi	H	24.9	7.0	3.6	3,465	26,103
Natural Science	Science	ar	M, H	9.1	3.0	4.0	120	1,239
Social Science	Busin. & Econ.	fr, de, hu, sq, mk, tr, pt	H	5.7	6.5	3.9	2,012	16,875
Social Science	Citizenship	vi	H	45.1	6.3	4.0	119	980
Social Science	Ethics	hr, it, sr	H	15.5	2.6	4.0	194	1,859
Social Science	Geography	bg, hr, fr, de, hu, it, sr, es, tr, vi	H	15.2	5.0	4.2	1,349	11,207
Social Science	History	bg, hr, it, sr, lt, sq, mk, tr, vi	H	16.6	5.9	4.1	3,300	32,709
Social Science	Philosophy	bg, hr, it, sr, sq, mk, tr, pt	H	16.5	3.9	4.1	1,903	19,373
Social Science	Politics	hr, hu, it, sr	H	18.2	2.8	3.0	493	5,068
Social Science	Psychology	hr, it, sr	H	16.5	3.9	4.1	1,903	19,373
Social Science	Social	ar	M, H	10.8	3.4	4.0	277	2,828
Social Science	Sociology	hr, it, sr, sq, mk, tr	H	15.2	3.4	4.0	566	6,374
Other	Agriculture	hu	H	7.9	3.6	4.3	215	1,918
Other	Fine Arts	sq, mk	H	12.1	3.8	4.0	757	5,691
Other	Forestry	hu	H	7.8	2.9	3.7	241	1,957
Other	Informatics	hr, it, sr	H	18.7	6.2	4.0	311	2,695
Other	Islamic Studies	ar	M, H	9.4	3.0	4.0	78	925
Other	Landscaping	hu	H	7.4	3.8	4.9	49	596
Other	Professional	pl	H	13.7	4.3	4.0	1,971	18,990
Other	Religion	hr, sr	H	10.3	3.6	4.0	222	2,159
Other	Tourism	de, hu	H	8.8	5.2	4.0	20	359

Table 5.12: Per-subject statistics. The grade is High (H), and Middle (M). The average length of the question (*Q Len*) and the choices (*Ch Len*) are measured in number of tokens, and the vocabulary size (*Vocab*) is shown in number of words.

5.5.2 Background Knowledge Corpus

Students need good textbooks to study before they can pass an exam, and the same holds for a good machine reading model. However, finding the information needed to answer a question, especially for questions in such a narrow domain as the subjects studied in high schools, usually requires a collection of specialized texts. The ARC Corpus (Clark et al., 2018) is an example of such a collection. It is built by querying a major search engine, and around 100 hand-written templates for 80 science topics covered by US elementary and middle schools. Albeit effective, this strategy relies on crafting templates for all language–subject pairs, making the task time-consuming if applied to multiple languages and subjects.

In my work, I used articles from Wikipedia to build a background knowledge corpus for each language. In particular, I parsed the text from the entire Wikipage, removing non-textual content, e.g., HTML tags, tables, etc. Following the common strategy used to solve similar tasks in English (Clark et al., 2018; Mihaylov et al., 2018), I split each document into sentences and I indexed them using an inverted index. In order to reduce the search space, and to mitigate the effect of known

Language	Wiki code	#Sentences (millions)	#Articles (millions)	Stop word removal	Stemming	Keyword extraction	Language specific
ARC Corpus	-	14.6	-	✓	✓	✓	✓
German	de	50.0	2.43	✓	✓	✓	✓
French	fr	30.0	2.22	✓	✓	✓	✓
Italian	it	17.5	1.61	✓	✓	✓	✓
Spanish	es	22.7	1.60	✓	✓	✓	
Polish	pl	15.6	1.41		✓		
Vietnamese	vi	6.4	1.25	✓	✓		✓
Portuguese	pt	11.6	1.03	✓	✓	✓	
Arabic	ar	6.0	1.04	✓	✓	✓	✓
Serbian	sr	4.6	0.63				
Hungarian	hu	7.1	0.47	✓	✓	✓	
Turkish	tr	4.0	0.35	✓	✓	✓	✓
Bulgarian	bg	3.0	0.26	✓	✓	✓	
Croatian	hr	2.7	0.22				
Lithuanian	lt	2.0	0.20	✓	✓	✓	
Macedonian	mk	1.6	0.11				
Albanian	sq	0.8	0.08				

Table 5.13: Description of the per-language indices used as a source of background knowledge in my experiments.

linguistic phenomena within the same language family, e.g., homonyms, partially shared alphabet, etc., I created a separate index for each language.

Table 5.13 describes the main characteristics of the indices created for each language from its Wikipedia dump.[13] I compared the size of the index to the one from ARC (Clark et al., 2018). The number of articles for each language is taken from Wikipedia's official statistics [14]. I also marked the language analysis applied on the index. Some of the languages in $E\chi\alpha\mu s$ are low-resource ones, especially the ones from the Balto-Slavic family, which is also clear from their Wikipedia sizes. In the table, we see that half of the languages have under one million articles, and Albanian even falls under 100K. Moreover, even more languages are comparable with the number of sentences in the ARC Corpus, which is also built from science books. Finally, some of the languages (Serbian, Croatian, Macedonian, and Albanian) are not processed with any language-specific ElasticSearch analyzers.

5.5.3 Baseline Models

I divide my baselines into the following two categories: (*i*) models without additional training, and (*ii*) fine-tuned models. The first group contains common baselines, i.e., random guessing and information retrieval solver (Clark et al., 2016). In addition, I evaluate the knowledge contained in the pre-trained language model,

[13] I used the official Wiki dumps from March 2020 for all languages. http://dumps.wikimedia.org/

[14] The statistics are extracted from http://meta.wikimedia.org/wiki/List_of_Wikipedias

i.e., mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), and I use it as an answering mechanism. The second group of baselines compare the learning ability of state-of-the-art multilingual models on the task of multiple-choice question answering. Since I have multi-choice questions, I adopt accuracy as an evaluation measure, as this is standard for this setup.

No Additional Training

Information Retrieval IR This IR baseline is from Clark et al. (2016), and it ranks the possible options o for each question q based on the relevance score returned by a search engine.[15] In particular, for each option o_i, I form a query by appending the option's text to the question's $(q + o_i)$, and I send this concatenation to the search engine. I then sum the returned scores for the top-10 hits, and I predict the choice with the highest score to be the correct answer. More detailed discussion can be found in Section 5.5.2.

Pre-trained Model as a Knowledge Base (KB) As we start to understand pre-trained BERT-like models better (Petroni et al., 2019; Rogers et al., 2020), we observe some interesting phenomena. Here, I evaluate the knowledge contained in the model by leveraging the standard masking mechanism used in pre-training. I tokenize each question-option pair into subwords, and then I replace all the pieces from the option with the special [MASK] token. Following the notation from Devlin et al. (2019), the input sequence can be written as follows:

[CLS] [Q_1] ... [Q_N] [M_O_1] ... [M_O_M] [SEP],

where Q is the question, and M_O is the masked option. Following the notation above, I obtain a score for each option in the question based on the normalized log-probability for the entire masked sequence. (see Eq. 5.9).

$$score(O_i) = \frac{1}{|O_i|} \sum_{t \in O_i} \log P_{MLM}(t|Q) \tag{5.9}$$

I could probably obtain better results for that evaluation if I form the question-option pairs as a single statement, e.g., "What is the purpose of *something*? [SEP] [M_O] → The purpose of *something* is [M_O]."

Fine-Tuned Models

I am interested in evaluating the ability of pre-trained models to transfer science-based knowledge across languages when fine-tuned.

In order to evaluate the QA capability of these models, I follow the established approach in this setting (Devlin et al., 2019; Liu et al., 2019; Sun et al., 2019), and

[15] I build and use a separate index for each language using ElasticSearch.

I fine-tune them to predict the correct answer in a multi-choice setting, given a selected context. This setup feeds the pre-trained model with a text, tokenized using the corresponding tokenizer for the model in the format:

[CLS] C [SEP] Q + O [SEP],

where C, Q and O are the tokenized *knowledge context* (see Section 5.5.2), the *question*, and the *option*, respectively. Each question-option pair (Q+O) is evaluated, and the one with the highest confidence of being an answer is selected.

In my experiments, I used the Transformers library (Wolf et al., 2020). I experimented with the best-performing multilingual models: the Multilingual version of BERT, or mBERT Devlin et al. (2019), and the recently proposed XLM-RoBERTa, or XLM-R (Conneau et al., 2020).

Multilingual BERT (Devlin et al., 2019) is a fundamental multilingual model trained on 104 languages with a vocabulary of 110K word-pieces, with a total of 172M parameters (12 layers, 768 hidden states, 12 heads).

XLM-RoBERTa (Conneau et al., 2020) is a recent multilingual model based on RoBERTa (Liu et al., 2019). It is trained on 100 languages, with a larger vocabulary of 250K sentence pieces. It comes in two sizes: *XLM-R$_{Base}$* (270M parameters, same architecture as mBERT, except vocab size), and *XLM-R* (550M parameters, 24 layers, 1,024 hidden states, 16 heads). For completeness, I include both in my experiments.

I fine-tuned the aforementioned models following the standard procedure for multiple-choice comprehension tasks, as described in (Devlin et al., 2019) and (Liu et al., 2019), using the Transformers library (Wolf et al., 2020). The training details can be found in Appendix B.1.1.

5.5.4 Experiments and Results

In this section, I evaluate the performance of the baseline models described in Section 5.5.3 on the *Eχαμs* dataset. In Table 5.14, I show the overall per-language performance of the evaluated models. The first group shows simple baselines: random guessing and IR over Wikipedia articles. IR is better than random guessing, but it is clear that most questions require reasoning beyond simple word matching. In the last group, I evaluate the knowledge contained in the models before and after the QA fine-tuning. First, I evaluate XLM-R as a knowledge base, and then I use the *Full* model but with the question–option pair only.

Multilingual Evaluation

The next two groups show (*i*) how continuous fine-tuning of XLM-R on multi-choice machine reading comprehension and multi-choice science QA helps, and (*ii*) how the different models (XLM-R, XLM-R$_{\text{Base}}$, and mBERT) compare. I follow a standard training scheme for such tasks: first I fine-tune on RACE (Lai et al., 2017) (∼85k EN

	ARC		R12	*Eχαμs*																
Lang/Set	**E**	**C**	**en**	**ar**	**bg**	**de**	**es**	**fr**	**hr**	**hu**	**it**	**lt**	**mk**	**pl**	**pt**	**sq**	**sr**	**tr**	**vi**	**All**
Random Guess	25.0	25.0	25.0	25.0	25.0	29.4	32.0	29.4	26.7	27.7	26.0	25.0	25.0	25.0	25.0	25.0	26.2	23.1	25.0	25.9
IR (Wikipedia)	-	-	-	31.0	29.6	29.3	27.2	32.1	31.9	29.7	27.6	29.8	32.2	29.2	27.5	25.3	31.8	28.5	27.5	29.5
XLM-R on RACE	61.6	45.9	57.4	39.1	43.9	37.2	40.0	37.4	38.8	39.9	36.9	40.5	45.9	33.9	37.4	42.3	35.6	37.1	35.9	39.1
w/ SciENs	**73.6**	51.2	68.4	39.1	44.2	35.5	37.9	37.1	38.5	37.9	39.5	**41.3**	49.8	36.1	**39.3**	42.5	37.4	37.4	35.9	39.6
then on *Eχαμs* (Full)	72.8	**52.6**	**68.8**	**40.7**	**47.2**	**39.7**	**42.1**	**39.6**	**41.6**	**40.2**	**40.6**	40.6	**53.1**	**38.3**	38.9	**44.6**	**39.6**	**40.3**	**37.5**	**42.0**
XLM-R_{Base} (Full)	54.2	36.4	54.6	34.5	35.7	*36.7*	38.3	*36.5*	*35.6*	*33.3*	33.3	33.2	41.4	*30.8*	29.8	*33.5*	32.3	30.4	*32.1*	34.1
mBERT (Full)	*63.8*	*38.9*	*57.0*	34.5	*39.5*	35.3	*40.9*	34.9	35.3	32.7	*36.0*	*34.4*	*42.1*	30.0	29.8	30.9	*34.3*	*31.8*	31.7	*34.6*
mBERT (*Eχαμs* only)	39.6	28.5	35.1	31.9	34.1	30.4	37.9	33.3	32.6	29.3	31.1	31.9	42.4	29.0	28.3	29.9	30.8	25.4	30.0	31.7
XLM-R as KB	30.8	26.2	27.2	31.0	27.2	31.7	37.9	29.9	27.6	29.3	28.0	28.3	23.5	24.6	27.0	25.6	25.4	24.4	24.9	27.0
XLM-R (Full) w/o ctx	45.4	39.2	47.6	30.2	34.8	34.3	30.2	33.0	33.6	33.4	28.5	30.9	37.5	30.0	32.4	36.7	32.1	31.7	30.4	32.8

Table 5.14: Overall per-language evaluation. The first three columns show the results on ARC Easy (E), ARC Challenge (C), and Regents 12 LivEnv (en). The following columns show the per-language and the overall results (the last column All) for all languages. *All* is the score averaged over all *Eχαμs* questions.

questions over documents), then on the AI2 English science datasets (I call them SciENs for shorter), including ∼9k EN questions with provided relevant contexts[16], and, finally, on the multilingual training set (see Section 5.5.1) with retrieved relevant contexts from Wikipedia (see Section 5.5.2), which is my desired multilingual evaluation setting and I call it *Full*. We can also see that training on the SciENs, which has mostly primary school questions from Natural Sciences, only yields +0.5% improvement on *Eχαμs*. Nevertheless, we see a 2.4% improvement with multilingual fine-tuning on *Eχαμs* and +0.5% for English. In the third group, I compare the results from mBERT, XLM-R_{Base}, and XLM-R after fine-tuning. Increasing the capacity of the model yields improvements: XLM-R scores 7.4% higher on *Eχαμs*, and more than 14% on English datasets, compared to its base version (XLM-R_{Base}). However, mBERT and XLM-R_{Base} have close performance, with mBERT having a small advantage in the multilingual setting.

Finally, I fine-tuned mBERT on *Eχαμs* only. As expected, the performance drops by 3% absolute compared to the *Full* setup.

Knowledge Evaluation

The last two rows of Table 5.14 evaluate the knowledge in the best model, namely XLM-R. With *XLM-R as KB* (see Section 5.5.3) we see small improvement over the random baseline: +5% ARC Easy, 2% on R12, and just +1% on *Eχαμs* and ARC Challenge. Furthermore, I evaluate the knowledge contained in the model after the *Full* fine-tuning by excluding the relevant knowledge context (*ctx*). This is better than the *XLM-R as KB*, but it still achieves inferior overall results, which shows that the stored knowledge is not enough, and that I need to explicitly obtain additional knowledge from an external source.

[16] I use the data described at http://leaderboard.allenai.org/arc/submission/blcotvl7rrltlue6bsv0

Lang	A_E	A_{Ch}	R12	de	es	fr	it	pt	bg	hr	lt	mk	pl	sr	hu	sq	tr	vi	ar
en_{all}	73.6*	51.2*	68.4*	35.5*	37.9	37.1	39.5	39.3	44.2	38.5	41.3	49.8	36.1	37.4	37.9	42.5	37.4	35.9	39.1
w/ it	+1.4	+1.3	+1.4	+6.2	**+4.2***	+0.3*	-	-3.7*	+1.2	+4.1	+0.9	+0.8	+1.5	+3.1	+2.8	+0.9	-1.3	**+1.8**	+1.8
w/ pt	+0.1	+1.2	-0.8	+2.2	+2.5*	-2.5*	+1.4*	-	+0.3	0.0	+2.0	+0.8	-0.1	-0.6	-0.6	-1.3	**+1.3**	+0.6	+1.1
w/ bg	+0.6	+0.4	-0.4	**+3.6**	+0.8	+1.6	**+3.4**	-1.9	-	+1.5*	**+2.9***	**+1.6***	+0.1*	**+1.5***	+2.0	**+2.3**	-0.9	-0.8	+0.8
w/ hr	+1.1	**+1.7**	-0.2	+4.8	+3.8	+0.3	+5.8	-2.8	+1.7*	-	+0.2*	-0.1*	+1.2*	+6.7*	+2.8	+1.7	+1.2	+0.5	-0.1
w/ mk	+1.5	-0.5	**+2.2**	+1.0	**+4.2**	-0.3	+2.0	-2.6	+1.8*	**+3.9***	+1.5*	-	+1.9*	0.0*	+2.0	+6.9	+4.8	+0.5	**+4.5**
w/ pl	-2.0	-1.5	-3.1	0.0	+0.4	-2.5	+0.1	-1.3	+1.1*	+1.0*	-0.5*	-0.2*	-	0.0*	-0.4	+0.3	+0.2	-1.4	+0.9
w/ sr	**+1.8**	-0.1	-1.2	+2.6	+5.1	+1.9	+2.8	-0.6	**+2.2***	+6.2*	+0.2*	+1.3*	+1.3*	-	+1.4	-0.4	-0.7	-1.0	+3.2
w/ hu	-0.8	-0.8	-1.0	+7.8	+10.2	+2.8	+1.1	-1.9	+0.7	+0.8	-3.2	+0.1	+0.9	+0.9	-	-0.2	-0.2	-0.6	-1.4
w/ sq	-0.1	+0.3	-1.5	+3.5	-0.5	-0.6	+0.8	+0.9	+0.9	+0.8	+1.0	+3.4	+0.6	+0.6	+1.9	-	+0.4	+0.3	+0.2
w/ tr	-0.5	+1.1	-1.5	+1.5	+3.0	-1.9	+2.3	-3.0	+1.0	+1.0	-2.7	+1.5	+0.2	+1.2	**+2.4**	+3.7	-	-1.0	+1.8
w/ vi	-0.5	+0.4	-0.8	+2.9	+3.4	**+4.1**	+1.1	**+1.1**	+1.5	+1.7	+0.4	+0.4	**+2.1**	0.0	+1.7	+0.8	+1.1	-	+3.4

Table 5.15: Cross-lingual zero-shot performance on $E\chi\alpha\mu s$. The first three columns show the performance on the test set of the AI2 science datasets (English), followed by per-language evaluation. The underlined values mark languages that have parallel data with the source language, and the ones with an asterisk* are from the same family.

Cross-lingual Evaluation

Table 5.15 shows the results from the cross-lingual zero-shot transfer compared to the English-only baseline en_{all}, from XLM-R fine-tuned on SciEN. The languages are ordered by family, and then alphabetically. I further fine-tune on a single source language and I test on all other languages using the splits described in Subsection 5.5.1. The results show that the additional fine-tuning on a single language is mostly positive. This is notable when fine-tuning on a language with similar linguistic characteristics to the target language, e.g., Balto-Slavic: bg-sr, hr-mk, pl-mk, sr-bg.

We also see gains when the source language contains more questions from largely represented and harder subjects. Examples of such are the experiments showing the positive effects of training on Vietnamese and Macedonian as source languages; they both contain such subjects: Biology, History, Chemistry, Physics, and Geography.

This is an indication that the knowledge from the same or from related subjects in a non-related language is preferred over knowledge from non-related subjects from a related language. For the same reasons, Portuguese and Polish show negative effects of fine-tuning on some of the target languages. They contain mostly niche subjects such as Professional, Philosophy, Economics, Geology. We see a noticeable drop in accuracy for Portuguese almost everywhere, but it has positive effect on languages that contain similar subjects (Biology, Economics) or are from the same language family such as Spanish and Italian (for Portuguese). We see the opposite in the Lithuanian-Polish pair, languages from the same family (but different subjects) have negative, or no effect on each other. Finally, I analyze the results from language pairs containing parallel examples (the underlined values). Such pairs show consistent improvement (+5 to +10), which suggests that the model

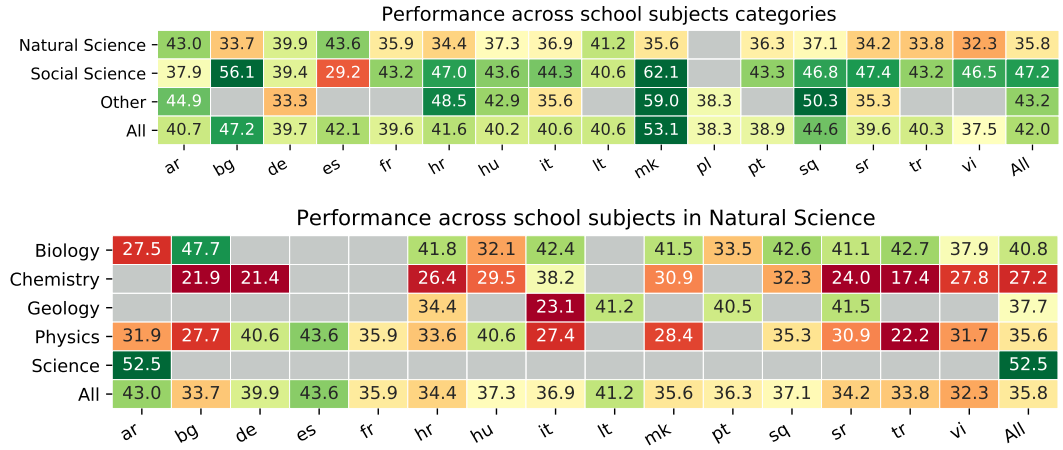


Figure 5.7: Fine-grained evaluation by language and school subjects.

learns to align the parallel knowledge from the source language to the target language. However, I also must note that the effect is strongly dependent on the size of the overlapping sets.

5.5.5 Per-Subject Fine-Grained Evaluation

Fine-grained evaluation (Mihaylov and Frank, 2019; Xu et al., 2020) allows an in-depth analysis of the question answering models. One of the nice features of *Εχαμς* is that it supports subject-related fine-grained evaluation. On Figure 5.7, the results are shown by subject group and per-subject for Natural Science.

We can see that the Natural Science questions are the most challenging ones, which is mostly due to Chemistry and Physics. Those questions require very complex reasoning and knowledge such as understanding physical models, processes and causes, comparisons, algebraic skills and multi-hop reasoning (see Section 5.5.1). These skills are currently beyond the capabilities of the current QA models, and pose interesting challenges for future work (Welbl et al., 2018; Yang et al., 2018; Saxton et al., 2019; Lample and Charton, 2020). Informatics is another challenging subject, as it requires understanding programming code and positional numerical systems among others.

Figure 5.8 shows fine-grained evaluation for two subject groups: *Social Science* and *Others*. We can see that these subjects are less challenging than Natural Science. One reason is that many of the subjects in these two groups such as Business & Economics, Geography, and History can be answered using knowledge that is easily accessible in sources such as Wikipedia (e.g., *"Who was the first prime minister of Poland after 1990?"*), i.e., without the need for complex reasoning or calculations, which are often needed in order to answer questions in subjects such as Physics and Chemistry. Nevertheless, while seeing scores as high as 60% for some subjects

Figure 5.8: Fine-grained evaluation by language and school subjects in *Social Science* and *Other*.

and languages, the current multilingual QA models are still far from perfect, which leaves a lot of room for improvement.

5.5.6 Discussion

My results show that initial fine-tuning on a large monolingual out-of-domain multi-choice machine reading comprehension dataset (RACE (Lai et al., 2017)) performs much better than *no training* baselines for answering multilingual *Exams* questions. Moreover, additional training on English science QA in lower school levels has no significant effect on the overall accuracy. These results suggest that further investigation of fine-tuning with other multilingual datasets (Gupta et al., 2018; Lewis et al., 2020; Clark et al., 2020; Efimov et al., 2020; d'Hoffschmidt et al., 2020; Artetxe et al., 2020; Longpre et al., 2021) is needed in order to understand the domain transfer benefits to science QA in *Exams*, even if they are not in a multi-choice setting (Khashabi et al., 2020). Using *domain-adaptive* and *task-adaptive pre-training* (Gururangan et al., 2020) to the multilingual science QA might offer further potential benefits.

Moreover, we need a better knowledge context for a given question–choice pair (the last row in Table 5.14). Knowing that the context retrieved from the noisy Wikipedia corpus is relevant for answering *Exams* questions, suggests that we need a better multilingual science corpus, similar to Clark et al. (2018); Pan et al. (2019);

Bhakthavatsalam et al. (2020). We further need better multilingual knowledge selection and ranking (Banerjee et al., 2019). Finally, my cross-lingual experiments show that we can align the knowledge between languages from parallel examples, which poses a new question: *Is it only due to keyword matching or could the model align full sentences?*

5.6 Summary

In this chapter, I presented a study on automating customer support on Twitter using two types of models: (*i*) retrieval-based (IR with BM25), and (*ii*) based on generative neural networks (Seq2seq with attention and Transformer). I evaluated these models without the need of human judgments, using evaluation measures based on (*i*) word-overlap (BLEU@2 and ROUGE-L), and (*ii*) semantics (Embedding Average, Greedy Matching, and Vector Extrema). For my experiments, I have divided the data by the timestamp of the post in order to simulate a real-world scenario. My experiments showed that generative neural models outperform retrieval-based ones, but they struggle when very few examples for a particular topic are present in the training data. Nonetheless, despite showing good results and being able to generate grammatically correct answers and mostly relevant to the question answers, the data provided only from chat logs is not enough to build an end-to-end customer support bot. It is due to the evolving nature of customer issues, while being accurate when they were posted, they tend to become obsolete with time.

Further, I have presented a novel framework for re-ranking answer candidates for conversational agents. In particular, I adopted techniques from the domain of machine reading comprehension (Chen et al., 2017; Seo et al., 2017; Yu et al., 2018) to evaluate the quality of a question–answer pair. My framework consists of two tasks: (i) an auxiliary one, aiming to fit a goodness classifier using QANet and negative sampling, and (ii) a main task that re-ranks answer candidates using the learned model. I further experimented with different model sizes and two types of embedding models: GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018). My experiments showed improvements in answer quality in terms of word-overlap and semantics when re-ranking using the auxiliary model. Last but not least, I argued that choosing the top-ranked answer is not always the best option. Thus, I introduced probabilistic sampling that aims to diversify the agent's language and to up-vote the popular answers, while taking their ranking scores into consideration.

Finally, I presented $E\chi\alpha\mu s$, a new challenging cross-lingual and multilingual benchmark for science QA in 16 languages and 24 subjects from high school examinations. I further proposed new fine-grained evaluation that allows precise comparison across different languages and school subjects. I performed various experiments and analysis with pre-trained multilingual models (XLM-R, mBERT),

and I demonstrated that there is a need for better reasoning and knowledge transfer in order to solve some of the questions from $E\chi\alpha\mu s$. I hope that my publicly available data and code will enable work on multilingual models that can reason about question answering in the challenging science domain.

Chapter 6

Conclusion and Future Work

6.1 Summary

In this thesis, I took up on the problem of building efficient task-oriented conversational agents for customer support. This is a complex problem and often requires a pipeline built on top of a modularized dialogue system that combines different components and models working in synchrony. I investigated several important components natural language understanding for slot tagging and intent detection, information retrieval from external knowledge sources, question answering system, text generation and re-ranking models. Moreover, my research is not limited to producing short-forms answers in English, but I also investigated multilingual and cross-lingual approaches for multiple-choice question answering, and retrieval of long-form documents and articles that can serve as explanations. Below, I offer a summary of each chapter.

In the first chapter, I started by presenting my motivation for working on building dialogue agents for customer support and the main challenges that these systems face. I further highlighted key concepts and I defined the NLP tasks that I explored through the following chapters. Moreover, I illustrated the main components in a chatbot pipeline and the information flow between them. Finally, I summarized the thesis aims and I outlined the research objections that originate from these aims.

Next, I presented an in-depth discussion of prior work on conversational agents and their application for customer support. I organized the section into six major categories: (*i*) task-oriented conversational agents, (*ii*) question answering approaches, (*iii*) retrieval of long-form explanations from external sources, (*iv*) end-to-end generative models for dialog, and finally (*v*) strategies for response selection from multiple sources.

In the next three chapters, I described in detail my ideas, the proposed frameworks, the approaches, and the collected datasets.

In Chapter 3, I extended and improved on state-of-the-art approaches for joint slot filling and intent classification on two well-known English datasets: ATIS and SNIPS. My proposed model incorporates hand-crafted features and attention-based mechanism in order to jointly reinforce both tasks, achieving intent accuracy of 97.87 (ATIS) and 98.86 (SNIPS). Or as a relative error reduction, it achieved almost 5% on ATIS, and over 40% on SNIPS, compared to the state of the art. In terms of slot filling, my models achieved an F1 score of 96.25 (+6.25%) for ATIS, and 96.57 (+37.64%) for SNIPS.

In Chapter 4, I studied approaches for curating answers from external knowledge sources. In particular, I focused on two main directions: (*i*) knowledge retrieval from inverted index with contextual passages for zero-shot multiple-choice question answering, (*ii*) answer retrieval from a pool of explanations, i.e., finding previously curated long-form answers to produce a detailed (long-form) answer to an user query. In particular, I explored the limitations of zero-shot transfer from English to Bulgarian on a newly collected dataset for multiple-choice question answering, showing improvements of more than 12.5% in terms of accuracy over strong baselines. Moreover, I proposed an approach to extract long-form documents that can serve as explanations based on self-adaptive training and distant supervision, achieving 90.3 MAP@5 and improving current state of the art by two points. Furthermore, I collected a new English dataset containing more than 330,000 unlabeled article–claim pairs from crowd fact-checkers, resulting in training sets of 3.5–50K high-quality examples[1], which is 1-2 orders of magnitude larger than pre-existing datasets.

Finally, in Chapter 5, I presented three directions for advanced conversation: (*i*) end-to-end generative conversational agents, (*ii*) strategies for combining answers from different sources or models, and (*iii*) multilingual and cross-lingual knowledge transfer. First, I explored the capabilities of sequence-to-sequence neural network models, both attentive RNNs and Transformers as a backbone of an end-to-end conversational agent for customer support, I demonstrated promising results compared to classical information retrieval models. Next, I proposed a novel re-ranking approach for response selection from multiple sources based on state-of-the-art QA model in combination of using pre-trained GloVe or ELMo embeddings to encode the words and negative sampling in order to find the most relevant answer from the set of candidates. Finally, I introduced the largest multilingual dataset for multiple-choice QA ($E\chi\alpha\mu s$), containing more than 24K questions in 24 subjects from high-school matriculation exams. The dataset covers a diverse set of languages: a total of sixteen languages from eight language families. Moreover, I performed various experiments and analysis with pre-trained multilingual models (XLM-R, mBERT), and I demonstrated that there is a need for better reasoning and knowledge transfer in order to solve some of the questions from $E\chi\alpha\mu s$.

[1] I estimate 90K correct article–claim pairs to be in this dataset (see Section 4.3.4).

6.2 Contributions

The key contributions of this thesis are as follows:

- **Exploring new models and algorithms**:
  - I proposed a novel enriched pre-trained language model to jointly model the tasks of intent detection and slot filling, namely, *Transformer-NLU*. Moreover, I designed a pooling attention layer in order to obtain intent representation beyond just the pooled one from the special start token. Further, I reinforced the slot filling with word-specific features, and the predicted intent distribution. My experiments on two standard datasets showed that Transformer-NLU outperforms other alternatives for all standard measures used to evaluate NLU tasks.
  - I proposed an approach for training from noisy data using self-adaptive learning and additional weights in the loss function. Furthermore, I demostrated the utility of the data collected and labeled using distant supervision (CrowdChecked), which yielded sizable performance gains of four points in terms of MRR, P@1, and MAP@5 over strong baselines that are trained on manually annotated data (Shaar et al., 2021). Moreover, I demonstrated improvements over the state of the art on the CheckThat '21 dataset by two points, achieving MAP@5 of 90.3, when using CrowdChecked and my newly proposed pipeline.
  - I designed an end-to-end approach the task of multiple-choice reading comprehension for low-resource languages. The model is built on top of a multilingual BERT model (Devlin et al., 2019), which I fine-tuned on large-scale English reading comprehension corpora, and open-domain commonsense knowledge sources (Wikipedia). My main experiments evaluated the model when applied to Bulgarian in a zero-shot fashion.
  - I developed an approach for automating customer support on Twitter using two types of models: (*i*) retrieval-based (IR with BM25), and (*ii*) based on generative neural networks (seq2seq with attention and Transformer). I evaluated these models without the need for human judgements, using evaluation measures based on (*i*) word-overlap (BLEU@2 and ROUGE-L), and (*ii*) semantics (Embedding Average, Greedy Matching, and Vector Extrema). My experiments showed that generative neural models outperform retrieval-based ones, but they struggle when very few examples for a particular topic are present in the training data.
  - I introduced a novel framework for re-ranking answer candidates for conversational agents. In particular, I adopted techniques from the domain of machine reading comprehension (Chen et al., 2017; Seo et al., 2017; Yu et al., 2018) to evaluate the quality of a question–answer pair.

My framework consists of two tasks: (i) an auxiliary one, aiming to fit a goodness classifier using QANet and negative sampling, and (ii) a main task that re-ranks answer candidates using the learned model. I further experimented with different model sizes and two types of embedding models: GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018). My experiments showed improvements in answer quality in terms of word overlap and semantics when re-ranking using the auxiliary model.

  - I designed a new challenging cross-lingual and multilingual benchmark for science QA from high school examinations. I evaluated the abilities of state-of-the-art models for zero-shot and cross-lingual transfer in massively multilingual settings. I showed that pre-training on large English out-of-domain datasets can help the model to learn the task, but further improvements can only be achieved by in-domain multilingual data.
  - I performed various experiments and analysis with pre-trained multilingual models (XLM-R, mBERT), and I demonstrated that there is a need for better reasoning and knowledge transfer in order to solve some of the questions from $E\chi\alpha\mu s$.

- **Creating new datasets**:
  - I collected a new Bulgarian corpus for multiple-choice reading comprehension with 2,633 questions from matriculation exams for twelfth grade in history and biology, and online exams in history without explanatory contexts.
  - I collected $E\chi\alpha\mu s$, a new challenging cross-lingual and multilingual benchmark for science QA in 16 languages and 24 subjects from high school examinations. I further proposed new fine-grained evaluation that allows precise comparison across different languages and school subjects.
  - I built CrowdChecked, a large-scale dataset for detecting previously fact-checked claims, with more than 330,000 pairs of tweets and corresponding fact-checking articles posted by crowd fact-checkers. I further investigated two techniques for labeling the tweet–article pairs using distance supervision, based on Jaccard similarity and the predictions from a neural network model resulting in new training sets of 3.5K–50K examples.

6.3 Directions for Future Research

Modularized (task-oriented) conversational agents offer a great flexibility in terms of model training, and allow to easily add new or to replace existing modules to the agent's pipeline. However, that flexibility brings several limitations along.

On one hand, there is a disconnect between different components (models) both during training and inference, that, in turn, leads to error accumulation along the pipeline (a snowball effect[2]), e.g., if the natural language understanding recognizes the wrong intent, this will result in different dialogue policy, and generating the wrong response. On the other hand, including too many components can increase the computational cost, hence deploying the dialogue system can become infeasible. Hereby, I believe that building end-to-end conversational agents is an exciting research direction that can help to overcome the limitations of module-based conversational agents.

In the short term, end-to-end differentiable architectures (Li et al., 2017; Bordes et al., 2017; Madotto et al., 2018) based on a combination of hierarchical neural networks, multi-task learning, and multi-model error propagation can be a step forward in that direction. Nonetheless, it is not always necessary to have one model per task in the pipeline, and simplifying the model architecture is another important research direction. A possible approach is to merge several modules together (Zhao and Eskenazi, 2016; Lei et al., 2018), but a more promising one is to train an expert model that can perform multiple tasks together, e.g., using multiple classification heads Weld et al. (2022), prompting (Raffel et al., 2020; Su et al., 2022; Sanh et al., 2022), or even demonstrations (Brown et al., 2020).

In the long term, in my opinion, single-model architectures based on end-to-end generative models can be a strong alternative to multi-model pipelines, even in task-oriented scenarios. Moreover, the release of several large pre-trained Transformers, trained on sequence generation (Raffel et al., 2020; Lewis et al., 2020; Liu et al., 2020; Xue et al., 2021) tasks is fostering further research in the direction of generative conversational agents. However, as I discuss in Chapter 5, there is currently plenty of room for improvements of these models. Moreover, even in an end-to-end scenario the task-oriented dialogue system will still need to communicate with external knowledge sources such as databases or APIs, in order to collect the information needed to serve the customer's request.

Nevertheless, even with the development of models with capacity increased to billions of trainable parameters (Shoeybi et al., 2019; Brown et al., 2020; Sun et al., 2021; Fedus et al., 2022; Zhang et al., 2022; Soltan et al., 2022), they are not widely adopted for real-world applications. The main thing hindering their usage is that they are vulnerable to both ethical and practical risks (Bender et al., 2021; Bommasani et al., 2021) such as hallucination (Dziri et al., 2021; Shuster et al., 2021; Dziri et al., 2022; Ji et al., 2022), injected biases both during pre-training or fine-tuning, e.g., learning toxic content (Ousidhoum et al., 2021; Zhou et al., 2021), or different cultural norms (Arora et al., 2022), among others. That being said, it is clear that we need more research and better models in order to release end-to-end models in dynamic real-world scenarios. Moreover, it is not enough to develop efficient

[2]https://dictionary.cambridge.org/dictionary/english/snowball-effect

mechanisms for updating the factual knowledge stored in the model itself (De Cao et al., 2021), but also to implement additional knowledge grounding (Zhao et al., 2020), and auto-debiasing (Guo et al., 2022) procedures, in order to ensure that the chatbots produce correct and factual responses. Finally, we need to develop mechanisms that prevent malicious actors to exploit the models (Vincent, 2016; Hancock et al., 2019; Vanderlyn et al., 2021).

In recent years, we see a growth of work focusing on explaining the decisions that the models make, in order to arrive at their outputs, and it is now becoming an important research area both in NLP (Danilevsky et al., 2020) and in other divisions of artificial intelligence (Došilović et al., 2018). The same trend holds for dialogue agents, especially in the cases when the system is expected to provide not only a confirmation, a short-form clarification question or an answer, but also an explanation that the user can use to achieve their goals, or to take an informed decision based on facts, which they can verify as well. This is of vital importance in domains where mistakes can have high cost such as in medical scenarios or in business applications.

In Chapter 4, I focused on retrieving answers from long-form documents that can serve as an explanation, but there are several interesting research directions that can be explored in future work. One promising direction is explanbility based on the reasoning chain that the model followed in order to generate the answer (Yang et al., 2018; Das et al., 2018). Another direction is forming long-form answers with detailed explanations based on evidence paragraphs (Kwiatkowski et al., 2019; Fan et al., 2019) and further enriching them on the fly with automatic edits, adding sources, etc. (Schick et al., 2022), or obtaining token-level explanations (Li and Yao, 2021; Arora et al., 2022).

Declaration of Authorship

I hereby declare that this dissertation contains original results obtained by me with the support and the assistance of my supervisors. The results, obtained by other scientists, are described in detail and cited in the bibliography. This dissertation has not been previously submitted for a degree or any other qualification at another University or any other institution.

Signed:

_____

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

[2] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

[3] Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively Multilingual Neural Machine Translation. In *Proceedings of the Conference of the North American Chapter of ACL*, NAACL-HLT '19, pages 3874–3884, Minneapolis, Minnesota, USA.

[4] Arnav Arora, Lucie-Aimée Kaffee, and Isabelle Augenstein. 2022. Probing Pre-Trained Language Models for Cross-Cultural Differences in Values. *arXiv preprint arXiv:2203.13722*.

[5] Siddhant Arora, Danish Pruthi, Norman Sadeh, William W. Cohen, Zachary C. Lipton, and Graham Neubig. 2022. Explain, Edit, and Understand: Rethinking User Study Design for Evaluating Model Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5277–5285.

[6] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 4623–4637, Online.

[7] Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

[8] Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual Extractive Reading Comprehension by Runtime Machine Translation. *arXiv preprint arXiv:1809.03275*.

[9] Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China.

[10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations*, ICLR '15, San Diego, California, USA.

[12] Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent Neural Methods on Dialogue State Tracking for Task-Oriented Dialogue Systems: A Survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251, Singapore and Online.

[13] Ramy Baly, Georgi Karadzhov, Jisun An, Haewoon Kwak, Yoan Dinkov, Ahmed Ali, James Glass, and Preslav Nakov. 2020. What Was Written vs. Who Read It: News Media Profiling Using Text Analysis and Social Media Context. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3364–3374, Online.

[14] Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful Selection of Knowledge to Solve Open Book Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 6120–6129, Florence, Italy.

[15] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

[16] Alberto Barrón-Cedeno, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, et al. 2020. Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 215–236. Springer.

[17] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[18] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, New York, USA.

[19] Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. GenericsKB: A Knowledge Base of Generic Statements. *ArXiv*, abs/2005.00660.

[20] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

[21] Michael Boratko, Harshit Padigela, Divyendra Mikkilineni, Pritish Yuvraj, Rajarshi Das, Andrew McCallum, Maria Chang, Achille Fokoue, Pavan Kapanipathi, Nicholas Mattei, Ryan Musa, Kartik Talamadupula, and Michael Witbrock. 2018. An Interface for Annotating Science Questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, EMNLP '18, pages 102–107, Brussels, Belgium.

[22] Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning End-to-End Goal-Oriented Dialog. In *Proceedings of the 2017 International Conference on Learning Representations*, ICLR '17, Toulon, France.

[23] Mostafa Bouziane, Hugo Perrin, Aurélien Cluzeau, Julien Mardas, and Amine Sadeq. 2020. Team Buster. ai at CheckThat! 2020 Insights and Recommendations to Improve Fact-Checking. In *CLEF (Working Notes)*.

[24] Samuel Bowman. 2022. The Dangers of Underclaiming: Reasons for Caution When Reporting How NLP Systems Fail. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7484–7499, Dublin, Ireland.

[25] Martin Boyanov, Preslav Nakov, Alessandro Moschitti, Giovanni Da San Martino, and Ivan Koychev. 2017. Building Chatbots from Forum Data: Model Selection Using Question Answering Metrics. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 121–129, Varna, Bulgaria.

[26] Alex Boyd, Raul Puri, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Large Scale Multi-Actor Generative Dialog Modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 66–84, Online.

[27] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

[28] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91.

[29] Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Automatic Spanish Translation of SQuAD Dataset for Multi-lingual Question Answering. In *Proceedings of the 12th Language Resources and Evaluation Conference*, LREC '20, pages 5515–5523, Marseille, France.

[30] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1870–1879, Vancouver, Canada.

[31] Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

[32] Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. 2017. A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1925–1935, Vancouver, Canada.

[33] Anton Chernyavskiy, Dmitry Ilvovsky, Pavel Kalinin, and Preslav Nakov. 2022. Batch-Softmax Contrastive Loss for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 116–126, Seattle, Washington, USA.

[34] Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021. Aschern at CLEF CheckThat! 2021: Lambda-Calculus of Fact-Checked Claims. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 484–493.

[35] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass.

2022. DiffCSE: Difference-based Contrastive Learning for Sentence Embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, Washington, USA.

[36] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

[37] Peter Clark. 2015. Elementary School Science and Math Tests as a Driver for AI: Take the Aristo Challenge! In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*, AAAI '15, pages 4019–4021, Austin, Texas, USA.

[38] Peter Clark, Niranjan Balasubramanian, Sumithra Bhakthavatsalam, Kevin Humphreys, J. Clint Kinkead, and Ashish Sabharwal. 2014. Automatic Construction of Inference-Supporting Knowledge Bases. In *Proceedings of 4th Workshop on Automated Knowledge Base Construction*, AKBC '14, Montreal, Canada.

[39] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

[40] Peter Clark, Oren Etzioni, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz. 2019. From 'F' to 'A' on the N.Y. Regents Science Exams: An Overview of the Aristo Project. *ArXiv*, abs/1909.01958.

[41] Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions. In *Proceedings of the 13th AAAI Conference on Artificial Intelligence*, AAAI '16, pages 2580–2586, Phoenix, Arizona, USA.

[42] Christopher Clarke, Joseph Peper, Karthik Krishnamurthy, Walter Talamonti, Kevin Leach, Walter Lasecki, Yiping Kang, Lingjia Tang, and Jason Mars. 2022. One Agent To Rule Them All: Towards Multi-agent Conversational AI. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3258–3267, Dublin, Ireland.

[43] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke

Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 8440–8451, Online.

[44] Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems 32*, NIPS '19, pages 7059–7069, Vancouver, Canada.

[45] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 2475–2485, Brussels, Belgium.

[46] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

[47] Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. 2015. Strategic dialogue management via deep reinforcement learning. *arXiv preprint arXiv:1511.08099*.

[48] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In *Proceedings of the Association for Computational Linguistics 2017, System Demonstrations*, ACL '17, pages 97–102, Vancouver, Canada.

[49] Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. 2018. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. *Alexa Prize Proceedings*.

[50] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 2978–2988, Florence, Italy.

[51] Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything Happens for a Reason: Discovering the Purpose of Actions in Procedural Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP '19, pages 4496–4505, Hong Kong, China.

[52] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China.

[53] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *International Conference on Learning Representations*.

[54] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing Factual Knowledge in Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic.

[55] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada.

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186, Minneapolis, Minnesota, USA.

[57] Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French Question Answering Dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online.

[58] Bosheng Ding, Junjie Hu, Lidong Bing, Mahani Aljunied, Shafiq Joty, Luo Si, and Chunyan Miao. 2022. GlobalWoZ: Globalizing MultiWoZ to Develop Multilingual Task-Oriented Dialogue Systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1639–1657, Dublin, Ireland.

[59] Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2022. A Survey of Natural Language Generation. *ACM Comput. Surv.* Just Accepted.

[60] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. 2018. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information*

and Communication Technology, Electronics and Microelectronics (MIPRO), pages 0210–0215.

[61] Nouha Dziri, Andrea Madotto, Osmar Zaïane, and Avishek Joey Bose. 2021. Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214, Online and Punta Cana, Dominican Republic.

[62] Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, Washington, USA.

[63] Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy.

[64] Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. SberQuAD – Russian Reading Comprehension Dataset: Description and Analysis. In *Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CLEF '20, pages 3–15, Thessaloniki, Greece.

[65] Hady Elsahar and Matthias Gallé. 2019. To Annotate or Not? Predicting Performance Drop under Domain Shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173, Hong Kong, China.

[66] Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeno, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019. Overview of the CLEF-2019 CheckThat! Lab: automatic identification and verification of claims. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 301–321. Springer.

[67] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany.

[68] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy.

[69] Yifan Fan, Xudong Luo, and Pingping Lin. 2020. A survey of response generation of dialogue systems. *International Journal of Computer and Information Engineering*, 14(12):461–472.

[70] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

[71] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, ASRU '15, pages 813–820, Scottsdale, Arizona, USA.

[72] Andreas Ferreira, William andVlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California.

[73] Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 268–277, Austin, Texas, USA.

[74] Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages. *arXiv preprint arXiv:2204.08582*.

[75] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

[76] Asbjørn Følstad and Marita Skjuve. 2019. Chatbots for Customer Service: User Experience and Motivation. In *Proceedings of the 1st International Conference on Conversational User Interfaces*, CUI '19, New York, New York, USA.

[77] Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *Proceedings of the NIPS Workshop on Modern Machine Learning and Natural Language Processing*, Montreal, Canada.

[78] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 564–569, Minneapolis, Minnesota, USA.

[79] Jianfeng Gao, Michel Galley, and Lihong Li. 2019. *Neural Approaches to Conversational AI: Question Answering, Task-oriented Dialogues and Social Chatbots*. IEEE Xplore.

[80] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic.

[81] Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

[82] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-Gated Modeling for Joint Slot Filling and Intent Prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana.

[83] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the Conference on Language Resources and Evaluation*, LREC '18, Miyazaki, Japan.

[84] Chulaka Gunasekara, Jonathan K. Kummerfeld, Lazaros Polymenakos, and Walter Lasecki. 2019. DSTC7 Task 1: Noetic End-to-End Response Selection. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 60–67, Florence, Italy.

[85] Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Auto-Debias: Debiasing Masked Language Models with Automated Biased Prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1012–1023, Dublin, Ireland.

[86] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

[87] Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. MMQA: A Multi-domain Multi-lingual Question-Answering Framework for English and Hindi. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC '18, pages 2777–2784, Miyazaki, Japan.

[88] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, ACL '20, pages 8342–8360, Online.

[89] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938.

[90] Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, INTERSPEECH '16, pages 715–719, San Francisco, USA.

[91] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-Teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8536–8546.

[92] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from Dialogue after Deployment: Feed Yourself, Chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy.

[93] Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A Richly Annotated Corpus for Different Tasks in Automated Fact-Checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, CoNLL '19, pages 493–503, Hong Kong, China.

[94] Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2022. A Survey on Stance Detection for Mis- and Disinformation Identification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1259–1277, Seattle, Washington, USA.

[95] Momchil Hardalov, Anton Chernyavskiy, Ivan Koychev, Dmitry Ilvovsky, and Preslav Nakov. 2022. CrowdChecked: Detecting Previously Fact-Checked Claims in Social Media. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, AACL-IJCNLP '22, Online.

[96] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards Automated Customer Support. In *Proceedings of the 18th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, AIMSA '18, pages 48–59, Varna, Bulgaria.

[97] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019. Beyond English-Only Reading Comprehension: Experiments in Zero-shot Multilingual Transfer for Bulgarian. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '19, pages 447–459, Varna, Bulgaria.

[98] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019. Machine Reading Comprehension for Answer Re-Ranking in Customer Support Chatbots. *Information*, 10(3).

[99] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2020. Enriched Pre-trained Transformers for Joint Slot Filling and Intent Detection. *arXiv preprint arXiv:2004.14848*.

[100] Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020. EXAMS: A Multi-subject High School Examinations Dataset for Cross-lingual and Multilingual Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 5427–5444, Online.

[101] Amelia Hardy, Ashwin Paranjape, and Christopher Manning. 2021. Effective Social Chatbot Strategies for Increasing User Initiative. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 99–110, Singapore and Online.

[102] Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. ClaimBuster: The First-Ever End-to-End Fact-Checking System. *Proc. VLDB Endow.*, 10(12):1945–1948.

[103] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS Spoken Language Systems Pilot Corpus. In *Speech and Natural Language: Proceedings of a Workshop*, Hidden Valley, Pennsylvania.

[104] Matthew Henderson. 2015. Machine Learning for Dialog State Tracking: A Review. In *Proceedings of The First International Workshop on Machine Learning in Spoken Language Processing*.

[105] Matthew Henderson, Rami Al-Rfou, B. Strope, Yun-Hsuan Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. *ArXiv 1705.00652*.

[106] Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and Accurate Conversational Representations from Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online.

[107] Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, and Michimasa Inaba. 2016. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3146–3150, Portorož, Slovenia.

[108] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[109] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '18, pages 328–339, Melbourne, Australia.

[110] Jian Hu, Gang Wang, Frederick H. Lochovsky, Jian-Tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, pages 471–480, Madrid, Spain.

[111] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. In *Proceedings of Machine Learning Research*, ICML '20, Online.

[112] Jiantao Huang, Yi-Ru Liou, and Hsin-Hsi Chen. 2021. Enhancing Intent Detection in Customer Service with Social Media Data. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 274–275.

[113] Lang Huang, Chao Zhang, and Hongyang Zhang. 2020. Self-Adaptive Training: beyond Empirical Risk Minimization. In *Advances in Neural Information Processing Systems*, volume 33.

[114] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv:1508.01991*.

[115] Chia-Chien Hung, Anne Lauscher, Simone Ponzetto, and Goran Glavaš. 2022. DS-TOD: Efficient Domain Specialization for Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland.

[116] Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Ponzetto, and Goran Glavaš. 2022. Multi2WOZ: A Robust Multilingual Dataset and Conversational Pretraining for Task-Oriented Dialog. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3687–3703, Seattle, Washington, USA.

[117] Srinivasan Janarthanam, Helen Hastie, Oliver Lemon, and Xingkun Liu. 2011. "The day after the day after tomorrow?" A machine learning approach to

adaptive temporal expression generation: training and evaluation with real users. In *Proceedings of the SIGDIAL 2011 Conference*, pages 142–151, Portland, Oregon.

[118] Minwoo Jeong and Gary Geunbae Lee. 2008. Triangular-chain conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1287–1302.

[119] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *arXiv preprint arXiv:2202.03629*.

[120] Yimin Jing, Deyi Xiong, and Zhen Yan. 2019. BiPaR: A Bilingual Parallel Dataset for Multilingual and Cross-lingual Reading Comprehension on Novels. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP '19, pages 2452–2462, Hong Kong, China.

[121] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

[122] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

[123] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1601–1611, Vancouver, Canada.

[124] Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. 2017. Cross-language Learning with Adversarial Neural Networks. In *Proc. of the Conference on Computational Natural Language Learning*, CoNLL '17, pages 226–237, Vancouver, Canada.

[125] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online.

[126] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question Answering via Integer Programming over

Semi-Structured Knowledge. In *Proceedings of the Twenty-fifth International Joint Conferences on Artificial Intelligence Organization*, IJCAI '16, pages 1145–1152, New York, New York.

[127] Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '18, pages 1905–1914, New Orleans, Louisiana, USA.

[128] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online.

[129] Tushar Khot, Peter Clark, Michal Guerquin, Paul Edward Jansen, and Ashish Sabharwal. 2020. QASC: A Dataset for Question Answering via Sentence Composition. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, AAAI '20, pages 8082–8090, New York, New York, USA.

[130] Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering Complex Questions Using Open Information Extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 311–316, Vancouver, Canada.

[131] Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A Textual Entailment Dataset from Science Question Answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '18, pages 5189–5197, New Orleans, Louisiana, USA.

[132] Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76(9):11377–11390.

[133] Kyungduk Kim, Cheongjae Lee, Sangkeun Jung, and Gary Geunbae Lee. 2008. A Frame-Based Probabilistic Framework for Spoken Dialog Management Using Dialog Examples. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 120–127, Columbus, Ohio.

[134] Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.

[135] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 2015 International Conference on Learning Representations*, ICLR '15, San Diego, California.

[136] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

[137] Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A Large-Scale Corpus for Conversation Disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy.

[138] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

[139] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College*, pages 282–289, Williamstown, MA, USA.

[140] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, EMNLP '17, pages 785–794, Copenhagen, Denmark.

[141] Guillaume Lample and François Charton. 2020. Deep Learning For Symbolic Mathematics. In *Proceedings of the 8th International Conference on Learning Representations*, ICLR '20, Online.

[142] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the 8th International Conference on Learning Representations*, ICLR '20, Online.

[143] Staffan Larsson and David R. Traum. 2000. Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. *Nat. Lang. Eng.*, 6(3–4):323–340.

[144] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

[145] Sungjin Lee. 2017. Toward continual learning for conversational agents. *arXiv preprint arXiv:1712.09943*.

[146] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia.

[147] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

[148] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 7315–7330, Online.

[149] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

[150] Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

[151] Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Dialogue Learning With Human-in-the-Loop. In *International Conference on Learning Representations*.

[152] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-End Task-Completion Neural Dialogue Systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 733–743, Taipei, Taiwan.

[153] Xiujun Li, Lihong Li, Jianfeng Gao, Xiaodong He, Jianshu Chen, Li Deng, and Ji He. 2015. Recurrent reinforcement learning: a hybrid approach. In *Proceedings of the 2016 International Conference on Learning Representations*, ICLR '16, San Juan, Puerto Rico.

[154] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A Survey on Truth Discovery. *SIGKDD Explor. Newsl.*, 17(2):1–16.

[155] Yang Li, Qingliang Miao, Ji Geng, Christoph Alt, Robert Schwarzenberg, Leonhard Hennig, Changjian Hu, and Feiyu Xu. 2018. Question Answering for Technical Customer Support. In *Natural Language Processing and Chinese Computing*, pages 3–15.

[156] Yangming Li and Kaisheng Yao. 2021. Interpretable NLG for Task-oriented Dialogue Systems with Heterogeneous Rendering Machines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13306–13314.

[157] Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. KorQuAD1.0: Korean QA Dataset for Machine Reading Comprehension. *ArXiv*, abs/1909.07005.

[158] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the ACL Workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.

[159] Chin-Yew Lin and Franz Josef Och. 2004. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics*, ACL '04, pages 605–612, Barcelona, Spain.

[160] Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC '16, pages 23–28, Portorož, Slovenia.

[161] Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, INTERSPEECH '16, pages 685–689, San Francisco, USA.

[162] Bing Liu and Sahisnu Mazumder. 2021. Lifelong and Continual Learning Dialogue Systems: Learning during Conversation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):15058–15063.

[163] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 2122–2132, Austin, Texas, USA.

[164] Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019. XQA: A Cross-lingual Open-domain Question Answering Dataset. In *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, ACL '19, pages 2358–2368, Florence, Italy.

[165] Pengyuan Liu, Yuning Deng, Chenghao Zhu, and Han Hu. 2019. XCMRC: Evaluating Cross-lingual Machine Reading Comprehension. In *Proceedings of the International Conference on Natural Language Processing and Chinese Computing*, NLPCC '19, pages 552–564, Dunhuang, China.

[166] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

[167] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[168] Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

[169] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, TeachingNLP '02, pages 63–70, Philadelphia, Pennsylvania, USA.

[170] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, New Orleans, Louisiana, USA.

[171] Samuel Louvan and Bernardo Magnini. 2020. Recent Neural Methods on Slot Filling and Intent Classification for Task-Oriented Dialogue Systems: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online).

[172] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1116–1126, Vancouver, Canada.

[173] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems . In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '15, pages 285–294, Prague, Czech Republic.

[174] Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2021. Multi-stage Training with Improved Negative Contrast for Neural Passage Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6091–6103, Online and Punta Cana, Dominican Republic.

[175] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '15, pages 1412–1421, Lisbon, Portugal.

[176] Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual Learning in Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467, Online and Punta Cana, Dominican Republic.

[177] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia.

[178] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, ACL '14, pages 55–60, Baltimore, Maryland.

[179] Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A Survey on Computational Propaganda Detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4826–4832. Survey track.

[180] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 591–598, Stanford, CA, USA.

[181] Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual Learning for Natural Language Generation in Task-oriented Dialog Systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3461–3474, Online.

[182] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question

Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 2381–2391, Brussels, Belgium.

[183] Todor Mihaylov and Anette Frank. 2019. Discourse-Aware Semantic Self-Attention for Narrative Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP '19, pages 2541–2552, Hong Kong, China.

[184] Simona Mihaylova, Iva Borisova, Dzhovani Chemishanov, Preslav Hadzhitsanev, Momchil Hardalov, and Preslav Nakov. 2021. DIPS at CheckThat! 2021: Verified Claim Retrieval. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 558–571.

[185] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119.

[186] Arindam Mitra, Peter Clark, Oyvind Tafjord, and Chitta Baral. 2019. Declarative Question Answering over Knowledge Bases containing Natural Language Text with Answer Set Programming. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI '19, pages 3003–3010, Honolulu, Hawaii, USA.

[187] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada.

[188] Preslav Nakov. 2003. Building an Inflectional Stemmer for Bulgarian. In *Proceedings of the 4th International Conference Conference on Computer Systems and Technologies: E-Learning*, CompSysTech '03, pages 419–424, Rousse, Bulgaria.

[189] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, pages 27–48, Vancouver, Canada.

[190] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*, CoCo@NIPS '16, Barcelona, Spain.

[191] Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. 2019. Learning to Attend On Essential Terms: An Enhanced Retriever-Reader Model for Open-domain Question Answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 335–344, Minneapolis, Minnesota, USA.

[192] Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and Response Selection for Multi-Party Conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2133–2143, Austin, Texas, USA.

[193] Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. 2021. Probing Toxic Content in Large Pre-Trained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4262–4274, Online.

[194] Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving Question Answering with External Knowledge. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, MRQA '19, pages 27–37, Hong Kong, China.

[195] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 257–266, Singapore, Singapore.

[196] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania.

[197] Prasanna Parthasarathi, Mohamed Abdelsalam, Sarath Chandar, and Joelle Pineau. 2021. A Brief Study on the Effects of Training Generative Dialogue Models with a Semantic loss. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 469–476, Singapore and Online.

[198] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, NeurIPS '19, pages 8024–8035.

[199] Anselmo Peñas, Eduard Hovy, Pamela Forner, Álvaro Rodrigo, Richard Sutcliffe, Corina Forascu, Yassine Benajiba, and Petya Osenova. 2012. Overview of QA4MRE at CLEF 2012: Question Answering for Machine Reading Evaluation. In *CLEF Working Note Papers*, pages 1–24, Rome, Italy.

[200] Anselmo Peñas, Christina Unger, and Axel-Cyrille Ngonga Ngomo. 2014. Overview of CLEF Question Answering Track 2014. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 300–306.

[201] Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192, Melbourne, Australia.

[202] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1532–1543, Doha, Qatar.

[203] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, NAACL-HLT '18, pages 2227–2237, New Orleans, Louisiana.

[204] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP '19, pages 2463–2473, Hong Kong, China.

[205] Dean Pomerleau and Delip Rao. 2017. Fake News Challenge Stage 1 (FNC-I): Stance Detection. https://www.fakenewschallenge.org/.

[206] Popat, Kashyap and Mukherjee, Subhabrata and Strötgen, Jannik and Weikum, Gerhard. 2016. Credibility Assessment of Textual Claims on the Web. In *CIKM*.

[207] Martin F Porter. 1980. An algorithm for suffix stripping. *Program*.

[208] Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to Pre-Train on? Efficient Intermediate Task Selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic.

[209] Albert Pritzkau. 2021. NLytics at CheckThat! 2021: Multi-class fake news detection of news articles and domain identification with RoBERTa - a baseline model. In *CLEF (Working Notes)*, pages 572–581.

[210] Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China.

[211] Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021. A Co-Interactive Transformer for Joint Slot Filling and Intent Detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197, Toronto, ON, Canada.

[212] Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816, Online.

[213] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 498–503, Vancouver, Canada.

[214] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/languageunderstandingpaper.pdf.

[215] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog.

[216] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

[217] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the Meeting of the Association for Computational Linguistics*, ACL '18, pages 784–789, Melbourne, Australia.

[218] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 2383–2392, Austin, Texas, USA.

[219] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

[220] Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi, and K. Chandrasekaran. 2017. A Survey of Design Techniques for Conversational Agents. In *Information, Communication and Computing Technology*, pages 336–350, Singapore.

[221] Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and LSTM models for lexical utterance classification. In *Sixteenth Annual Conference of the International Speech Communication Association*, INTERSPEECH '15, pages 135–139, Dresden, Germany.

[222] Evgeniia Razumovskaia, Goran Glavaš, Olga Majewska, Edoardo Ponti, and Ivan Vulić. 2022. Natural Language Processing for Multilingual Task-Oriented Dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 44–50, Dublin, Ireland.

[223] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

[224] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China.

[225] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 193–203, Seattle, Washington, USA.

[226] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, Edinburgh, Scotland, UK.

[227] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

[228] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

[229] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for Building an Open-Domain Chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online.

[230] Oscar J. Romero, Antian Wang, John Zimmerman, Aaron Steinfeld, and Anthony Tomasic. 2021. A Task-Oriented Dialogue Architecture via Transformer Neural Language Models and Symbolic Injection. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 438–444, Singapore and Online.

[231] Vasile Rus and Mihai Lintean. 2012. A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162, Montreal, Canada.

[232] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 3856–3866, Long Beach, CA, USA.

[233] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations*.

[234] Shiki Sato, Reina Akama, Hiroki Ouchi, Jun Suzuki, and Kentaro Inui. 2020. Evaluating Dialogue Generation Systems via Response Selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 593–599, Online.

[235] Jacques Savoy. 2007. Searching strategies for the Bulgarian language. *Inform. Retrieval*, 10(6):509–529.

[236] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, New Orleans, Louisiana, USA.

[237] Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. PEER: A Collaborative Language Model. *arXiv preprint arXiv:2208.11663*.

[238] Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online.

[239] Timo Schick and Hinrich Schütze. 2021. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online.

[240] Carissa Schoenick, Peter Clark, Oyvind Tafjord, Peter D. Turney, and Oren Etzioni. 2017. Moving beyond the Turing Test with the Allen AI Science Challenge. *Communications of the ACM*, 60:60–64.

[241] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 1715–1725.

[242] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the 2017 International Conference on Learning Representations*, ICLR '17, Toulon, France.

[243] Iulian V. Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A Survey of Available Corpora For Building Data-Driven Dialogue Systems: The Journal Version. *Dialogue & Discourse*, 9(1):1–49.

[244] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 3776–3783.

[245] Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a Known Lie: Detecting Previously Fact-Checked Claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online.

[246] Shaden Shaar, Fatima Haouari, Watheq Mansour, Maram Hasanain, Nikolay Babulkov, Firoj Alam, Giovanni Da San Martino, Tamer Elsayed, and Preslav Nakov. 2021. Overview of the CLEF-2021 CheckThat! Lab Task 2 on Detecting Previously Fact-Checked Claims in Tweets and Political Debates. In *CLEF (Working Notes)*, pages 393–405.

[247] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP'15, pages 1577–1586, Beijing, China.

[248] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

[249] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

[250] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. BlenderBot 3: A deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

[251] A.B. Siddique, Fuad Jamour, and Vagelis Hristidis. 2021. Linguistically-Enriched and Context-AwareZero-Shot Slot Filling. In *Proceedings of the Web Conference 2021*, WWW '21, page 3279–3290.

[252] Kiril Ivanov Simov, Petya Osenova, Georgi Georgiev, Valentin Zhikov, and Laura Tolosi. 2012. Bulgarian Question Answering for Machine Reading. In *CLEF Working Note Papers*, Rome, Italy.

[253] Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can You Put it All Together: Evaluating Conversational Agents' Ability to Blend Skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online.

[254] Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, Chandana Satya Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gokhan Tur, and Prem Natarajan. 2022. AlexaTM 20B: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv*.

[255] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. Learning from Noisy Labels with Deep Neural Networks: A Survey. *ArXiv*, abs/2007.08199.

[256] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

[257] Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149*.

[258] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 553–562, Melbourne, Australia.

[259] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado.

[260] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses . In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '15, pages 196–205, Denver, Colorado.

[261] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[262] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training Very Deep Networks. In *Advances in Neural Information Processing Systems*, volume 28.

[263] Peter Stefanov, Kareem Darwish, Atanas Atanasov, and Preslav Nakov. 2020. Predicting the Topical Stance and Political Leaning of Media using Tweets. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 527–537, Online.

[264] Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative Deep Dyna-Q: Robust Planning for Dialogue Policy

Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3813–3823, Brussels, Belgium.

[265] Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland.

[266] Sethuramalingam Subramaniam, Pooja Aggarwal, Gargi B. Dasgupta, and Amit Paradkar. 2018. COBOTS - A Cognitive Multi-Bot Conversational Framework for Technical Support. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 597–604, Richland, SC.

[267] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems*, volume 28.

[268] Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. BORT: Back and Denoising Reconstruction for End-to-End Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2156–2170, Seattle, Washington, USA.

[269] Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A Challenge Data Set and Models for Dialogue-Based Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

[270] Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving Machine Reading Comprehension with General Reading Strategies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 2633–2643, Minneapolis, Minnesota, USA.

[271] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

[272] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, NIPS '14, pages 3104–3112, Montreal, Canada.

[273] Oyvind Tafjord, Peter Clark, Matt Gardner, Wen tau Yih, and Ashish Sabharwal. 2019. QuaRel: A Dataset and Models for Answering Questions about

Qualitative Relationships. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI '19, pages 7064–7071, Honolulu, Hawaii, USA.

[274] Chongyang Tao, Jiazhan Feng, Rui Yan, Wei Wu, and Daxin Jiang. 2021. A Survey on Response Selection for Retrieval-based Dialogues. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4619–4626. Survey Track.

[275] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1–11, Florence, Italy.

[276] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-range Reasoning for Machine Comprehension. *arXiv preprint arXiv:1803.09074*.

[277] Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A knowledge graph of fact-checked claims. In *International Semantic Web Conference*, pages 309–324. Springer.

[278] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online.

[279] Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

[280] James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA.

[281] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana.

[282] Kristina Toutanvoa and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China.

[283] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, RepL4NLP '19, pages 191–200, Vancouver, Canada.

[284] Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and Practical BERT Models for Sequence Labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3632–3636, Hong Kong, China.

[285] Gilad Tsur, Yuval Pinter, Idan Szpektor, and David Carmel. 2016. Identifying Web Queries with Question Intent. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 783–793.

[286] Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in ATIS? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24, Berkeley, California, USA. IEEE.

[287] Kiet Van Nguyena, Khiem Vinh Trana, Son T Luua, and Anh Gia-Tuan. 2020. Enhancing lexical-based approach with external knowledge for Vietnamese multiple-choice reading comprehension. *ArXiv*, abs/2001.05687.

[288] Lindsey Vanderlyn, Gianna Weber, Michael Neumann, Dirk Väth, Sarina Meyer, and Ngoc Thang Vu. 2021. "It seemed like an annoying woman": On the Perception and Ethical Considerations of Affective Language in Text-Based Conversational Agents. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 44–57, Online.

[289] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, NIPS '17, pages 5998–6008, Long Beach, CA, USA.

[290] Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open Intent Extraction from Natural Language Interactions. In *Proceedings of The Web Conference 2020*, WWW '20, page 2009–2020.

[291] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, Canada.

[292] James Vincent. 2016. Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day. *The Verge*, 24.

[293] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems*, volume 28.

[294] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR*, abs/1506.05869.

[295] Nguyen Vo and Kyumin Lee. 2019. Learning from Fact-Checkers: Analysis and Generation of Fact-Checking Language. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 335–344.

[296] Nguyen Vo and Kyumin Lee. 2020. Where Are the Facts? Searching for Fact-checked Information to Alleviate the Spread of Fake News. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731, Online.

[297] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online.

[298] Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019. Learning with Noisy Labels for Sentence-level Sentiment Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6286–6292, Hong Kong, China.

[299] Weishi Wang, Steven C.H. Hoi, and Shafiq Joty. 2020. Response Selection for Multi-Party Conversations with Dynamic Topic Tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6581–6591, Online.

[300] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788.

[301] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada.

[302] Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, W-NUT '17, pages 94–106, Copenhagen, Denmark.

[303] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

[304] Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A Survey of Joint Intent Detection and Slot Filling Models in Natural Language Understanding. *ACM Comput. Surv.* Just Accepted.

[305] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain.

[306] Jason Williams. 2007. Applying POMDPs to Dialog Systems in the Troubleshooting Domain. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, pages 1–8, Rochester, NY.

[307] Jason Williams. 2008. Demonstration of a POMDP Voice Dialer. In *Proceedings of the ACL-08: HLT Demo Session*, pages 1–4, Columbus, Ohio.

[308] Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

[309] Jason D Williams and Geoffrey Zweig. 2016. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.

[310] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, EMNLP '20, pages 38–45, Online.

[311] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 808–819, Florence, Italy.

[312] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270.

[313] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[314] Yuexin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2019. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7289–7296.

[315] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot User Intent Detection via Capsule Neural Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099, Brussels, Belgium.

[316] Dongfang Xu, Peter Jansen, Jaycie Martin, Zhengnan Xie, Vikas Yadav, Harish Tayyar Madabushi, Oyvind Tafjord, and Peter Clark. 2020. Multi-class Hierarchical Question Classification for Multiple Choice Science Exams. In *Proceedings of the 12th Language Resources and Evaluation Conference*, LREC '20, pages 5370–5382, Marseille, France.

[317] Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. Learning New Skills after Deployment: Improving open-domain internet-driven dialogue with human feedback. *arXiv preprint arXiv:2208.03270*.

[318] Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019. End-to-End Knowledge-Routed Relational Dialogue System for Automatic Diagnosis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7346–7353.

[319] Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83, Olomouc, Czech Republic. IEEE.

[320] Puyang Xu and Ruhi Sarikaya. 2013. Exploiting shared information for multi-intent natural language sentence classification. In *Fourteenth Annual Conference of the International Speech Communication Association*, pages 3785–3789, Lyon, France.

[321] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online.

[322] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building Task-Oriented Dialogue Systems for Online Shopping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

[323] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *Proceedings of the Conference of the North American Chapter of ACL*, NAACL-HLT '19, pages 72–77, Minneapolis, Minnesota, USA.

[324] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, NIPS '19, pages 5754–5764, Vancouver, Canada.

[325] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 2369–2380, Brussels, Belgium.

[326] Qi Ye, Feng Wang, and Bo Li. 2016. StarrySky: A Practical System to Track Millions of High-Precision Query Intents. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 961–966.

[327] Di You, Nguyen Vo, Kyumin Lee, and Qiang Liu. 2019. Attributed multi-relational attention network for fact-checking url recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1471–1480.

[328] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

[329] Steve J. Young, Milica Gasic, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management. *Comput. Speech Lang.*, 24:150–174.

[330] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *Proceedings of the 2018 International Conference on Learning Representations*, ICLR '18, Vancouver, Canada.

[331] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages

111–120, Hong Kong, China.

[332] Zengfeng Zeng, Dan Ma, Haiqin Yang, Zhen Gou, and Jianping Shen. 2021. Automatic Intent-Slot Induction for Dialogue Systems. In *Proceedings of the Web Conference 2021*, WWW '21, page 2578–2589.

[333] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint Slot Filling and Intent Detection via Capsule Neural Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy.

[334] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

[335] Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-Oriented Dialog Systems That Consider Multiple Appropriate Responses under the Same Context. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9604–9611.

[336] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online.

[337] Zhen Zhang, Hao Huang, and Kai Wang. 2020. Using Deep Time Delay Neural Network for Slot Filling in Spoken Language Understanding. *Symmetry*, 12(6).

[338] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

[339] Tiancheng Zhao and Maxine Eskenazi. 2016. Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–10, Los Angeles.

[340] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-Grounded Dialogue Generation with Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 3377–3390, Online.

[341] Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '18, pages 1458–1467, Melbourne, Australia.

[342] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *Computational Linguistics*, 46(1):53–93.

[343] Wenxuan Zhou and Muhao Chen. 2021. Learning from Noisy Labels for Entity-Centric Information Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic.

[344] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127, Melbourne, Australia.

[345] Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah Smith. 2021. Challenges in Automated Debiasing for Toxic Language Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3143–3155, Online.

[346] Su Zhu, Zijian Zhao, Rao Ma, and Kai Yu. 2020. Prior Knowledge Driven Label Embedding for Slot Filling in Natural Language Understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1440–1451.

[347] Arkaitz Zubiaga. 2018. A longitudinal assessment of the persistence of Twitter datasets. *Journal of the Association for Information Science and Technology*, 69(8):974–984.

[348] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and Resolution of Rumours in Social Media: A Survey. *ACM Comput. Surv.*, 51(2).

[349] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PloS one*, 11(3):1–29.

[350] Victor Zue, Stephanie Seneff, James R Glass, Joseph Polifroni, Christine Pao, Timothy J Hazen, and Lee Hetherington. 2000. JUPlTER: a telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):85–96.

Appendix A

Curating Answers from External Knowledge Sources

A.1 Answer Retrieval from a Pool of Explanations

A.1.1 Hyperparameters and Fine-Tuning

Common Parameters [1]

- The models are developed in Python using PyTorch (198), the Transformers library (310) and the Sentence Transformers library (224).[2]
- I used NLTK (169) to filter out English stop words, *Twitter Tokenizer* to split the tweets and to strip the handles, and Porter's stemmer (207) to stem the tokens.
- For model optimization I use AdamW (170) with weight decay 1e-8, β_1 0.9, β_2 0.999, ϵ 1e-08, for 10 epochs and maximum sequence length of 128 tokens (per encoder).[3]
- All SentenceBERT models are initialized from the *'stsb-bert-base'*[4] checkpoint.
- The SBERT models use cosine similarity both during training inside the MNR loss and during inference for ranking.
- The values of the hyper-parameters were selected on the development set of CheckThat '21[5] and I chose the best model checkpoint based on the performance on the development set (MAP@5).
- I ran each experiment three times with different seeds and averaged all the metrics.

[1] The code and the data will be made available with the camera-ready version.
[2] github.com/UKPLab/sentence-transformers
[3] When needed, I truncated the sequences token by token, starting from the longest sequence in the pair.
[4] huggingface.co/sentence-transformers/stsb-bert-base
[5] https://gitlab.com/checkthat_lab/clef2021-checkthat-lab/-/tree/master/task2

- The models were evaluated on each epoch or 250 steps, whichever is less.
- The evaluation metrics are calculated using the official code from the CheckThat '21 competition (246)[6] and the SentenceTransformer's library.
- In this thesis, I list 199 examples for the development set of CheckThat '21, while Shaar et al. (246) lists 200. The difference comes from one duplicate row in the development set, which I found and filtered out.
- I trained my models on 5x Tesla T4 GPUs and 1x GeForce GTX 1080Ti, depending on the dataset size, the experiments took between 10 minutes and 5 hours.

Baseline SBERT

- Baseline SentenceBERT is trained w/ LR 2e-05, warmup 0.1, and batch size 32.
- I set the temperature (τ) in the MNR loss to 1.0, i.e., using unmodified MNR.
- The model consists 110M params, same as the bert-base Devlin et al. (56), as it uses a bi-encoder scheme.

Proposed Pipeline

- The model is trained w/ LR 1e-05, warmup 0.1, and batch size 8, group size of 4 during the dataset shuffling.
- I tuned settings of the self-adaptive training approach: momentum α to 0.9, refurbishment process starting at the second epoch.
- I set the learning rate for temperature (τ) in the MNR loss to 0.4.
- In the re-ranking, I used 800 training examples to train SBERT and the remaining 199 to train LambdaMART.
- I re-ranked the top-100 results from the best SentenceBERT model with LambdaMART.
- All other training details I kept from (34).
- The model consists 330M params, 3x as the size of the Baseline SBERT, as it trains three separate models.
- In my preliminary experiments, SBERT-base and SBERT-large models achieved the same results in terms of MAP@5, therefore I experiment with the *base* versions.

[6] https://gitlab.com/checkthat_lab/clef2021-checkthat-lab/-/tree/master/task2/scorer

A.1.2 Annotations

Setup and Guidelines Each annotator was provided by the same guidelines and briefed in from one of the authors of this paper. For annotation I used a Google Sheets document, where non of the annotators had access to the annotations from the others. The annotation sheet contained the following fields:

- *tweet_text* – the text of the fact-checking tweet
- *text_conversation* – the text of the root of the conversation
- *text_reply* – the text of the last tweet before the fact-checking one
- *title* – the title of the Snopes article
- *subtitle* – the subtitle of the Snopes article.

The task annotation task is to mark if 'Conversation matches' and 'Replay matches' using a check-boxes. I also allowed them to add comments as a free form text.

Demographics I recruited three annotators – 2 male and 1 female on age between 25 and 30. The annotators have higher education (at least a bachelors degree), and are currently enrolled in a Masters or Ph.D. programs in computer science. Each annotator is proficient in English but is not a native speaker of the language.

Inter-annotator Agreement Here, I present the inter-annotator agreement. I measure the overall agreement using Fleiss kappa (75) (shown in Figure A.1) but also the agreement between each two annotators using Cohen's Kappa (shown in Table A.2). The overall level of agreement of agreement between the annotators is *good*. Moreover, we can see that between annotator A and C the agreement is almost perfect both for the replies and conversations. The lowest agreement is between A and B but still substantial.

	Replay	Conversation
Fleiss Kappa	0.745	0.750

Table A.1: Fleiss Kappa inter-annotator agreement between my three annotators (A, B, C).

Annotators	Replay	Conversation
	Cohen Kappa	
A $\leftrightarrow$ B	0.650	0.655
A $\leftrightarrow$ C	0.885	0.922
B $\leftrightarrow$ C	0.698	0.673

Table A.2: Cohen Kappa inter-annotator agreement between the three annotators (A, B, C).

Disagreement Analysis After the annotations procedure was finished I analyzed the examples the annotators disagree on:

(i) The first type of claims that cause disagreement are the ones depend on information external sources, e.g., *'Blame Russia again? [URL]'*.

(ii) The second type are tweets containing multiple claims that needs to be fact-checked, however the referenced article does not target the main claim, e.g., *'It sounds like someone who is scared as heck that they will not win," Shermichael Singleton says of Pres. Trump's remarks encouraging his supporters to vote twice.'* and its crowd fact-check *'Did Trump Tell People To Vote Twice?'*. Here, the main claim is in the quote itself, while the remark about voting twice is secondary.

(iii) Third type are – the claim is ambiguous *Fanta (soft drink) was created so that the Nazi's could replace Coca-Cola during WWII [URL]*, and the fact-check is about *'Was Fanta invented by the Nazis?'*. Here, it is not clear who created Fanta. The final pattern is – the claim is partial match with the fact-check, e.g., *'did President Trump have a great economy and job creation for 1st 3 years???'*, and the fact-check is *'Did Obama's Last 3 Years See More Jobs Created Than Trump's First 3?'*

.

Tweet-Article Pairs Analysis In Table A.3, I show examples of *correct* (✓) and *incorrect* (✗) matching pairs. I sorted the examples within each group based on the word overlap between the claim and the verified claim, e.g., (1) and (2) have more words in common between the two texts compared to the overlaps in (3), and similarly for (4)–(6).

First, I can see that high overlap does not guarantee a correct pair, just like low overlap does not mean an incorrect pair, which is also visible from the analysis of the Jaccard similarity in Table 4.9. These two phenomena can be seen in (*3*), i.e., a correct pair with low overlap, and in (*4*), i.e., an incorrect match with high overlap. Next, some tweets may not contain a claim such as (*4*), as the user only asks questions, rather than stating something that can be fact-checked. In contrast, (*6*) contains a verifiable claim about *gas prices*, but the linked Snopes article fact-checks whether *COVID spreads through gas pumps*, which is irrelevant in this case. Row (*5*) is a partial match, and the tweet contains a check-worthy claim, but the article by the crowd fact-checker focuses on the IQ of the Fox News viewers, rather than on how well informed they are, and thus again the match is incorrect. Finally, in row (*1*), we can see that the verified claim is almost exactly included in the tweet, which is an easy case to match. In contrast, for the example in row (*3*), the model should do a semantic match based on some prior knowledge that the other name for *influenza A virus subtype H1N1* is *swine flu*, and moreover, *10,000* should be associated with the word *thousands*.

	Tweet w/ Claim	Snopes Verified Claim and Article
	Correct Matches ✓	
(1)	"Mussolini may have done many brutal and tyrannical things; he may have destroyed human freedom in Italy; he may have murdered and tortured citizens whose only crime was to oppose Mussolini; but 'one had to admit' one thing about the Dictator: he 'made the trains run on time.'" [URL]	Italian dictator Benito Mussolini made the trains run on time snopes.com/fact-check/loco-motive/
(2)	"Full list of songs Clear Channel banned following the 911 attacks. Some of these don't make any sense at all. 12 [URL]"	Clear Channel Communications banned their American radio stations from playing specified songs in order to avoid offending listeners. snopes.com/fact-check/radio-radio/
(3)	@user @user OMG! Were you on this planet when Obama did nothing during H1N1 crisis? Only difference was H1N1 caused more than 10000 deaths and Obama was golfing. Took 6 mos for him to even have a press conference!	U.S. President Barack Obama waited until millions were infected and thousands were dead before declaring a public health emergency concerning swine flu. snopes.com/fact-check/obama-wait-swine-flu-n1h1/
	Incorrect Matches ✗	
(4)	Dick Van Dyke? What's next? Penis Van Lesbian? What. Is. NEXT???	Dick Van Dyke's real name is Penis Van Lesbian. snopes.com/fact-check/dick-van-dyke/
(5)	"I've just found a 2012 report on how well informed TV viewers are NPR was top, of course. That's the one the Republicans want to defund, as it's contrary to their interests Also Fox viewers were less well informed than people who did not watch TV news at all"	A four-year study has found that Fox News viewers have IQs 20 points lower than average. snopes.com/fact-check/news-of-the-weak/
(6)	Trump just said he has seen gas prices at $.89-$.99 per gallon. Where I am it is currently $1.70. Anyone see prices Trump is talking about?	The COVID-19 coronavirus disease is "spreading quickly from gas pumps." snopes.com/fact-check/covid19-gas-pump-handles/

Table A.3: Examples from CrowdChecked, showing correct (✓) and incorrect matches (✗). The examples in each group are sorted by their overlap with the claim made in the tweet.

Appendix B

Advanced Conversation

B.1 Multi- and Cross-Linguality

B.1.1 Hyperparameters and Fine-Tuning

In this thesis, I am interested in the cross-lingual transferability of multilingual models such as mBERT (56) and XLM-RoBERTa (43), each of which comes pre-trained on more than 100 languages. I evaluated the QA capabilities of these models, following the established protocol (56; 167; 270), namely I fine-tuned them to predict the correct answer in a multi-choice setting, given a selected context. The aforementioned setup feeds the pre-trained model with a text, processed using the model's tokenizer in the following format:

[CLS] C [SEP] Q + O [SEP]

where C, Q and O are the tokenized *knowledge **C**ontext* (see Section 5.5.2), ***Q**uestion*, and ***O**ption*, respectively.

I used the Transformers library (310). I fine-tuned mBERT, XLM-R, and XLM-R_{Base} in three steps. 23 first fine-tuned the models with RACE (140), a multiple-choice reading comprehension dataset with around 85k questions for training. Then, I trained on the combination of ARC (39), OpenBookQA (182), and Regents Living Environments, as in the *AristoRoBERTaV7* ARC Challenge leaderboard entry[1]; I refer to these datasets as *SciENs* (**Sci**ence **En**glish datasets). I used the resulting pre-trained models as base models for my *Multilingual* and *Cross-lingual* evaluations (Section 5.5.4 in the paper). For the multilingual evaluation, I continued training the model, previously fine-tuned on the SciENs datasets, with my multilingual $Train_{Mul}$ set, validating on Dev_{Mul} and testing on $Test_{Mul}$. For my cross-lingual evaluation, I continued training the SciENs model on separate languages, as described in Section 5.5.4.

In Table B.1, I show the values of the hyper-parameters for each fine-tuning step and corresponding model. Note that these hyper-parameters were **not** obtained with an exhaustive search, and thus a better setting might exist for each

[1] https://leaderboard.allenai.org/arc/submission/blcotvl7rrltlue6bsv0

Model	Batch Size	Accum. Steps	Max Seq. Len.	Learn Rate	Warmup	Weight Decay
fine-tune on RACE (Step 1)						
mBERT	4	64	320	0.00005	0.1	-
XLM-R XLM-R_{Base}	2	16	320	0.00001	0.1	0.06
fine-tune on SciENs (Step 2)						
mBERT XLM-R XLM-R_{Base}	2	16	320	0.00001	0.2	0.06
$E\chi\alpha\mu s$ $Train_{Mul}$ (Step 3 - Multilingual)						
mBERT XLM-R XLM-R_{Base}	2	16	320	0.00001	0.2	0.06
for each source language (Step 3 - Cross-lingual)						
mBERT XLM-R XLM-R_{Base}	2	8	320	0.00001	0.2	0.06

Table B.1: The hyper-parameter values I used for fine-tuning.

model and dataset. Initially, I used the hyper-parameters for *AristoRoBERTaV7* ARC Leaderboard submission for English-only RoBERTa (167): epochs = 4, learning rate = 1e-5.

With these parameters alone, the models did not perform well, and thus I added a warmup of 0.1 and a weight decay of 0.06, which stabilized the training. In all experiments, I used the Adam optimizer with β_1=0.9, β_2=0.999, and ϵ=1e-08.

I further performed manual tuning of the hyper-parameters: I experimented with variations thereof, depending on the performance on the corresponding development sets, and I ended up with the values in Table B.1. Moreover, I adjusted the batch size and the accumulation steps depending on the availability of the GPUs on the cluster: Nvidia GTX 1080 Ti (Pascal, 11GB memory) or Nvidia Quadro RTX 6000 (24GB). For each examined setting, I trained for up-to 6 epochs, evaluating the model on the corresponding development set every 100 to 1000 update steps, depending on the dataset size and the effective batch size. For the final evaluations, I chose the model with the highest accuracy score on the corresponding development set.

Fine-tuning XLM-R (550M parameters) on Nvidia Quadro RTX 6000 (24GB) with the given hyper-parameters took around three hours per epoch when fine-tuned on RACE (∼85k examples), 30 minutes per epoch when fine-tuned on SciENs (∼9k examples), and 30 minutes on $E\chi\alpha\mu s$ on $Train_{Mul}$ (∼8k examples). Fine-tuning XLM-R_{Base} (270M parameters) and mBERT (172M parameters) on Nvidia GTX 1080 Ti (Pascal, 11GB memory) with the given hyper-parameters took roughly 2 to 2.5 hours per epoch when fine-tuned on RACE (∼85k examples), 30 to 35 minutes per

epoch when fine-tuned on SciENs (∼9k examples), and additional 30 minutes on the $E\chi\alpha\mu s$ Train_{Mul} (∼8k examples).

B.1.2 Subject Definitions

Next, I give a brief description of the less commonly known subjects included in $E\chi\alpha\mu s$.

Agriculture covers questions about soil farming and preservation, small animals breeding and their general health care, and vehicle maintenance and repair.

Business & Economics is a term used to combine five similar subjects related to business and economics. The questions in these subjects cover theoretical questions on economics basis, marketing questions, business questions with elements of accountancy, finances, and organizational studies.

Citizenship is a specific subject from the Vietnamese school system, which tries to inform and give better perspective on different social situations, to educate students in how to perform better, and to be a more aware member of the society by analyzing different norms and personal morality.

Fine Arts contains analytical and historical questions about different forms of art such as movies, music, art, etc.

Forestry studies the craft of managing, using, conserving, and repairing forests, woodlands, and associated resources around them such as water sources and soil.

Geology is the study of the Earth, with the general exclusion of present-day life, flow within the ocean, and the atmosphere. Questions from this subject cover branches of Geology such as Economical Geology, Marine Geology, Geomorphology, and Geophysics.

Informatics consists of questions about basic hardware knowledge and software management as well as basics of different positional numeral systems (e.g., binary and hexadecimal).

Islamic Studies refers to the academic studies of Islam, Quran excerpts, and Muslim morality. This a subject studied in the Qatari educational system during both middle and high school.

Landscaping teaches about modifying the visible features of an area of land, trees and park decorations. It also contains questions about plants and soils.

Politics covers Croatia's political system, historical questions about the country's development, as well as different regulations and laws, international relations and contracts.

Professional subject is present in the Polish school system and covers knowledge on specific professions such as flight attendant, babysitter, care taker, office worker in terms of profession's regulations, rules and established norms, etc.

Religion subject covers Christianity studies such as Bible knowledge, related traditions, e.g., baptism, marriage, etc.

Tourism covers hospitality management, as well as basis of business and traditions in Hungary and its neighboring countries.

Science which is used in the Arabic school system throughout middle and high grade studies combines general science questions from Biology, Chemistry, Physics Geology and their branches such as as Biophysics, Astrophysics, and Biochemistry.

Social subject, similarly to Science, combines questions from political, cultural, historical and geographical studies.

Sociology is the study of society, patterns of social relationships, social interaction, and culture that surrounds our everyday life.