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

SOFIA UNIVERSITY
St. KLIMENT OHRIDSKI
FACULTY OF
MATHEMATICS AND INFORMATICS

Intelligent Context-Aware Natural Language Dialogue Agent

by

Momchil Emilov Hardalov

Abstract

for a thesis submitted in fulfillment of the requirements
for the scientific degree of "Doctor"

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

The dissertation contains 179 pages, of which 8 pages are appendices. Includes 18 figures and 39 tables in 6 chapters, general conclusions, contributions and 2 appendices. The bibliography covers 350 titles, all of which in English. The list of the author's publications, directly reflecting the results of the dissertation, contains 6 titles. There are currently 77 known citations to these publications (found with Google Scholar).

Contents

Chapter 1. Introduction

1.1 Motivation and Relevance of the Topic

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

[1] https://www.drift.com/blog/state-of-conversational-marketing/
[2] ,https://cx-trends-report-2022.zendesk.com/growth-areas

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]

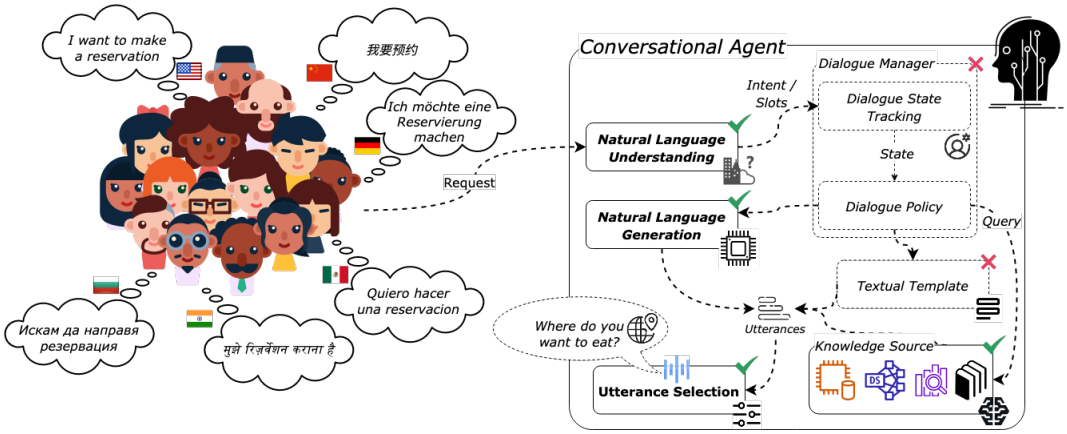


Figure 1.1: Conceptual diagram illustrating the information flow pipeline of a **task-oriented conversational agent**. The components I explore in this thesis are marked with ✓, and the ones that are *not* – with ✗.

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*. 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

[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/

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.

1.2 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.

1.3 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

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

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.

1.4 Published Papers

- 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

- 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 a wide range of holistic approaches to conversational agents, including datasets used for training and I provide sufficient background for the rest of the thesis.

First, 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 QA, covering full resource and zero-shot approaches applied in mono- and multilingual scenario. Further, I cover the problem of curating answers from external knowledge sources. I study this problem through the lenses of *finding previously fact-checked claims*, and thus 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, 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.

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 (Transformer-NLU). 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. 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.

This chapter is mainly based on Hardalov et al. (2020a).

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*.

3.1 Dataset

In my experiments, I use two publicly available datasets (see Table 3.2), 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.

3.2 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, (*ii*) I obtain predictions for the word case and named entities for each token (word features),

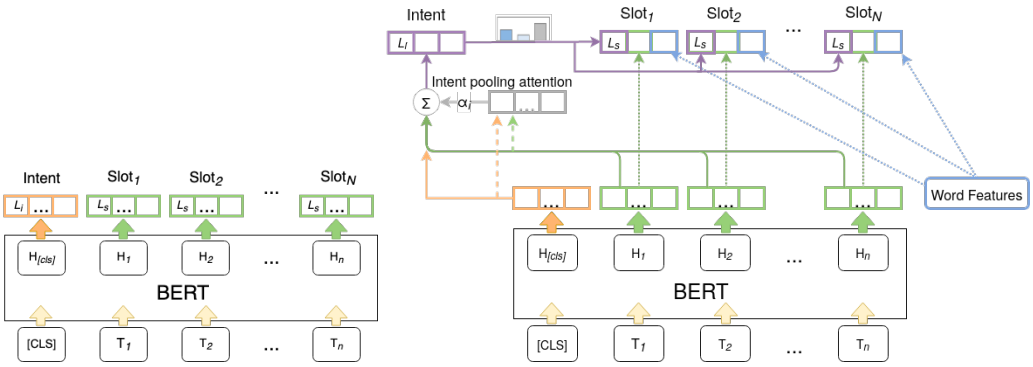
	ATIS	SNIPS
Vocab Size	722	11,241
Average Sentence Length	11.28	9.05
#Intents	21	7
#Slots	120	72
#Training Samples	4,478	13,084
#Dev Samples	500	700
#Test Samples	893	700

Table 3.2: Statistics about the ATIS and SNIPS datasets.

and (*iii*) I feed the predicted intent distribution vector, BERT's last hidden representations, and word features into a slot filling layer. The complete architecture of the model is shown in Figure 3.1b.

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. 3.1). 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.

$$\mathcal{L}_{joint} = \gamma * \mathcal{L}_{intent} + (1 - \gamma) * \mathcal{L}_{slot} \tag{3.1}$$



(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.

3.3 Experiments and Analysis

3.3.1 Evaluation Results

Table 3.3 presents quantitative evaluation results in terms of (*i*) intent accuracy, (*ii*) sentence accuracy, and (*iii*) slot F1. 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.

Model	ATIS			SNIPS		
	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.

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 $model_b$.

$$RER = 1 - \frac{Error_{model_a}}{Error_{model_b}} \tag{3.2}$$

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 in the dissertation). 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. 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.

3.3.2 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

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

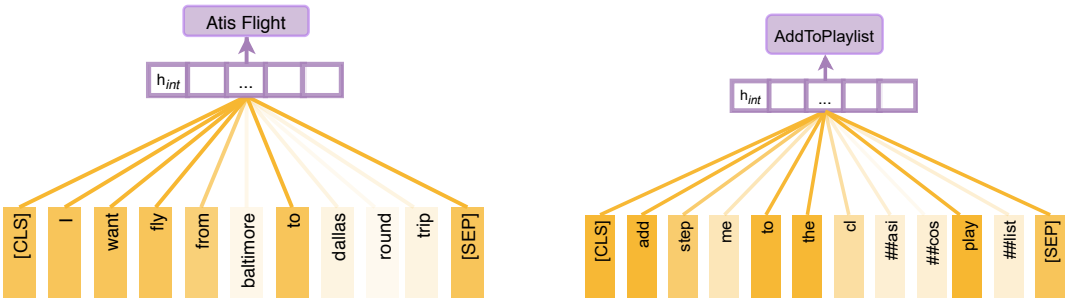
Table 3.4: Comparing *Transformer-NLU:BERT* to the two baselines: *(i)* current SOTA for each measure, and *(ii)* conventionally fine-tuned BERT-Joint without the improvements, in terms of relative error reduction (Eq. 3.2).

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.

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.

Finally, I visualize the learned attention weights on Figure 3.2. It presents a request from the ATIS (Figure 3.2a) and SNIPS (Figure 3.2b) datasets.



(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.

3.4 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 the first part of the chapter, 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 the second part of the chapter, 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 Hardalov et al. (2019a) and Hardalov et al. (2022).

4.1 Knowledge Retrieval

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.

4.1.1 Model

The model has three components (see Figure 4.1): (*i*) a context retrieval module, which tries to find good explanatory passages for each question-answer pair, from a corpus of non-English documents, (*ii*) a multiple-choice reading comprehension

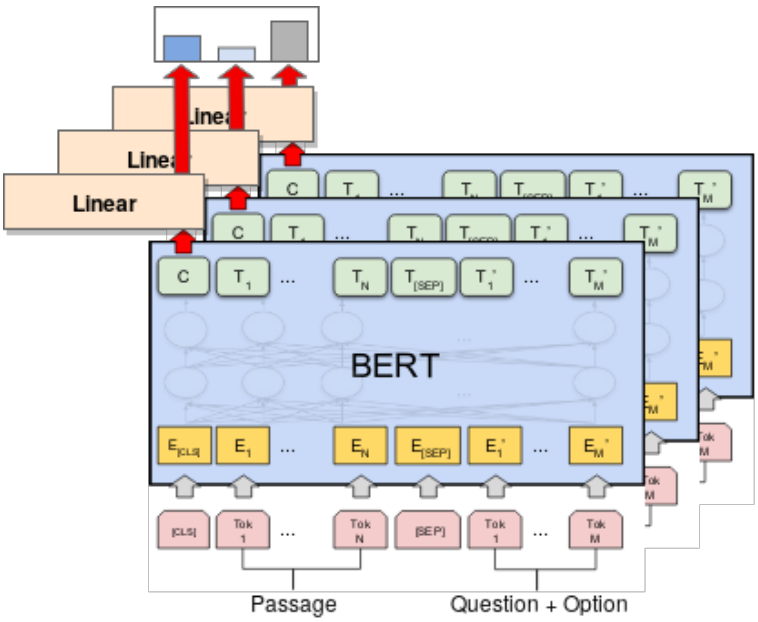


Figure 4.1: BERT for multiple-choice reasoning.

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, and (*iii*) a voting mechanism, which combines multiple passages from (*i*) and their scores from (*ii*) in order to obtain a single (most probable) answer for the target question.

In order to enable search for appropriate passages for non-English questions, I created an inverted index from Wikipedia articles using Elasticsearch. Following the notation of Devlin et al. (2019), the input sequence can be written as follows:

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

I normalize the scores after the projection layer by adding a softmax. During fine-tuning, I optimize the model's parameters by maximizing the log-probability of the correct answer.

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. Thus, instead of using only the first-hit document, we should also evaluate lower-ranked ones. In my experiments, I adopt a simple summing strategy. I evaluate each result from the context retriever against the question and the possible options, thus obtaining a list of raw probabilities.

4.1.2 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.

I collect my own dataset for Bulgarian, resulting in 2,633 multiple-choice questions, without contexts, from different subjects: biology (16.6%), philosophy (23.93%), geography (23.24%), and history (36.23%). Table 4.1 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 4.2 shows the distribution of questions per subject category, the length (in words) for both the questions and the options (candidate answers), and the

(**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

Table 4.1: Example questions, one per subject, from the Bulgarian dataset. The correct answer is marked in green.

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 4.2: Statistics about my Bulgarian dataset compared to the RACE dataset.

#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.

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.

Finally, I examine the vocabulary richness of the two datasets. The total number of unique words is shown in the last column of Table 4.2 (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.

4.1.3 Experiments and Evaluation

BERT Fine-Tuning

I divide the fine-tuning into two groups of models (*i*) Multilingual BERT, and (*ii*) Slavic BERT[1]. Table 4.3 below presents the results in the multiple-choice comprehension task on the dev dataset from RACE.

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

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.

Wikipedia Retrieval and Indexing

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: (*i*) *ngram*: word-based 1–3 grams; (*ii*) *bg*: lowercased, stop-words removed and stemmed; (*iii*) *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. Finally, I use a list of top-N hits for each candidate answer.

Experimental Results

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 fine-tune the models on the RACE dataset and 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 final column in the table, *Overall*, shows the accuracy calculated over all questions in the RACE testset. 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.

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 following rows show the results for experiments conducted with a model trained for three epochs on RACE.

From my experiments, 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.

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.

I further study the impact of the size of the results list returned by the retriever on the accuracy for the different categories. I further analyze the average accuracy for a given query size S_q over all performed experiments, where $S_q \in \{1, 2, 5, 10, 20\}$. My experiments show 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.

4.2 Answer Retrieval from a Pool of Explanations

In this section, 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 4.5). 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.

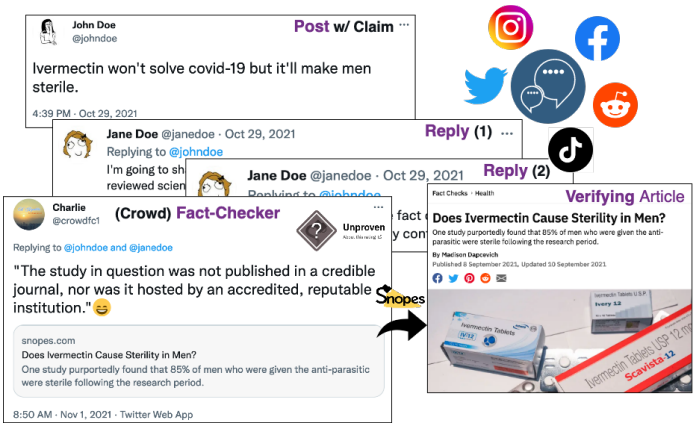


Figure 4.2: Crowd fact-checking thread on Twitter. The first tweet (**Post w/ claim**) makes the claim that *Ivermectin causes sterility in men*, which then receives **replies**. A **(crowd) fact-checker** replies with a link to a **verifying article** from a fact-checking website. I pair the *article* with the *tweet that made this claim* (the first post ✓), as it is irrelevant (✗) to the other *replies*.

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. Figure 4.2 shows an example.

4.2.1 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.[2] An example result from the query is shown in Figure 4.2, 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 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.[3]

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.

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

[3] 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.

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 4.5: 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.

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 4.6: Statistics about my 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.

The conversational threads are organized in a similar way shown Figure 4.2, 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).

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 paper. 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 4.6 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 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, and thus I use distant supervision to label them, with the resulting dataset sizes of matching pairs shown in Table 4.7. Here, I want to emphasize that there is absolutely no overlap between CrowdChecked and CheckThat '21 in terms of tweets/claims.

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, that received a fact-checking reply and the title/subtitle of the liked fact-checking (Snopes) article in that reply, and (*ii*) based on the predictions of a model trained on CheckThat '21.

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.

4.2.2 Method

General Scheme As a base for my models, I use Sentence-BERT (SBERT). 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. 4.1). Moreover, I propose a new variant of the MNR loss that accounts for the noise in the dataset.

Enriched Scheme Here, 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 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).

Training with Noisy Data

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).

The adapted version of the MNR loss is defined as follows:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m} {y^r}_i \Big(\frac{c_i^T v_i}{\tau} - \log \sum_{j=1}^{m} \exp(\frac{c_i^T v_j}{\tau}) \Big) \tag{4.1}$$

If I set $y_i^r = 1$, then Eq. 4.1 resembles the MNR loss definition. The parameter τ is the temperature.

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. After applying both modifications the impact of each training example is proportional to the square of the corrected label, i.e., in Eq. 4.1 y_i^r is now squared.

Re-ranking

I adopt the re-ranking 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 12 features in the ensemble and 4 in the single model.

4.2.3 Experiments

Datasets

Table 4.7 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.

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

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 4.8: 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**.

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. 4.1)	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 4.9: 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**.

Experimental Results

Threshold Selection Analysis Table 4.8 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.

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.

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. In Table 4.9, I ablate each of these modifications by adding them iteratively to the baseline SBERT model.

4.3 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 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 highest 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. First , 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.

Next, 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 the next section.

Finally, 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, *Knowledge Retrieval*.

This chapter is mainly based on Hardalov et al. (2018), Hardalov et al. (2019b) and Hardalov et al. (2020b).

5.1 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.

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.

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 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. Table 5.1 show statistics about the dataset.

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

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.

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

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.

5.2.1 Method

Preprocessing

Since Twitter has its own specifics of writing in terms of both length 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, slang words, URLs with <*url*>, all user mentions with <*user*>, and all hashtags with<*hashtag*>. I chose the top N words when building the model, and I replaced the instances of the remaining words with a special symbol <*unk*>.

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.

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

Table 5.2: Results based on word-overlap measures.

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.

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. 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.

5.2.2 Experiments

Table 5.2 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 5.3 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 to avoid adding biases, as it has been suggested in (Liu et al., 2016; Lowe et al., 2017).

	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 5.3: Results based on semantic measures.

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. 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.

5.3 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 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 for question–answer pairs.

5.3.1 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 5.1.

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

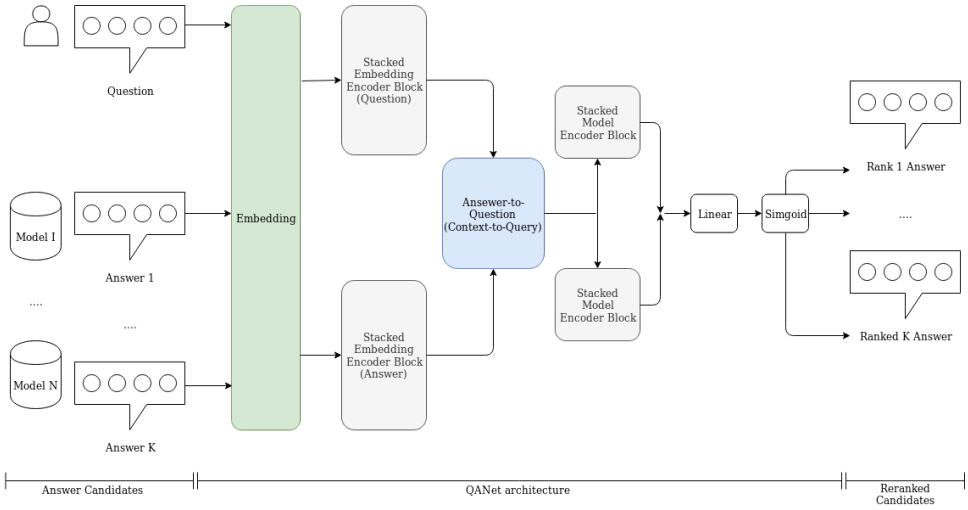


Figure 5.1: My answer re-ranking framework, based on the QANet architecture.

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.

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 embedding encoder layer is based on a convolution, followed by self-attention (Vaswani et al., 2017) and a feed-forward network. The output of the layer is $f(layernorm(x)) + x$, where *layernorm* is the layer normalization operation. 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.

The attention layer is followed by a model layer, which takes as input the concatenation of $[a; a2q; a \odot a2q; a \odot q2a]$, these are 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.

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 5.4: Auxiliary task: question–answer appropriateness classification results.

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 5.1. 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 empirically found that the answer selection based on the *max* strategy does not always perform well. 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).

5.3.2 Evaluation Results

Auxiliary Task: Question–Answer Appropriateness Classification

Table 5.4 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. The last column reports the accuracy. 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 ($d_{model} = 128$, $\#Heads = 8$).

Model	Word Overlap		Semantic Similarity		
	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 5.5: Main task: performance of the individual models. Single model results results are reported in Tables 5.2 and 5.3.

Answer Selection/Generation: Individual Models

Table 5.5 reports the performance of the individual models: information retrieval (IR), sequence-to-sequence (Seq2seq), and the Transformer. The same experimental setup is used for the experiments described in Section 5.2. 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, 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,

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

which helps avoid bias, as has been suggested in (Liu et al., 2016; Lowe et al., 2017).

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 5.5 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 5.6, 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.

5.4 Multi- and Cross-Linguality

Here, I present *Exams*, a new dataset and benchmark for multilingual and cross-lingual evaluation of models and methods for answering diverse school science questions (see Figure 5.2).

5.4.1 *Exams* Dataset

Dataset Statistics

I collected *Exams* 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

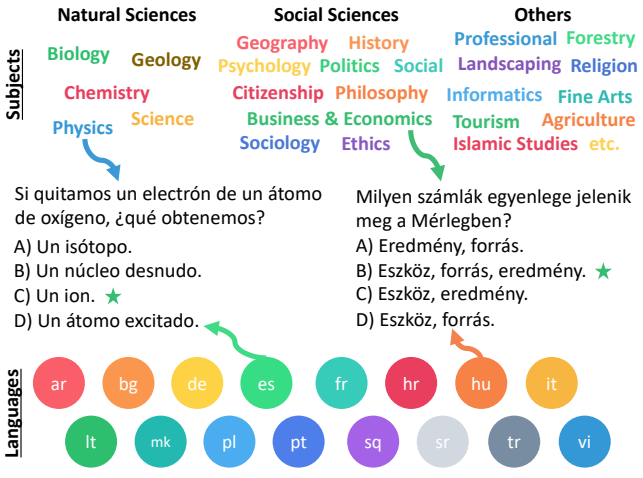


Figure 5.2: Properties and examples from *Exams*.

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.

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 5.7 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.

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 5.7: 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.

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 5.8. 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).

Data Splits

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.[2] 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.

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

	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 5.8: Parallel questions for different language pairs.

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

Since I have parallel data for several languages (discussed in Section 5.4.1), 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. The number of examples per language and the total number of multilingual sets are shown in the first three columns of Table 5.9.[3]

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}.[4] The last three columns of Table 5.9 show the number of examples used for training and validation with the corresponding language.

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

[4]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$_{Mul}$

	ARC		R12	*Exams*																
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 *Exams* (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 (*Exams* 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 5.10: 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 *Exams* questions.

5.4.2 Baseline Models

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.[5] 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.

Pre-trained Model as a Knowledge Base (KB) 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. 5.1).

$$score(O_i) = \frac{1}{|O_i|} \sum_{t \in O_i} \log P_{MLM}(t|Q) \tag{5.1}$$

Fine-Tuned Models

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

5.4.3 Experiments and Results

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_{Base}, and mBERT) compare. I follow a

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

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 5.11: 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.

standard training scheme for such tasks: first I fine-tune on RACE (Lai et al., 2017) (∼85k EN questions over documents), then on the AI2 English science datasets (I call them SciENs for shorter), including ∼9k EN questions with provided relevant contexts,[6] and, finally, on the multilingual training set (see Section 5.4.1) with retrieved relevant contexts from Wikipedia, 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 $E\chi\alpha\mu s$. Nevertheless, we see a 2.4% improvement with multilingual fine-tuning on $E\chi\alpha\mu s$ 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 $E\chi\alpha\mu s$, 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 $E\chi\alpha\mu s$ only. As expected, the performance drops by 3% absolute compared to the *Full* setup.

Knowledge Evaluation

The last two rows of Table 5.10 evaluate the knowledge in the best model, namely XLM-R. With *XLM-R as KB* (see Section 5.4.2) we see small improvement over the random baseline: +5% ARC Easy, 2% on R12, and just +1% on $E\chi\alpha\mu s$ 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.

Cross-lingual Evaluation

Table 5.11 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

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

language and I test on all other languages using the splits described in Section 5.4.1. 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.

5.5 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 appropriateness 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 *Exams*, 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 *Exams*. 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 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 Check-That '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.

6.2 Directions for Future Research

Modularized (task-oriented) conversational agents provide a great flexibility in terms of model training, and allow for easy adding new or to replace existing modules to the agent's pipeline. However, that flexibility brings several limitations along. First, there is a disconnect between different components (models) both during training and inference, that, in turn, leads to error accumulation along the pipeline. And second, including too many components can increase the computational cost, hence deploying the dialogue system can become infeasible. Here, I outline several promising directions for future research:

- In the short term, end-to-end differentiable architectures based on a combination of hierarchical neural networks, multi-task learning and multi-model error propagation can be a step forward in that direction.

- 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.
- Even with the development of models with capacity increased to billions of trainable parameters models are still vulnerable to both ethical and practical risks (Bender et al., 2021; Bommasani et al., 2021). That said, it is clear that we need more research and better models in order to release end-to-end models in a dynamic real-world scenarios. Some directions are:
  - Developing efficient mechanisms for updating the factual knowledge stored in the model itself (De Cao et al., 2021)
  - Implementing additional knowledge grounding (Zhao et al., 2020),
  - Working on auto-debasing (Guo et al., 2022) procedures, in order to ensure that the chatbots produce correct and factual responses.
  - Finally, we need to develop mechanism that prevent malicious actors to exploit the models (Hancock et al., 2019; Vanderlyn et al., 2021).
- Explainability is now becoming an important research area in NLP (Danilevsky et al., 2020). Some interesting future directions are: methods that focus on explaining the reasoning chain (Yang et al., 2018; Das et al., 2018); 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 (Schick et al., 2022) with automatic edits, adding sources, etc., or obtaining token-level explanations (Li and Yao, 2021; Arora et al., 2022).

Appendices A–B

- **Appendix A** discusses the hyper-parameters used for training the models used in Section 4.3 *Answer Retrieval from a Pool of Explanations*. Moreover, it shows the annotation guideliness, annotator demographics, inter-annotator agreement and disagreement analysis.
- **Appendix B** provides definitions for all subjects included in the $E\chi\alpha\mu s$ dataset (Section 5.5 *Multi- and Cross-Linguality*). Additionally, it describes the fine-tuning procedure and the models' hyper-parameters.

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

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.

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.

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.

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.

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.

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.

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*.

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.

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.

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.

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.

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*.

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.

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*.

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.

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.

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.

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.

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.

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.

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

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.

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.

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.

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.

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.

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.

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.

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. 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. 2020a. Enriched Pre-trained Transformers for Joint Slot Filling and Intent Detection. *arXiv preprint arXiv:2004.14848*.

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.

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.

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.

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*.

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

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

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

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.

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.

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.

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.

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.

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.

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.

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*.

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.

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.

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.

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.

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.

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.

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.

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.

Popat, Kashyap and Mukherjee, Subhabrata and Strötgen, Jannik and Weikum, Gerhard. 2016. Credibility Assessment of Textual Claims on the Web. In *CIKM*.

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.

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.

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.

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.

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.

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.

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*.

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.

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.

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.

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.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training Very Deep Networks. In *Advances in Neural Information Processing Systems*, volume 28.

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.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-range Reasoning for Machine Comprehension. *arXiv preprint arXiv:1803.09074*.

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.

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.

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.

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.

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.

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.

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.

Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

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*.

Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270.

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.

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.

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.

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.

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.