UNIVERSITY OF CALIFORNIA SAN DIEGO

Text Understanding and Question Answering for Consumer Health Applications

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Khalil Mrini

Committee in charge:

Professor Ndapandula Nakashole, Chair
Professor Taylor Berg-Kirkpatrick
Professor Michael Hogarth
Professor Jingbo Shang

2022

The dissertation of Khalil Mrini is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

A mon père, à qui je dois mon éthique de travail.

A ma mère, à qui je dois l'inspiration pour ce doctorat.

A ma femme, à qui je dois mon bonheur et ma motivation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Ndapa Nakashole, for guiding me and providing me with opportunities during my PhD. I am also grateful to my thesis defense committee members, Prof. Taylor Berg-Kirkpatrick, Prof. Michael Hogarth, and Prof. Jingbo Shang.

During my industry internships, I have worked with great collaborators: Franck Dernoncourt, Walter Chang, Trung Bui, Seunghyun Yoon, and Quan Tran at Adobe Research; Markus Dreyer, and Can Liu at Amazon Alexa; Shaoliang Nie, Maziar Sanjabi, Hamed Firooz, Jiatao Gu, and Sinong Wang at Meta AI; Wei Han, Yuan Cao, and Jeffrey Zhao at Google Brain. I would especially like to thank Franck Dernoncourt and Walter Chang for their prolonged collaboration with me, well beyond the internship.

I have had awesome collaborators at UCSD, whom I am grateful to: Naba Rizvi, Casey Mac Meehan, Janet Johnson, Chen Chen, Emilia Farcas, Nadir Weibel, Michael Hogarth, and Allison Moore. My mentee Harpreet Singh was a delightful collaborator during my last PhD project. My friends and colleagues at UCSD have been great support, and I would like to thank Julaiti Alafate, Sophia Sun, Utkrisht Rajkumar, Mary Anne Smart, Fatemehsadat Mireshghallah, Matin Yarmand, Yao Qin, Aditi Ashutosh Mavalankar, Amanda Song, Rajdeep Das, Kai-En Lin, Mohammad Shafiei, Zhi Wang, and many others.

I have made lifelong connections during my PhD. I made many great friends during my only in-person internship at Adobe Research in summer 2019: Xuanli He, Anthony Colas, Logan Lebanoff, Shm Garanganao Almeda, Julia Gong, and many others. During the COVID-19 pandemic, I co-founded two affinity groups that gave me a great sense of community when I needed it: North Africans in NLP, and MoroccoAI. I am very grateful to the people who made this possible: Asma Ben Abacha, Meriem Beloucif, and Nedjma Ousidhoum at North Africans in NLP; Salim Chemlal, Ihsane Gryech, Hicham Hammouchi, Abderrahmane Issam, Imane Khaouja, Redouane Lguensat, and Abdelhak Mahmoudi at MoroccoAI. I would like to especially thank Abderrahmane Issam, who is a knowledgeable, autonomous, and hard-working collaborator.

I was lucky to work with him on our paper on summarization in Moroccan Darija, my passion side project.

I am grateful to the research mentors I had that enabled me to join this PhD program: Prof. Francis Bond, Prof. Pierre Dillenbourg, Prof. Jean-Cédric Chappelier, Dr. Martin Benjamin, Prof. Martin Jaggi, and Dr. Claudiu Musat.

Finally, I am most grateful to my life partner, my love and my personal inspiration: Naba Rizvi. She has single-handedly given me all the happiness, strength and motivation required to finish my PhD. She has always been there to console me during the many difficult moments of my PhD. Without her, you would not be reading this thesis. To finish a difficult life milestone like a PhD, one needs to be complete, and she completes me.

I would like to thank my family for their support, especially my brother Salim Mrini, his wife Asmaa Talbi, and my two, most adorable nieces, Ghita and Sofia. They have supported me through many milestones. Salim is the kindest brother one could wish for, and together with Asmaa they form the most heartwarming pair.

Enfin, je suis éternellement et infiniment redevable à mes deux parents: Abdelghani Mrini et Amina El Ghorfi. Je leur dédie tous mes accomplissements et tous mes succès, parce qu'ils ont tellement sacrifié pour que j'arrive à cette étape cruciale de mon éducation. A chaque étape, bonne ou mauvaise, mes parents m'ont soutenu et appuyé inconditionnellement. Je suis tellement heureux et chanceux d'avoir eu des parents si inspirants et affectueux. Mon père est un travailleur infatigable et un professeur compétent, qui m'a inculqué des valeurs d'assiduité qui m'ont guidé durant mon doctorat. Ma mère, Prof. Amina El Ghorfi, n'a jamais su accepter autre chose que l'excellence de ma part. Son ambition infinie pour mon futur m'a poussé à arriver là où je suis aujourd'hui. Je ne leur dirai jamais autant que je le souhaiterai: je vous aime plus que tout.

Chapter 2, in full, is a reformatted version of the material as it appears in "Recursive Tree-Structured Self-Attention for Answer Sentence Selection," Khalil Mrini, Emilia Farcas, and Ndapa Nakashole [Mrini et al., 2021e]. The material has been submitted, accepted and published

at the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021. The dissertation author was the primary investigator and first author of this paper.

Chapter 3, in full, is a reformatted version of the material as it appears in "A Gradually Soft Multi-Task and Data-Augmented Approach to Medical Question Understanding," Khalil Mrini, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, and Ndapa Nakashole [Mrini et al., 2021c]. The material has been submitted, accepted and published at the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021. The dissertation author was the primary investigator and first author of this paper.

Chapter 4, in full, is a reformatted version of the material as it appears in "Medical Question Understanding and Answering with Knowledge Grounding and Semantic Self-Supervision," Khalil Mrini, Harpreet Singh, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, and Ndapa Nakashole. The material has been submitted to the 29th International Conference on Computational Linguistics, COLING 2022. The dissertation author was the primary investigator and first author of this paper.

Chapter 5, in full, is a reformatted version of the material as it appears in "Detection, Disambiguation, Re-ranking: Autoregressive Entity Linking as a Multi-Task Problem," Khalil Mrini, Shaoliang Nie, Jiatao Gu, Sinong Wang, Maziar Sanjabi, Hamed Firooz [Mrini et al., 2022]. The material has been submitted, accepted and published at the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, held in Dublin, Ireland. The dissertation author was the primary investigator and first author of this paper.

VITA

2013-2016	B. S. in Computer Science, Ecole Polytechnique Fédérale de Lausanne, Switzerland
2015-2016	Exchange Student in Computer Science, Nanyang Technological University, Singapore
2016	Research Intern, Infosys – Bangalore, India
2016-2018	M. S. in Computer Science, Ecole Polytechnique Fédérale de Lausanne, Switzerland
2019	Research Intern, Adobe Research – San Jose, CA
2020	Research Intern, Amazon Alexa – Seattle, WA
2021	Research Intern, Facebook AI – Menlo Park, CA
2021-2022	Research Intern, Google Brain – Mountain View, CA
2018-2022	Ph. D. in Computer Science, University of California San Diego

PUBLICATIONS

Khalil Mrini, Shaoliang Nie, Jiatao Gu, Sinong Wang, Maziar Sanjabi, Hamed Firooz, "Detection, Disambiguation, Re-ranking: Autoregressive Entity Linking as a Multi-Task Problem", *The 60th Annual Conference of the Association for Computation Linguistics, ACL* 2022, in Dublin, Ireland.

Casey Meehan, **Khalil Mrini**, Kamalika Chaudhuri, "Sentence-level Privacy for Document Embeddings", *The 60th Annual Conference of the Association for Computation Linguistics, ACL* 2022, in Dublin, Ireland.

Abderrahmane Issam, **Khalil Mrini**, "Goud.ma: a News Article Dataset for Summarization in Moroccan Darija", *AfricaNLP Workshop at the 10th International Conference on Learning Representations, ICLR* 2022.

Khalil Mrini, Can Liu, Markus Dreyer, "Rewards with Negative Examples for Reinforced Topic-Focused Abstractive Summarization", *Workshop on New Frontiers in Summarization (NewSum) at EMNLP* 2021.

Khalil Mrini, Walter Chang, Trung Bui, Quan Tran, Franck Dernoncourt, "Interpretable Label-Attentive Encoder-Decoder Parser", *United States Patent Application Publication*, 2021.

Khalil Mrini, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, Ndapa Nakashole, "A Gradually Soft Multi-Task and Data-Augmented Approach to Medical Question Understanding", *The 59th Annual Conference of the Association for Computation Linguistics, ACL* 2021.

Khalil Mrini, Emilia Farcas, Ndapa Nakashole, "Recursive Tree-Structured Self-Attention for Answer Sentence Selection', *The 59th Annual Conference of the Association for Computation Linguistics, ACL* 2021.

Chen Chen, **Khalil Mrini**, Kemeberly Charles, Ella Lifset, Michael Hogarth, Alison Moore, Nadir Weibel, Emilia Farcas, "Toward a Unified Metadata Schema for Ecological Momentary Assessment with Voice-First Virtual Assistants", *The ACM Conversational User Interface (CUI) Conference*, 2021.

Khalil Mrini, Franck Dernoncourt, Walter Chang, Emilia Farcas, Ndapa Nakashole, "Joint Summarization-Entailment Optimization for Consumer Health Question Understanding", *Workshop on NLP for Medical Conversations (NLPMC) at NAACL* 2021. **Best Student Paper Award.**

Khalil Mrini, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, Ndapa Nakashole, "UCSD-Adobe at MEDIQA 2021: Transfer Learning and Answer Sentence Selection for Medical Summarization", *Workshop on Biomedical NLP (BioNLP) at NAACL* 2021.

Khalil Mrini, Chen Chen, Ndapa Nakashole, Nadir Weibel, Emilia Farcas. "Medical Question Understanding and Answering for Older Adults", *The 3rd Southern California (SoCal) NLP Symposium*, 2021.

Khalil Mrini, Franck Dernoncourt, Quan Tran, Trung Bui, Walter Chang, Ndapa Nakashole, "Rethinking Self-Attention: Towards Interpretability in Neural Parsing", *The 25th Conference on Empirical Methods in Natural Language Processing, EMNLP* 2020.

Janet Johnson, **Khalil Mrini**, Michael Hogarth, Alison Moore, Ndapa Nakashole, Nadir Weibel, Emilia Farcas, "Voice-Based Conversational Agents for Older Adults", *Workshop on Conversational Agents for Health and Wellbeing at CHI* 2020.

Naba Rizvi, **Khalil Mrini**, "Using HCI to Tackle Race and Gender Bias in ADHD Diagnosis", *Workshop on Engaging in Race in HCI at CHI* 2020.

Khalil Mrini, Franck Dernoncourt, Trung Bui, Walter Chang, Ndapa Nakashole, "Highlights of Attention Mechanisms for Model Interpretability", *The 2nd Southern California (SoCal) NLP Symposium*, Los Angeles, CA, 2019.

Thibault Asselborn, Arzu Guneysu, **Khalil Mrini**, Elmira Yadollahi, Ayberk Ozgur, Wafa Johal, Pierre Dillenbourg, "Bringing letters to life: handwriting with haptic-enabled tangible robots", *The 17th ACM Conference on Interaction Design and Children (IDC)*, Trondheim, Norway, 2018.

Khalil Mrini, Marc Laperrouza, Pierre Dillenbourg, "Building a Question-Answering Chatbot using Forum Data in the Semantic Space", *The 3rd Swiss Text Analytics Conference (SwissText)*, Winterthur, Switzerland, 2018. **Best Presentation Award.**

Khalil Mrini, Francis Bond, "Putting Figures on Influences on Moroccan Darija from Arabic, French and Spanish using the WordNet", *The 9th Global WordNet Conference (GWC)*, Singapore, 2018.

Khalil Mrini, Francis Bond, "Building the Moroccan Darija Wordnet (MDW) using Bilingual Resources", *The International Conference on Natural Language, Signal and Speech Processing (ICNLSSP)*, Casablanca, Morocco, 2017.

Khalil Mrini, Martin Benjamin, "Towards Producing Human-Validated Translation Resources for the Fula language through WordNet Linking", *Workshop on Human-informed Translation and Interpreting Technology (HiT-IT) at RANLP* 2017.

ABSTRACT OF THE DISSERTATION

Text Understanding and Question Answering for Consumer Health Applications

by

Khalil Mrini

Doctor of Philosophy in Computer Science

University of California San Diego, 2022

Professor Ndapandula Nakashole, Chair

The overarching problem that Natural Language Processing (NLP) research tries to solve is linguistic constructs and their meaning. There has been tremendous progress in recent years in contextual text representations, leading to the emergence of self-attention and language models. Language models have pushed the state of the art in question answering, natural language understanding, and a range of other NLP tasks. Whereas language models have revolutionized the way text is represented, they need large amounts of training data, and they may not understand text written in informal, non-mainstream styles. As a result, knowledge-hungry domains such as healthcare or underrepresented users such as older adults have not benefited from this progress.

This dissertation introduces methods that aim to enable language models to adapt to do-

mains with user-generated text as input, and that require specialized knowledge, with challenging, noisy or small training datasets. In particular, I develop methods for text understanding and question answering for consumer health applications, or users of medical language technology systems. First, I tackle Answer Sentence Selection through recursive language models. I show that the popular transformer architecture can leverage tree structures in formally written text, yet fail to do so in informal, user-written text. Then, I propose to better understand user-written questions, or Consumer Health Questions: I propose a new parameter-sharing method that jointly trains question summarization and entailment for the medical domain. Afterwards, I bring together answer selection and question understanding to design a system for medical Question Understanding and Answering. The proposed system takes a long, user-written medical question as input, and selects the best answer from a medical knowledge base using self-supervised losses. Finally, I study text understanding through the lens of entity linking for utterances written by users on social media.

Chapter 1

Introduction

One of the tangible goals of Natural Language Processing (NLP) is equipping computers with the ability to converse with humans at a similar level of intelligence. ELIZA is an early conversational agent [Weizenbaum, 1976], that simulates conversation with humans using rule-based pattern matching. It is estimated to be one of the earliest NLP applications to be able to undergo the Turing test – a philosophical test requiring a computer to communicate such that it is reliably indistinguishable from humans. The current state of NLP research decomposes the goal of passing the Turing test as various subfields, including natural language understanding and question answering.

Question Answering (QA) is the subfield of natural language processing that deals with a computer's ability to provide correct answers to questions asked by users. Question Answering covers a range of tasks, from understanding the intent of questions, to selecting and/or generating content that answers a given question. Applications of question answering pervade our daily lives, through voice assistants like Siri and Alexa, and search engines like Google. Question answering can also be fine-tuned to fit the needs of a particular domain of knowledge. For example, the US National Institutes of Health have developed CHIQA, a question answering system for the medical domain. Their system uses question understanding, a task at the intersection of natural

language understanding and question answering.

Natural Language Understanding (NLU) is the subfield of natural language processing that aims to teach computers to understand the intent of what users write. For example, for a user whose search history relates to Pakistani topics, a query related to "*best restaurants in Hyderabad*" should likely be disambiguated to get results for Hyderabad, Pakistan rather than Hyderabad, India. This particular example is at the intersection of query understanding and entity linking. More generally, NLU can be applied to numerous applications related to disambiguating text. NLU methods can use lexicons like the WordNet [Miller, 1998], or use corpus statistics like in machine translation.

1.1 State of the Art

How can computers best represent the meaning of language? The overarching problem that NLU and NLP try to solve is the representation of text and meaning. Early text representation models, such as word2vec [Mikolov et al., 2013] and GloVe [Pennington et al., 2014], embed words using the co-occurrence of words within windows of context. Recurrent neural networks [Cho et al., 2014, Sutskever et al., 2014] can encode sequences of text and generate text using an encoder-decoder architecture. However, the recurrent nature of the encoder makes it difficult to model long-range dependencies in text. Moreover, the decoder relies on the last encoder output, and the latter may not accurately represent the entire text sequence in the absence of an adequate aggregation mechanism. To resolve these issues, Bahdanau et al. [2014] introduce the attention mechanism, that takes into account sequential order.

The self-attention mechanism and the transformer architecture [Vaswani et al., 2017] aim to better represent the meaning of text – self-attention takes into account all words in a sequence, not just in sequential order. Their introduction was a seismic shift in natural language understanding, and has enabled the emergence of powerful contextualized text representations,

most notably BERT [Devlin et al., 2019a]. Language models like BERT – or Bidirectional Encoder Representations from Transformers – and its derivatives [Yang et al., 2019, Liu et al., 2020, Rogers et al., 2021] are powered by the pre-training of language modeling tasks on huge amounts of text. These language models have pushed the state of the art in various NLP applications, including machine translation [Edunov et al., 2018], abstractive summarization [Lewis et al., 2020a], and syntactic parsing [Mrini et al., 2020].

1.2 Challenges

Whereas language models have revolutionized the way text is represented, they need large amounts of training data, and they may not understand text written in informal, non-mainstream styles. There are two main challenges that we have identified.

Lack of Applicability of Language Models for User-Generated Text. In recent years, NLP conferences – such as NAACL 2021 – have proposed new themes related to understudied problems and methods that enable models to generalize and perform well outside of laboratory settings. Most of the popular benchmark datasets are carefully curated or formally written datasets, and as such do not necessarily represent real-life settings. Therefore the reported performance of a language model in benchmark datasets is not indicative of its generalizing ability, as the performance may drop significantly for user-generated text, such as text found on social media or online forums. In our work, our research question is: how can we develop methods that enable language models to process user-generated text?

Lack of Applicability of Language Models to Domains with Scarce Datasets. As large language models perform best when trained on large training data, their performance is reduced for domains requiring specialized knowledge, and where there is a scarcity of datasets. In this work, we take the consumer health domain as our low-resource domain of application. The consumer health domain is a subset of the medical domain, where the input is user-generated

text. Datasets in this domain are scarce, and vary from curated and small to noisy and large. In this line of work, we ask: how can language models learn domain adaptation efficiently from noisy or small datasets?

1.3 Contributions and Dissertation Organization

This dissertation introduces methods that aim to enable language models to adapt to domains with user-generated text as input, and that require specialized knowledge, with challenging, noisy or small training datasets. We make the following contributions:

Answer Selection through Recursive Language Models. In chapter 2, we investigate whether tree structures can boost performance in Answer Sentence Selection (AS2). We introduce the Tree Aggregation Transformer: a novel recursive, tree-structured self-attention model for AS2. The recursive nature of our model is able to represent all levels of syntactic parse trees with only one additional self-attention layer. Without transfer learning, we establish a new state of the art on the popular TrecQA and WikiQA benchmark datasets. Additionally, we evaluate our method on four Community Question Answering datasets, and find that tree-structured representations have limitations with noisy user-generated text.

Question Understanding through Summarization and Entailment. Having uncovered a weakness in language models for user-generated text in the previous chapter, we propose in chapter 3 a novel Multi-Task Learning (MTL) method with data augmentation for medical question understanding. We first establish an equivalence between the tasks of question summarization and Recognizing Question Entailment (RQE) using their definitions in the medical domain. Based on this equivalence, we propose a data augmentation algorithm to use just one dataset to optimize for both tasks, with a weighted MTL loss. We introduce gradually soft parameter-sharing: a constraint for decoder parameters to be close, that is gradually loosened as we move to the highest layer. Our method outperforms existing MTL methods across 4 datasets of medical question

pairs, in ROUGE scores, RQE accuracy and human evaluation.

Question Understanding and Answering as Summarization and Retrieval. In the previous two chapters, we addressed question understanding and question answering. In chapter 4, we introduce a medical question understanding and answering approach with knowledge grounding and semantic self-supervision. Our system is a pipeline that first summarizes a long, medical, user-written question, using a supervised summarization loss. Then, our system performs a two-step retrieval to return answers. The system first matches the summarized user question with an FAQ from a trusted medical knowledge base, and then retrieves a fixed number of relevant sentences from the corresponding answer document. In the absence of labels for question matching or answer relevance, we design three novel, self-supervised and semantically-guided losses. We evaluate our model against two strong retrieval-based question answering baselines. Evaluators ask their own questions and rate the answers retrieved by our baselines and own system according to their relevance. They find that our system retrieves more relevant answers, while achieving speeds 20 times faster. Our self-supervised losses also help the summarizer achieve higher scores in ROUGE, as well as in human evaluation metrics.

Text Understanding as Entity Linking. In chapter 5, we move beyond the understanding of questions, and we propose to apply understanding to social media utterances using multi-task learning, in a similar approach to chapter 3. We propose an autoregressive entity linking model, that is trained with two auxiliary tasks, and learns to re-rank generated samples at inference time. Our proposed novelties address two weaknesses in the literature. First, a recent method proposes to learn mention detection and then entity candidate selection, but relies on predefined sets of candidates. We use encoder-decoder autoregressive entity linking in order to bypass this need, and propose to train mention detection as an auxiliary task instead. Second, previous work suggests that re-ranking could help correct prediction errors. We add a new, auxiliary task, match prediction, to learn re-ranking. Without the use of a knowledge base or candidate sets, our model sets a new state of the art in two benchmark datasets of entity linking: COMETA in the biomedical

domain, and AIDA-CoNLL in the news domain.

Finally, in Chapter 6, we summarize our contributions and elicit possible directions for future work on enabling NLP techniques to perform well beyond the mainstream benchmarks.

Chapter 2

Answer Selection

In this thesis, we aim to develop methods to build systems that understand the intent expressed by users in text, and that respond to the queries or questions of the users accordingly. To respond to users' questions, the first step we work on is teaching machine learning models to select what answers fit a given question. This chapter focuses on improving and studying the performance of current state-of-the-art systems on the task of Answer Sentence Selection (AS2).

2.1 Introduction

Motivation. Natural language text is characterized by structure. For instance, syntactic parse trees decompose a sentence into syntactic groups, which in turn are decomposed recursively until we get to single-word spans. Therefore, syntactic parse trees have a varying number of levels that can be accurately represented by recursive model architectures.

Tree-structured LSTM networks [Tai et al., 2015] are the recursive extension of LSTM networks [Hochreiter and Schmidhuber, 1997], and allow for syntactic trees to be represented hierarchically. Tree-LSTMs and bidirectional Tree-LSTMs [Teng and Zhang, 2017] do not represent sequence position information, whereas the hybrid neural inference networks [Chen et al., 2017c] represent sequence position information separately from tree-structured hierarchical

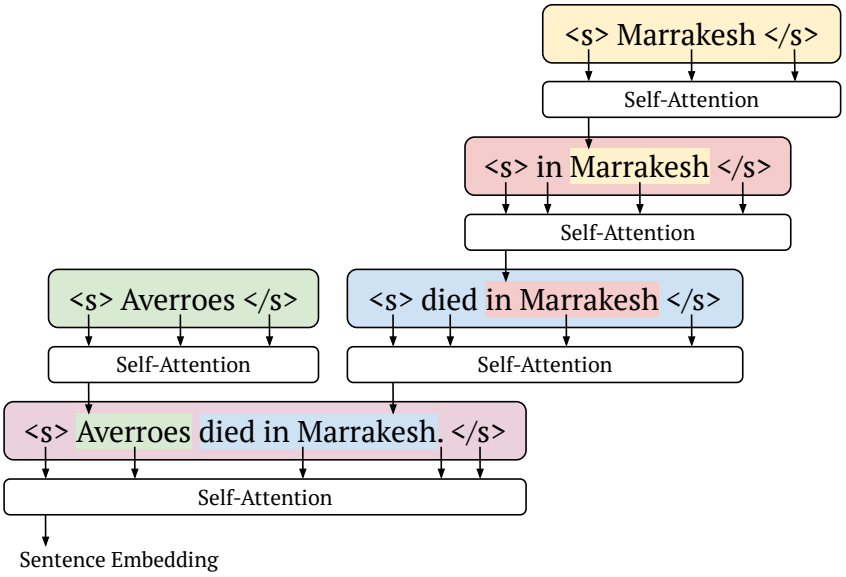


Figure 2.1: Embedding a sentence with our proposed recursive tree-structured self-attention using the corresponding constituency parse tree. There is only one set of parameters for the recursive self-attention.

information.

Tree-structured models have been applied to the tasks of natural language inference [Chen et al., 2017c], sentence pair similarity [Tai et al., 2015], dependency parsing [Kiperwasser and Goldberg, 2016], and text embeddings [Mrini et al., 2019]. In this paper, we consider the problem of Answer Sentence Selection (AS2), where the goal is to predict for a question-sentence pair whether the sentence contains an answer to the question. Given that tree-structured models have performed strongly on a task that takes a sentence pair as input – sentence pair similarity, we hypothesize that tree structures can help in AS2, another sentence pair task.

The most recent top-performing model architectures for Answer Sentence Selection have been based on the self-attention transformer architecture [Vaswani et al., 2017]. Three of them [Lai et al., 2019, Garg et al., 2019, Tran et al., 2020] use transfer learning on large AS2 datasets; another one [Laskar et al., 2020] uses direct fine-tuning on pre-trained transformer-based language encoders, whereas all three use pre-trained BERT [Devlin et al., 2019b] and/or RoBERTa embeddings [Liu et al., 2019b].

Contribution. We investigate whether tree structures are useful for AS2. We introduce the *Tree Aggregation Transformer*: a novel *recursive and tree-structured self-attention model* for Answer Sentence Selection. We use the syntactic parse trees of questions and candidate answer sentences to model them in a tree-structured way. We then form representations for questions and candidate answers using one additional self-attention layer in a recursive, bottom-up fashion, as shown in Figure 2.1. We learn syntactic embeddings to represent hierarchical order and phrase-level syntactic information. We find in an ablation study that our learned syntactic embeddings improve performance.

Without using AS2 datasets for transfer learning, our model establishes a new state of the art for the clean versions of TrecQA and WikiQA, two widely used benchmark datasets in question answering and AS2. Our tree-structured self-attention matches or exceeds the state of the art – which is fine-tuning on RoBERTa – on 2 out of 4 Community Question Answering (CQA) datasets. We conduct experiments for 3 probing tasks to establish what information our models leverage to increase performance, and likewise what they fail to leverage when they do not exceed baselines. We find that tree-structured representations that successfully absorb the provided syntactic information consistently perform better than baselines. Our probing task results suggest that there is more work to be done for tree structures to adapt to noisy user-generated text.

2.2 Related Work

Tree-structured Transformers. To the best of our knowledge, our method is the first to introduce tree self-attention to Answer Sentence Selection. There is a growing body of work incorporating tree structures in self-attention for a range of other NLP tasks.

Nguyen et al. [2019] introduce a transformer-based encoder-decoder that incorporates tree-structured attention. The tree-structured attention is accumulated hierarchically. A token in the tree has as many representations as overall children, therefore it is first accumulated in a

bottom-up fashion (vertically), and then horizontally to compute a token's representation. Their model is not recursive and uses different parameters for each level. The authors evaluate their model in machine translation and text classification.

Sun et al. [2020] develop a tree-structured transformer encoder-decoder architecture for code generation. Here, the tree structure is based on the code syntax. The model uses character-level embeddings as input.

Harer et al. [2019] introduce Tree-Transformer: a model with a tree convolution block for correction of code and grammar. Wang et al. [2019] propose a model of the same name, where the model learns syntactic parse trees in an unsupervised manner. The model uses up to 12 layers of non-recursive self-attention on top of a pre-trained BERT.

Ahmed et al. [2019] introduce Constituency and Dependency Tree Transformer models, largely inspired by the Constituency and Dependency Tree-LSTM models [Tai et al., 2015] and RvNN models [Socher et al., 2011, 2012, 2013]. On 4 datasets of semantic relatedness, natural language inference and paraphrase identification, their transformer models achieve performance on par with Tree-LSTM models, and do not set a new state of the art. The authors use two convolution layers to form a parent representation from the corresponding children. Their model does not learn an explicit syntactic representation, and the authors do not analyze the fluctuating results.

Answer Sentence Selection (AS2). The recent state-of-the-art models in the AS2 task all use transfer learning from large-scale datasets, and do not incorporate syntactic information. All of them use a standard linear (or sequential) input format, where the first input sentence is the question and the second is the candidate answer.

Lai et al. [2019] introduce the Gated Self-Attention Memory Network (GSAMN). It combines gated attention [Dhingra et al., 2017, Tran et al., 2017], memory networks [Sukhbaatar et al., 2015] and self-attention [Vaswani et al., 2017] in one model. The authors use transfer learning with their Stack Exchange QA dataset.

Garg et al. [2019] propose the *TandA* method: Transfer and Adapt. The method is simply fine-tuning directly on a pre-trained BERT or RoBERTa model. The *transfer* step is transfer learning: fine-tuning a large pre-trained BERT or RoBERTa on the ASNQ dataset: a large-scale answer sentence selection dataset extracted from Google's Natural Questions [Kwiatkowski et al., 2019]. The second step is to *adapt* the language model fine-tuned for answer sentence selection to the smaller, target benchmarks TrecQA and WikiQA.

Tran et al. [2020] build upon the work of Lai et al. [2019]. They propose to use a neural Turing machine [Graves et al., 2014] as a controller for the memory network, instead of the gated attention that Lai et al. [2019] use. Like Garg et al. [2019], they use the ASNQ dataset for transfer learning.

Laskar et al. [2020] achieve state-of-the-art results on a wide range of QA and CQA datasets by directly fine-tuning on the target datasets, without transfer learning from an external large-scale dataset. They show results for two methods: the first trains a self-attention layer while freezing pre-trained language model layers, and the second directly fine-tunes on the language model.

2.3 Tree Aggregation Transformer for Answer Sentence Selection

In the AS2 task, the input is a pair of sentences, where the first one is the question and the second is a candidate answer. This is a binary classification problem on whether or not the candidate answer sentence contains an answer to the question. We therefore design our model to form a representation of the question and a representation of the candidate answer, in a bottom-up tree aggregation fashion.

Semantic and Syntactic Representation. We define a token embedding in our input representation as the concatenation of a semantic embedding and a syntactic embedding. The

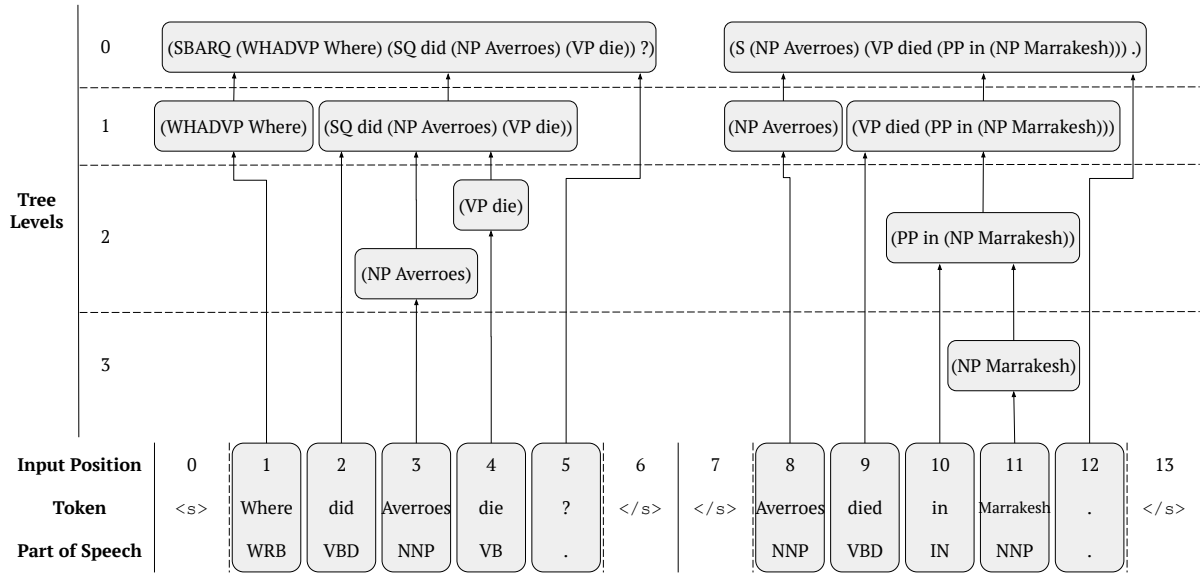


Figure 2.2: Input representation of an example question-sentence pair using RoBERTa.

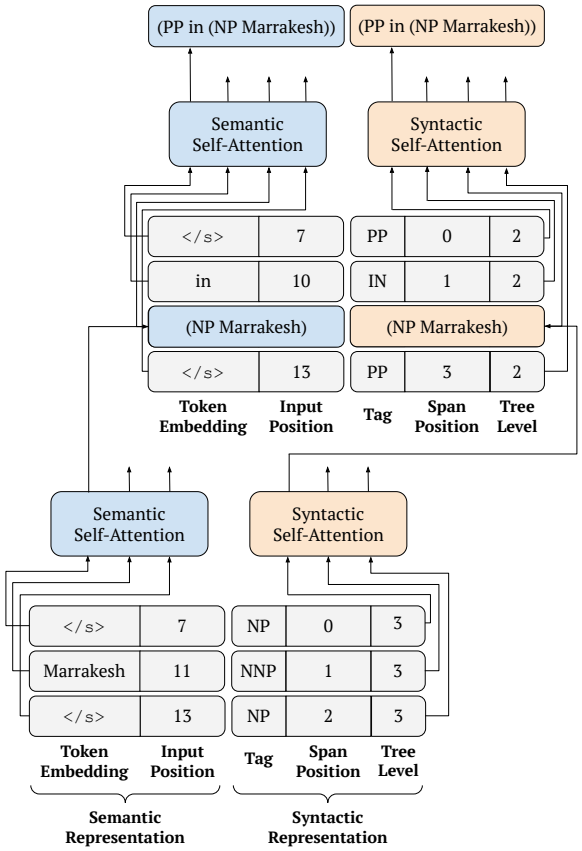


Figure 2.3: Detailed example of recursive tree aggregation.

semantic embedding is a projection of the token embedding from a given pre-trained language model, whereas the syntactic embedding contains information from part-of-speech tags, syntactic categories, and the level within the syntactic parse tree.

The syntactic embedding is the sum of three learned embeddings. The first embedding represents the token's tag – a part-of-speech tag if the token is a word, or a syntactic category if the token is a classification or separator token. The second embedding represents the token's level within the tree, inherited from the head of the token's constituent span. Our recursive model allows to represent sentences with as many tree levels as the corresponding syntax tree has. The third embedding represents the position of a token within the constituent span, as seen in the example in Figure 2.2. This position embedding puts the token within its span context, whereas the position embedding of the semantic (language model) embedding puts the token within the context of the question-sentence pair.

More formally, given a token t, its language model embedding $\mathbf{x_t}$, its position index p_t, its part-of-speech tag or syntactic category s_t, and its tree level l_t, the token's semantic embedding $\mathbf{e_t}$ and syntactic embedding $\mathbf{n_t}$ are as follows:

$$\mathbf{e_t} = \mathbf{W_1} * \mathbf{x_t} + \mathbf{b_1} \tag{2.1}$$

$$\mathbf{n_t} = \mathbf{W_2}\left[\mathbf{E^s}\left[s_t\right] + \mathbf{E^p}\left[p_t\right] + \mathbf{E^l}\left[l_t\right]\right] + \mathbf{b_2} \tag{2.2}$$

where $\mathbf{W_1}$, $\mathbf{W_2}$, $\mathbf{b_1}$, $\mathbf{b_2}$ are learned, and $\mathbf{E^s}$, $\mathbf{E^p}$ and $\mathbf{E^l}$ are learned embedding layers, respectively for the part-of-speech tag or syntactic category, the position index, and the tree level.

Recursive Self-Attention. We add 1 layer of recursive self-attention layer on top of the language model layers. The recursive self-attention layer has separate attention distributions $\mathbf{a_t^e}$ and $\mathbf{a_t^n}$ for the semantic embedding $\mathbf{e_t}$ and syntactic embedding $\mathbf{n_t}$:

$$\mathbf{a_t^e} = \text{softmax}\left(\frac{\mathbf{q_t^e} * \mathbf{K^e}}{\sqrt{d_e}}\right) \tag{2.3}$$

$$\mathbf{a_t^n} = \text{softmax}\left(\frac{\mathbf{q_t^n} * \mathbf{K^n}}{\sqrt{d_n}}\right) \tag{2.4}$$

where d_n and d_e are the dimensions of the query and key vectors for the semantic and syntactic embeddings respectively, and $\mathbf{K^e}$ and $\mathbf{K^n}$ are the learned matrices of key vectors of input tokens. $\mathbf{q_t^e}$ and $\mathbf{q_t^n}$ are the query vectors for the token t, such that:

$$\mathbf{q_t^e} = \mathbf{W^{Q,e}} * \mathbf{e_t} \tag{2.5}$$

$$\mathbf{q_t^n} = \mathbf{W^{Q,n}} * \mathbf{n_t} \tag{2.6}$$

where $\mathbf{W^{Q,e}}$ and $\mathbf{W^{Q,n}}$ are learned.

The resulting vectors $\mathbf{o_t^e}$ and $\mathbf{o_t^n}$ are computed as:

$$\mathbf{o_t^e} = \mathbf{e_t} + \mathbf{W^{O,e}} * (\mathbf{a_t^e} * \mathbf{V^e}) + \mathbf{b^{O,e}} \tag{2.7}$$

$$\mathbf{o_t^n} = \mathbf{n_t} + \mathbf{W^{O,n}} * (\mathbf{a_t^n} * \mathbf{V^n}) + \mathbf{b^{O,n}} \tag{2.8}$$

where $\mathbf{V^e}$ and $\mathbf{V^n}$ are the value vectors for the input tokens, and $\mathbf{W^{O,e}}$, $\mathbf{W^{O,n}}$, $\mathbf{b^{O,e}}$, $\mathbf{b^{O,n}}$ are learned. Finally, we apply separate position-wise feed-forward layers to these output vectors.

Usually, self-attention includes residual dropout over the attention-weighted value vectors. We found in preliminary experiments that the performance on the dev set improved when we omitted dropout regularization. We omit dropout in both self-attention and position-wise feed-forward layer.

The recursiveness of the self-attention allows the model to re-use the same sets of parameters across each tree level, instead of training new ones as in previous work [Nguyen et al., 2019, Wang et al., 2019].

Constituent Span Embedding. Each input sentence is represented in a tree-structured fashion using its constituency parse tree. We use a pre-trained parser, whose parameters are fixed, to produce the trees before training time.

The constituent span is fed to the recursive self-attention as a matrix of token vectors. This matrix includes the embeddings of the words of the constituent span, preceded by a first, start-of-sentence embedding, and followed by an end-of-sentence embedding. The start-of-sentence token is the classification token if the span is part of the question, or a separator token if the span is part of the candidate sentence. Figure 2.3 shows how we compose a constituent span embedding for RoBERTa models.

The constituent span embedding is the output embedding of the first token. The first token embedding obtains through the recursive self-attention an attention-weighted sum of all of the span's token embeddings. This creates a span-specific embedding, conscious of the entire question-sentence pair input as a result of the language model layers, but focused on the tokens of a span as a result of the recursive self-attention.

In using only one layer of recursive self-attention, the first token embedding gets an attention-weighted sum of value vectors that contains token embeddings that did not go through a layer of self-attention, and syntactic embeddings that came directly out of the embedding layers.

Efficient Tree Aggregation. To obtain an aggregate sentence embedding, we proceed by embedding from the deepest level of the tree (the leaves) to the root, as shown in Figure 2.3. The computations are done on the same two sets of self-attention parameters.

To reduce training time, we compute the constituent span embeddings one level at a time. For instance, in Figure 2.2, we compute the NP, VP and PP groups at once when computing the span embeddings at tree level 2.

We efficiently compute all span embeddings only once, and keep all computed span embeddings, as they will be used in the next level.

The sentence embedding is obtained from the first token output of the computation at the root of the tree, as shown in Figure 2.1.

Prediction. Finally, we concatenate the aggregate embeddings for the question-sentence input pair. Given the question's aggregate semantic embedding $\mathbf{w_q^e}$ and aggregate syntactic embedding $\mathbf{w_q^n}$, and the sentence's aggregate semantic embedding $\mathbf{w_s^e}$ and aggregate syntactic embedding $\mathbf{w_s^n}$, we obtain the prediction values as follows:

$$\mathbf{p}(\mathbf{s}|\mathbf{q}) = \text{softmax}\left(\mathbf{W} * \tanh\left[\mathbf{w_q^e}; \mathbf{w_q^n}; \mathbf{w_s^e}; \mathbf{w_s^n}\right] + \mathbf{b}\right) \tag{2.9}$$

where $\mathbf{W}$ and $\mathbf{b}$ are learned. We use binary cross-entropy as our loss function.

Our model can optionally include a residual connection, by adding the classification token embedding output of the language model to the beginning of the question-sentence pair vector. This residual connection does not contain syntactic information, and the classification token embedding is not projected in this case.

2.4 Experiments

2.4.1 Datasets

We evaluate our proposed *Tree Aggregation Transformer* on six English-language benchmark datasets for answer sentence selection. The first two – TrecQA and WikiQA – are widely used benchmarks in Question Answering (QA). The other four – YahooCQA and SemEval 2015, 2016 and 2017 – are all from the Community Question Answering (CQA) domain. We show the statistics of these six datasets in Table 2.1.

TrecQA [Wang et al., 2007] is collected from labeled sentences of the QA track of the Text REtrieval Conference (TREC). Over time, the dataset has evolved into two versions: the raw

version includes all question-sentence pairs, whereas the clean version excludes questions with only non-relevant or only relevant candidate answers.

WikiQA [Yang et al., 2015a] contains questions originally sampled from Bing query logs, and matched with candidate answer sentences from the first paragraph of relevant Wikipedia articles. Likewise, it also has a raw and a clean version. Following Lai et al. [2019], Tran et al. [2020], we evaluate our method on the clean versions of TrecQA and WikiQA.

YahooCQA [Tay et al., 2017] is a filtered and pre-processed subset of the large-scale *Yahoo! Answers Manner Questions* dataset [Surdeanu et al., 2008]. The latter is based on the *Yahoo! Answers* online forum.

SemEval 2015 CQA [Nakov et al., 2015] is the challenge dataset of Subtask A of Task 3 of SemEval 2015. It is based on the Qatar Living online forum, and the goal is to predict the relevance scores of candidate answers given a question. The original subtask divides labels into three categories: definitely relevant, potentially useful, and irrelevant. Following previous work [Sha et al., 2018, Laskar et al., 2020], only definitely relevant candidate answers are marked as relevant in our binary classification setting.

SemEval 2016 CQA [Nakov et al., 2016] corresponds as well to Subtask A of Task 3 of SemEval 2016, about question-comment similarity. It is a new dataset also based on the Qatar Living online forum. The training set includes the training, development and testing sets of the SemEval 2015 CQA, and two new training sets. The authors of the dataset have described the first one as highly reliable, and the second one as noisier.

SemEval 2017 CQA [Nakov et al., 2017] is the latest version of the community question answering task. The training and development sets are the same as the 2016 version, but the testing set is different.

In Figure 2.2, we show an example of question-sentence pairs for a QA dataset and a CQA dataset. The aim is to illustrate the difference in style and length between formal (QA) and informal (CQA) text.

Table 2.1: Statistics of the six benchmark datasets.

Dataset		Number of Questions			Number of Answers		
		Train	Dev	Test	Train	Dev	Test
TrecQA Clean		1,229	65	68	53,417	1,117	1,442
WikiQA Clean		873	126	243	8,672	1,130	2,351
YahooCQA		50,112	6,289	6,283	253,440	31,680	31,680
SemEval	2015	2,600	300	329	16,541	1,645	1,976
	2016	4,879	244	327	36,198	2,440	3,270
	2017	4,879	244	293	36,198	2,440	2,930

Table 2.2: Samples of question-sentence pairs from the training sets of WikiQA and SemEval 2016-2017 (both years share the same training dataset). Here, the sentence contains an answer to the question.

Dataset	Question	Answer
WikiQA	how are glacier caves formed ?	A glacier cave is a cave formed within the ice of a glacier .
SemEval 2016-2017	Why people are crossing red signals on Doha Roads? I think signals are changing quickly than on Dubai roads and its hard for the motorists to control their vehicles? Moreover; motorists are bit panic fearing the penalties as per the new traffic law.	also i traffic lights here does not have standard options. some have blinking green light; some chage to yellow right away then red. several times alredy i found my self driving in the middle of the crossing in red light luckily at the moment no fines. hehehe :) pykester

2.4.2 Setup

The standard evaluation metrics in answer sentence selection are Mean Average Precision (MAP) and Mean Reciprocical Rank (MRR). Both metrics are widely used in Information Retrieval (IR) and are averaged per query – in this case per question. Our model produces relevance scores going from 0 (irrelevant) to 1 (relevant) for each candidate answer, and therefore produces a list of candidate answers that can be ranked by relevance. Whereas MRR scores how

Table 2.3: Ablation study on syntactic representations: Results for our Tree Aggregation Transformer with and without learned syntactic embeddings for all of our benchmark dev sets, on RoBERTa Large.

Representation	TrecQA		WikiQA		YahooCQA		SemEval CQA			
							2015		2016-2017	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
Semantic Only	0.932	0.958	0.892	0.901	0.929	0.929	**0.947**	0.959	0.911	0.950
Semantic + Syntactic	**0.946**	**0.961**	**0.898**	**0.912**	**0.933**	**0.933**	0.945	**0.962**	**0.914**	**0.957**

early a first relevant answer appears in that candidate list, MAP scores the order in which all candidate answers are listed for each question.

To produce parse trees, we use the NLTK part-of-speech tagger [Loper and Bird, 2002] trained on the part-of-speech tagset of the English Penn Treebank (PTB) [Marcus et al., 1994], and the English-language parser of Mrini et al. [2020], which is the state of the art on the parse trees of the PTB.

2.4.3 Training Parameters

We use 1 layer of recursive self-attention for all datasets. We use the residual connection described in §2.3 for TrecQA only. For all our models, we use either BERT large or RoBERTa large, so as to match our baselines. Our recursive self-attention layers have: 16 attention heads, a feed-forward dimension of 4096, and a hidden dimension of 2048. We use half of the dimensions to encode semantic information, and the rest to encode syntactic information.

2.4.4 Ablation Study on Syntactic Embeddings

We perform an ablation study by removing the syntactic embedding part of the input representation. In this experiment, we are quantifying the added value of the learned syntactic embeddings for span position, part-of-speech tags and syntactic categories, and tree levels.

Our results on the dev sets are in Table 2.3. SemEval 2016 and 2017 results are the same since both have the same dev set. Across all AS2 datasets, we notice that there is an advantage to learning syntactic embeddings, as the sum of MRR and MAP scores are higher for the variant that includes learned syntactic embeddings. The advantage is clearer for QA datasets, suggesting that formal language tends to benefit more from learned syntactic information. We use syntactic embeddings in our next experiments.

2.4.5 Baselines

We conside five strong baselines, described in §2.2:

(1) GSAMN [Lai et al., 2019]: Gated Self-Attention Memory Networks.

(2) TandA [Garg et al., 2019]: the two-step Transfer and Adapt method.

(3) Regular Self-Attention [Laskar et al., 2020]: a self-attention layer fine-tuned over frozen BERT Large embeddings.

(4) Direct Fine-tuning [Laskar et al., 2020]: directly fine-tuning on a pre-trained language model.

(5) Evidence Memory [Tran et al., 2020]: the neural Turing machine as memory controller.

Baselines 1, 2, and 5 are available only on TrecQA and/or WikiQA, whereas baselines 3 and 4 use the exact same datasets as we do.

2.4.6 Results and Discussion

The results of our experiments with the QA datasets are in Table 2.4, and the results of our experiments with CQA datasets are in Table 2.5.

State of the Art in QA datasets

Our results in Table 2.4 establish a new state of the art in TrecQA and WikiQA, two widely used benchmark datasets in answer sentence selection.

In TrecQA, our average of MAP and MRR scores matches the one for *TandA* [Garg et al., 2019] in BERT, without any transfer learning on a large dataset. This shows that our model is able to leverage the tree structure to increase performance on relatively small datasets.

For the RoBERTa results in WikiQA, the added value between the direct fine-tuning and our recursive self-attention confirms that our model is beneficial to formally written text, such as

Table 2.4: Our results in comparison with recent work on the TrecQA and WikiQA benchmark datasets. * indicates use of transfer learning on large-scale datasets.

Model	**TrecQA**		**WikiQA**	
	MAP	**MRR**	**MAP**	**MRR**
Chen et al. [2017b]	0.781	0.851	0.721	0.731
Bian et al. [2017]	0.821	0.899	0.754	0.764
Tay et al. [2018a]	0.784	0.865	0.712	0.727
Chen et al. [2018b]	0.823	0.889	0.736	0.745
Chen et al. [2018a]	0.841	0.917	0.730	0.743
Sha et al. [2018]	-	-	0.746	0.758
Madabushi et al. [2018]	0.865	0.904	-	-
Tymoshenko and Moschitti [2018]	-	-	0.762	0.776
Kamath et al. [2019]	-	-	0.700	0.716
Models using BERT Large				
GSAMN [Lai et al., 2019]*	0.914	0.957	**0.857**	**0.872**
TandA [Garg et al., 2019]*	0.912	**0.967**	-	-
Reg. Self-Attention [Laskar et al., 2020]	0.789	0.887	0.714	0.731
Direct Fine-tuning [Laskar et al., 2020]	0.905	**0.967**	0.843	0.857
Our Tree Aggregation Transformer	**0.917**	0.961	0.851	0.868
Models using RoBERTa Large				
TandA [Garg et al., 2019]*	0.943	0.974	-	-
Direct Fine-tuning [Laskar et al., 2020]	0.936	0.978	0.900	0.915
Our Tree Aggregation Transformer	**0.950**	**0.985**	**0.906**	**0.920**
Models using RoBERTa Large and Evidence Memory				
Evidence Memory [Tran et al., 2020]*	0.961	0.993	0.936	0.952
Our Tree Aggregation Transformer	**0.970**	**0.995**	**0.941**	**0.958**

Table 2.5: Our results in comparison with recent work on the YahooCQA and SemEvalCQA benchmark datasets.

Model	YahooCQA		SemEval CQA					
			2015		2016		2017	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
Nakov et al. [2017]	-	-	-	-	-	-	0.884	0.928
Tay et al. [2018a]	0.801	0.801	-	-	-	-	-	-
Sha et al. [2018]	-	-	-	-	0.801	0.872	-	-
Models using BERT Large								
Regular Self-Attention [Laskar et al., 2020]	0.778	0.778	0.883	0.923	0.765	0.831	0.867	0.922
Direct Fine-tuning [Laskar et al., 2020]	**0.951**	**0.951**	0.935	0.961	**0.866**	**0.927**	**0.921**	**0.963**
Our Tree Aggregation Transformer	0.946	0.946	**0.946**	**0.972**	0.844	0.900	0.902	0.955
Models using RoBERTa Large								
Direct Fine-tuning [Laskar et al., 2020]	**0.955**	**0.955**	0.947	0.970	**0.888**	**0.938**	**0.943**	**0.974**
Our Tree Aggregation Transformer	0.949	0.949	**0.961**	**0.981**	0.863	0.918	0.926	**0.974**

Table 2.6: Results for three probing tasks comparing sequential [Laskar et al., 2020] and tree-structured (ours) representations. In the last two columns, we show the Spearman correlation of the probing task and the AS2 performance differences between the tree-structured and sequential representations.

Probing Task	Representation	TrecQA	WikiQA	YahooCQA	SemEval			Spearman's ρ	
					2015	2016	2017	MAP	MRR
Top Constituent Prediction (F1 score)	**Tree-Structured**	0.1573	0.1949	0.0354	0.2058	0.0674	0.1151	0.8214	0.9550
	Sequential	0.0475	0.0463	0.0364	0.0434	0.0505	0.0483		
Tree Depth Prediction (F1 score)	**Tree-Structured**	0.1568	0.1638	0.0354	0.1682	0.0621	0.1340	0.8214	0.9550
	Sequential	0.0481	0.0476	0.0354	0.0451	0.0523	0.0481		
Input Length Regression (MSE)	**Tree-Structured**	0.0266	0.0273	4.51e-06	0.0652	0.0989	0.0416	-0.0360	0.1429
	Sequential	0.0822	0.1200	4.14e-06	0.2915	0.3338	0.1484		

the one found in Wikipedia.

The increase in performance compared to the Evidence Memory models [Tran et al., 2020] when we add our tree representation shows that our tree aggregation method brings about a consistent and robust added value for the QA datasets.

Limitations in CQA datasets

As shown in Table 2.5, our *Tree Aggregation Transformer* is able to establish a new state of the art in SemEval 2015, and our BERT-based version exceeds other BERT-based baselines. However, our method scores below the state of the art in YahooCQA and SemEval 2016, and only manages to match the MRR – but not the MAP – of the state of the art in SemEval 2017.

Therefore, there is a contrast in the performance of our recursive tree-structured self-attention between the QA and the CQA datasets. The difference lies in the style of the datasets, as questions and sentences can be much longer in QA datasets than in CQA datasets. On average, a training set pair in QA has 32 words for WikiQA, and 39 words in TrecQA, whereas a training set pair in CQA has 78 words for SemEval 2015, 85 words for SemEval 2016-2017, and 40 words for YahooCQA. As shown in the example, CQA pairs may also have spelling mistakes or lack coherent structure. Thus, the informal writing style and larger text length of CQA datasets may be decreasing the ability of our model to leverage tree structures. Accordingly, we see that our model achieves very competitive scores for YahooCQA, and that it has a text length that is very close to the QA datasets. The SemEval 2015 exception could be explained by the fact that the 2015 training dataset is less noisy than the 2016-2017 training dataset, as pointed out by the authors of the SemEval CQA datasets.

2.4.7 Do Tree Structures Improve Performance?

We investigate how tree structures are leveraged in the Answer Sentence Selection task across the different datasets. We evaluate our tree-structured representations and compare them

with the corresponding sequential representations, using three probing tasks from Conneau et al. [2018].

Probing Tasks

The three probing tasks are as follows:

(1) Top Constituent Prediction. This task looks to predict the top constituent sequence of the question-sentence pair: the sequence of syntactic categories immediately below the `S` (Sentence) syntactic category. Following Conneau et al. [2018], we define this task as a 20-way classification problem, where the first 19 classes are the 19 most popular top constituent sequences, and the last category is for all the remaining top constituent sequences.

(2) Tree Depth Prediction. The tree depth is the number of hops from the root node of the syntactic tree to the lowest-level leaf nodes.

(3) Input Length Regression. This tasks investigates whether the embedding is aware of how many words it contains. The length of the question-sentence pair input is defined as the number of its tokens – full words and punctuation symbols.

The first two tasks are syntactic, and investigate whether our tree-structured representations absorbed the syntactic category information that we fed it – respectively syntactic categories and tree levels – and whether that information was already present in the sequential representations.

Probing Experiment Setup

In our probing experiments, we consider all six datasets used both in our work and in Laskar et al. [2020]. We consider the sequential representation of a question-answer pair to be the classification token embedding used for prediction in the RoBERTa-based models of Laskar et al. [2020]. We take our own RoBERTa-based tree-structured models (without evidence memory), where we consider the tree-structured representation to be the classification token embedding fed to the prediction layer. The tree-structured and sequential representations have the same number

of dimensions.

The probing model architecture is a simple MLP with a layer of the same size as the input embeddings, a ReLU activation, and a prediction layer. We train 36 probing models for each of the 36 combinations of a probing task, a dataset and a representation type. The input embeddings are frozen, so that the training does not change the weights of the pre-trained AS2 models. All experiments are trained for the same number of epochs, and use the same train/dev/splits as AS2 experiments.

Probing Results and Discussion

Our probing experiment results are shown in Table 2.6. We compute the Spearman correlations of the added values of the tree-structured representations compared to the sequential representations in each probing task with the same added value in the AS2 task. We compute the added value of the tree representation in a given task by subtracting the performance of the sequential representations [Laskar et al., 2020] from the performance of the tree-structured representations (ours).

For the syntactic probing tasks (the first two), the tree-structured representation gets an F1 score about 3 to 4 times higher than the one obtained by the sequential representation in 4 datasets: TrecQA, WikiQA, and SemEval 2015 and 2017. These 4 datasets correspond to the ones in which our tree-structured AS2 models set a new state of the art or matched the performance of the fine-tuning baseline of Laskar et al. [2020]. In the other datasets, the tree-structured representation's F1 score is just slightly higher than the sequential representation's F1 score, if not about the same. This shows that when the tree-structured representations successfully absorb the syntactic information we fed it, there is a consistent increase in performance in the answer sentence selection task. The high correlation values for both MAP and MRR confirm that successfully absorbing syntactic information is associated with higher performance in AS2. The weakness of tree-structured representations in certain datasets may be due to the lack of

generalization of syntactic parsers trained on the Penn Treebank.

In the input length probing experiment, we observe that the mean-squared error (MSE) of the tree-structured representations is consistently and significantly lower than the one of the sequential representations, except for YahooCQA. This shows that the recursion of our tree-structured AS2 model makes representations aware of the length of their question-sentence pair, but the correlation values show that this information does not necessarily help in the AS2 task.

2.5 Conclusions

We introduce the *Tree Aggregation Transformer*: a novel, recursive and tree-structured self-attention model for AS2. Our method embeds sentences by aggregating word representations following the corresponding parse tree. We show that our model leverages tree structure and, through an ablation study, that its learned syntactic embeddings increase performance. Our method establishes a new state of the art in the TrecQA and WikiQA benchmark datasets with only one additional self-attention layer. Our tree-structured self-attention exceeds or matches the state of the art in 2 out of 4 CQA datasets, where text is informal and longer. To investigate this mixed performance, we devise 3 probing tasks to examine what our tree-structured representations learn compared to their sequential counterparts. We find that there is a strong correlation between a tree-structured model's ability to absorb syntactic information and its ability to increase performance in the AS2 task compared to baselines. Our findings suggest that there is more work to be done for tree-structured representations to adapt to noisy user-generated text.

Acknowledgements

This chapter is a reformatted version of the material as it appears in "Recursive Tree-Structured Self-Attention for Answer Sentence Selection," Khalil Mrini, Emilia Farcas, and

Ndapa Nakashole [Mrini et al., 2021e]. The material has been submitted, accepted and published at the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021. The dissertation author was the primary investigator and first author of this paper.

We gratefully acknowledge the award from NIH/NIA grant R56AG067393. This work is part of the VOLI project [Mrini et al., 2021a, Johnson et al., 2020].

Chapter 3

Question Understanding

In the previous chapter, we studied current state-of-the-art systems in answer sentence selection, and found that the performance in this task is worse for informal, user-written text, compared to formally worded text. Therefore, we cannot directly use NLP systems to understand intents in user queries. In this chapter, we tackle Question Understanding: we introduce multi-task learning methods to teach models to summarize long and informal user questions into short, single-sentence and formally worded questions.

3.1 Introduction

In order to retrieve relevant answers, one of the basic steps in Question Answering (QA) systems is understanding the intent of questions [Chen et al., 2012, Cai et al., 2017]. This is particularly important for medical QA systems [Wu et al., 2020a], as consumer health questions – questions asked by patients – may use a vocabulary distinct from doctors to describe similar health concepts [Ben Abacha and Demner-Fushman, 2019b]. Consumer health questions may also contain peripheral information like patient history [Roberts and Demner-Fushman, 2016], that are not necessary to answer questions. There is a growing number of approaches to medical question understanding, including query relaxation [Ben Abacha and Zweigenbaum, 2015, Lei

Source User-written Question or *Consumer Health Question (CHQ)*: SUBJECT: Morgellon Disease. MESSAGE: It appears as if I have had this horrible disease for many, many years and it is getting worst. I am trying to find a physician or specialist in the South Carolina area who can treat me for this medical/mental disease. It seems as if this disease has "NO" complete treatment and it is more least a disability!
Reference Summarized Question or *Frequently Asked Question (FAQ)*: What are the treatments for Morgellon Disease, and how can I find physician(s) in South Carolina who specialize in it?
BART Trained on Summarization Loss Only (Baseline): Where can I find physician(s) who specialize in morgellon disease?
Our Gradually Soft Multi-Task and Data-Augmented Model: Where can I find a physician or specialist in South Carolina who can treat Morgellon Disease?

Figure 3.1: We highlight the main four aspects of the CHQ. Our method learns from the task of Recognizing Question Entailment to generate more informative summaries compared to the baseline.

et al., 2020], question entailment [Ben Abacha and Demner-Fushman, 2016, 2019a, Agrawal et al., 2019], question summarization [Ben Abacha and Demner-Fushman, 2019b], and question similarity [Ben Abacha and Demner-Fushman, 2017, Yan and Li, 2018, McCreery et al., 2019].

Medical question summarization is the task of summarizing consumer health questions into short, single-sentence questions that capture essential information needed to give a correct answer. The task of Recognizing Question Entailment (RQE) is defined by Ben Abacha and Demner-Fushman [2016] in the medical domain as a binary classification task. For the purpose of this task, a first question is considered to entail a second one if and only if every answer to the second question is a correct, and either full or partial answer to the first question.

We find in initial experiments [Mrini et al., 2021b] that RQE can teach question summarizers to distinguish salient information from peripheral details, and likewise that question summarization can benefit RQE classifiers. In our setting, we cast the medical question understanding task as a Multi-Task Learning (MTL) problem involving the two tasks of question summarization and Recognizing Question Entailment. We use a simple sum of learning objectives in Mrini et al. [2021b]. In this paper, we introduce a novel, *gradually soft multi-task and*

data-augmented approach to medical question understanding.[1]

Previous work on combining summarization and entailment uses at least 2 datasets – 1 from each task [Pasunuru et al., 2017, Guo et al., 2018]. We first establish an equivalence between both tasks. This equivalence is the inspiration behind the data augmentation schemes introduced in our previous work [Mrini et al., 2021b]. The goal of the data augmentation is to use a single dataset for Multi-Task Learning. We propose to use a weighted loss function to simultaneously optimize for both tasks. Then, we propose a gradually soft parameter-sharing MTL approach. We conduct ablation studies to show that our two novelties – data augmentation and gradually soft parameter-sharing – improve performance in both tasks.

Our proposed gradually soft multi-task and data-augmented approach outperforms existing single-task and multi-task learning methods on architectures achieving state-of-the-art results in abstractive summarization. Compared to single-task learning, our approach achieves a 12% increase in accuracy on a medical RQE dataset, and an average increase of 3.5% in ROUGE-1 F1 scores across 3 medical question summarization datasets. Additionally, we perform human evaluation and find our approach generates more informative summarized questions. Finally, we find that our approach is more efficient at leveraging smaller amounts of data, and yields better performance under 4 low-resource settings.

3.2 Background and Related Work

Recognizing Question Entailment (RQE). Ben Abacha and Demner-Fushman [2016] introduce the task of RQE. It is closely related — but not exactly similar — to the task of Recognizing Textual Entailment (RTE) [Dagan et al., 2005, 2013], and early definitions of question entailment [Groenendijk and Stokhof, 1984, Roberts, 1996].

The task of RQE is to predict, given two pairs of questions A and B, whether A entails B.

[1] Our code is available at: `https://github.com/KhalilMrini/Medical-Question-Understanding`

RQE considers that question A entails question B if every answer to B is a correct answer to A, and answers A either partially or fully. It differs from traditional definitions of entailment, where we consider that the premise entails the hypothesis if and only if the hypothesis is true only if the premise is true.

Ben Abacha and Demner-Fushman [2016] define RQE within the context of Medical Question Answering. The goal is to match a Consumer Health Question (CHQ) to a Frequently Asked Question (FAQ), and ultimately match the CHQ to an expert-written answer.

Summarization and Entailment. There is a growing body of work combining summarization and entailment [Lloret et al., 2008, Mehdad et al., 2013, Gupta et al., 2014].

Falke et al. [2019] use textual entailment predictions to detect factual errors in abstractive summaries generated by state-of-the-art models. Pasunuru and Bansal [2018] propose an entailment reward for their reinforced abstractive summarizer, where the entailment score is obtained from a pre-trained and frozen natural language inference model.

Pasunuru et al. [2017] propose an LSTM encoder-decoder model that incorporates entailment generation and abstractive summarization. The authors optimize alternatively between the two tasks, and use separate Natural Language Inference (NLI) and abstractive summarization datasets. Only the decoder parameters are shared.

Li et al. [2018] closely follow the MTL setting of Pasunuru et al. [2017], and propose a model with a shared encoder, an NLI classifier and an NLI-rewarded summarization decoder.

Guo et al. [2018] introduce a pointer-generator summarization model with coverage loss [See et al., 2017]. They build upon the work of Pasunuru et al. [2017], and add question generation on top of the two tasks of abstractive summarization and entailment generation. They also alternate between the three different objectives. The authors propose to share all parameters except the first layer of the encoder and the last layer of the decoder, and show that soft parameter-sharing improves over hard parameter-sharing. Their method outperforms the pointer-generator networks of See et al. [2017] on the CNN-Dailymail news summarization baseline. Here, the

authors show performance increase in entailment on some batch sizes and decrease on other batch sizes, and they consider entailment as an auxiliary task.

Transfer Learning for Medical QA. BioNLP is one of many NLP applications to benefit from language models that use multi-task learning and transfer learning. There are pretrained language models that are geared towards BioNLP applications, that are based on BERT [Devlin et al., 2019b]. Those include SciBERT [Beltagy et al., 2019] which has been fine-tuned using biomedical text from PubMed. BioBERT [Lee et al., 2020] has been fine-tuned on the PMC dataset, whereas models named ClinicalBERT [Huang et al., 2019, Alsentzer et al., 2019] additionally use the MIMIC III dataset [Johnson et al., 2016].

Transfer learning was a popular approach at the 2019 MEDIQA shared task [Ben Abacha et al., 2019] on medical NLI, RQE and QA. The question answering task involved re-ranking answers, not generating them [Demner-Fushman et al., 2020]. For the RQE task, the best-performing model [Zhu et al., 2019] uses transfer learning on NLI and ensemble methods.

3.3 Methodology

We consider the multi-task learning of medical question summarization and medical RQE. The input to both tasks is a pair of medical questions. The first question is called a Consumer Health Question (CHQ), and the second question is called a Frequently Asked Question (FAQ). The CHQ is written by a patient and is usually longer and more informal, whereas the FAQ is usually a single-sentence question written by a medical expert. The purpose of both tasks is to match a CHQ to an FAQ, and ultimately to an expert-written answer that matches the FAQ. An example pair is shown in Figure 3.1.

Our novel gradually soft multi-task and data-augmented learning approach to medical question understanding has four main components. First, we establish the equivalence between medical question pairs in question summarization and RQE. Then, we use our equivalence

observation to propose a scheme for data augmentation. Third, we show our simultaneous multi-task learning model architecture and learning objective. Finally, we describe our gradually soft parameter-sharing scheme.

3.3.1 Equivalence of Question Summarization and RQE

In the following, we evidence the equivalence between medical question summarization and medical RQE. We first consider a pair of medical questions C and F, where C is a CHQ and F and is an FAQ, such that C is longer than F.

Ben Abacha and Demner-Fushman [2016] define question entailment as: question C entails question F ($C \Rightarrow F$) if and only if every answer to F is also a correct answer to C, whether partially or completely **(1)**.

According to the guidelines set in the data creation of a medical question summarization dataset by Ben Abacha and Demner-Fushman [2019b], doctors were told to grade manually written summarized questions (FAQs) as perfect, acceptable or incorrect. The two conditions for a perfect FAQ are: first, an FAQ should enable to retrieve "*complete and correct answers*" to the original CHQ, and second, the summarized question should not be so short that it violates the first condition. The resulting medical question summarization dataset includes perfect and acceptable FAQs. We assume that a perfect FAQ provides complete and correct answers to the corresponding CHQ, and that an acceptable FAQ provides correct answers to the corresponding CHQ, whether partially or completely. We therefore conclude that: F is a good summary of C, if and only if F enables to retrieve correct answers to C, whether partially or completely **(2)**.

We have: F enables to retrieve correct answers to C, if and only if answers to F are correct answers to C. Therefore, F enables to retrieve correct answers to C, if and only if every answer to F is also a correct answer to C, whether partially or completely. Given the equivalences **(1)** and **(2)** above, it follows that: question F is a good summary of question C, if and only if question C entails question F **(3)**.

3.3.2 Data Augmentation

Medical question understanding datasets are scarce, and new high-quality datasets are complex and costly to create. We propose in Mrini et al. [2021b] to augment existing datasets in one of the two tasks to create a synthetic dataset of the same size for the other task. Our two-way data augmentation algorithm is inspired by the equivalence shown in the previous subsection, and enables us to train in a simultaneous multi-task setting. Our data augmentation method also addresses a weakness in previous work in multi-task learning, where each task involves a distinct dataset, often from a different domain. Our data augmentation will enable us to use datasets in the same domain, and we hypothesize this can benefit performance in both tasks.

For summarization datasets, we create equivalent RQE pairs. For each existing summarization pair, we first choose with equal probability whether the equivalent RQE pair is labeled as entailment or not. If it is an entailment case, we use the equivalence in **(3)** and create an RQE pair identical to the summarization pair. If it is not an entailment case, then we have: **(3)** $\Leftrightarrow$ question F is not a summary of question C if and only if question C does not entail question F **(4)**. Therefore, to create an equivalent RQE pair labeled as not entailement, the RQE CHQ is identical to the CHQ of the summarization pair, and the RQE FAQ is randomly selected from a distinct question pair from the same dataset split.

Inversely, for the RQE dataset, we create equivalent summarization pairs. For each existing RQE pair, we consider two cases. If the RQE pair is labeled as entailment, we create an identical summarization pair. If the RQE pair is labeled as not entailment, then following **(4)**, we create a summarization pair that is identical to a randomly selected and distinct RQE pair labeled as entailment from the same dataset split.

3.3.3 Simultaneous Multi-Task Learning

Previous work on multi-task learning with summarization and entailment [Pasunuru et al., 2017, Guo et al., 2018] optimize for the objectives of the different tasks by alternating between them. This alternating multi-task training follows a ratio between the different tasks, that depends on the size of the dataset of each task (e.g. a ratio of 10:1 means training for 10 batches on one task, and then for 1 batch on the other task). In our approach, we propose to optimize simultaneously for the objectives of both tasks. We do not use ratios, as we are not alternating between objectives and the resulting datasets from our data augmentation algorithm are of equal size.

Whereas many previous multi-task settings chose generation tasks (entailment generation and question generation), we choose the BART Large architecture [Lewis et al., 2019] as it enables to optimize for a classification task (RQE) and a generation task (summarization) using the same architecture. In addition, BART is adequate as it achieves very strong results in benchmark datasets of recognizing textual entailment and abstractive summarization. The input works differently between both tasks. For summarization, the encoder takes the CHQ as input and the decoder takes the FAQ as input. For RQE, both the encoder and decoder take the entire RQE pair as input. We add a classification head for RQE, to which we feed the last decoder output, as it attends over all decoder and encoder positions. We show an overview of our architecture in Figure 3.2.

We propose to optimize a single loss function that combines objectives of both tasks. Our loss function is the weighted sum of the negative log-likelihood summarization objective, and the binary cross-entropy classification objective of RQE.

More formally, given a CHQ embedding $\mathbf{x}$, the corresponding FAQ embedding $\mathbf{y}$, and the entailment label $l_{entail} \in \{0, 1\}$, we optimize the following multi-task learning loss function:

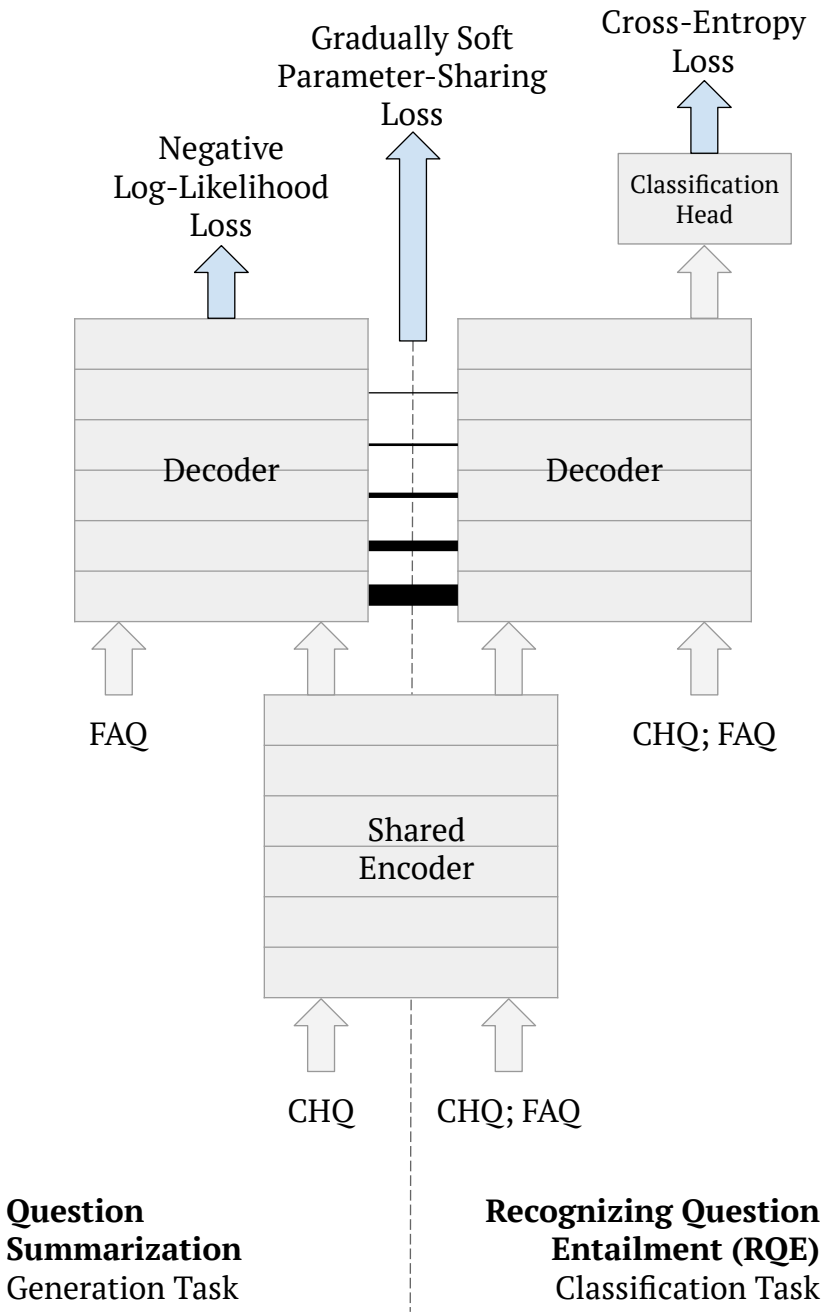


Figure 3.2: Overview of the architecture of our proposed gradually soft multi-task and data-augmented model. The gradually thinning links between decoder layers represent the loosening parameter-sharing constraint.

$$\begin{aligned}\mathcal{L}_{\text{MTL}}(\theta) = &-\lambda * \log p(\mathbf{y}|\mathbf{x};\theta) \\ &+ (1-\lambda) * \text{BCE}\left([\mathbf{x};\mathbf{y}], l_{entail};\theta\right)\end{aligned} \tag{3.1}$$

where BCE is binary cross entropy, and λ is a hyperparameter between 0 and 1.

3.3.4 Gradually Soft Parameter-Sharing

In multi-task learning, there are two widely used approaches: hard parameter-sharing and soft parameter-sharing. Guo et al. [2018] propose soft parameter-sharing for all parameters except the first layer of the encoder and last layer of the decoder. Liu et al. [2019a] introduce MT-DNN and show that hard parameter-sharing of all of the transformer encoder layers, and only having task-specific classification heads produces results that set a new state of the art for the GLUE benchmark [Wang et al., 2018].

We propose a hybrid approach, where we apply hard parameter-sharing for the encoder, and a novel *gradually soft parameter-sharing* approach for the decoder layers. We define gradually soft parameter-sharing as a smooth transition from hard parameter-sharing to task-specific layers. It is a soft parameter-sharing approach that is gradually toned down from the first layer of the decoder to the last layer, which is entirely task-specific.

In gradually soft parameter-sharing, we constrain decoder parameters to be close by penalizing their l_2 distances, and the higher the layer the looser the constraint. Given a decoder with N layers, the gradually soft parameter-sharing loss term is as follows:

$$\mathcal{L}_{\text{GS}}(\theta) = \gamma * \sum_{n=1}^{N-1} \left(e^{\frac{N-n}{N}} - 1\right) \left\| \theta^{\text{QS}}_{\text{dec},n} - \theta^{\text{RQE}}_{\text{dec},n} \right\|^2 \tag{3.2}$$

where γ is a hyperparameter, $\theta^{\text{QS}}_{\text{dec},n}$ represents the decoder parameters for the question summarization at the n-th layer, and likewise $\theta^{\text{RQE}}_{\text{dec},n}$ represents the decoder parameters for the RQE task

Table 3.1: Statistics of the medical dataset splits.

DATASET	TRAIN	DEV	TEST
MeQSum	400	100	500
HealthCareMagic	181,122	22,641	22,642
iCliniq	24,851	3,105	3,106
MEDIQA RQE	8,588	302	230

at the n-th layer. We iterate from the 1st to the $(N-1)$-th layer, as the N-th layer is entirely task-specific and unconstrained. We show a high-level representation in Figure 3.2.

3.4 Experiments

3.4.1 Datasets

We consider 3 medical question summarization datasets and 1 medical RQE dataset. We show dataset statistics in Table 3.1. MeQSum and MEDIQA RQE can be considered low-resource, whereas the other two are far larger. Our datasets are in the English language. Due to space constraints, we briefly introduce the datasets and leave additional details in the appendix.

The medical question summarization datasets are MeQSum [Ben Abacha and Demner-Fushman, 2019b], HealthCareMagic and iCliniq. We extract in Mrini et al. [2021b] and in Mrini et al. [2021d] the HealthCareMagic and iCliniq datasets from the large-scale MedDialog dataset [Chen et al., 2020]. Whereas MeQSum is a high-quality dataset from the U.S. National Institutes of Health (NIH), HealthCareMagic and iCliniq are from online healthcare service platforms. HealthCareMagic's summaries are more abstractive and are written in a formal style, unlike iCliniq's patient-written summaries.

The medical RQE dataset is the MEDIQA RQE dataset from the 2019 MEDIQA shared task [Ben Abacha et al., 2019]. Similarly to MeQSum, the question pairs match a longer CHQ received by the U.S. National Library of Medicine (NLM) and a FAQ from NIH institutes. Whereas the train and dev sets have automatically generated CHQs, the test set has manually

Table 3.2: Dev set results for the ablation studies on our two main novelties: our data augmentation algorithm, and our gradually soft parameter-sharing method. The R1, R2 and RL metrics refer to the F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L [Lin, 2004].

DATASET	MeQSum			HealthCareMagic			iCliniq			RQE
METRIC	R1	R2	RL	R1	R2	RL	R1	R2	RL	Accuracy
ABLATION OF DATA AUGMENTATION										
Gradually Soft MTL + Existing Dataset	51.3	32.3	47.5	45.1	22.9	40.3	59.4	46.0	54.5	81.1%
ABLATION OF GRADUALLY SOFT PARAMETER-SHARING										
Hard-shared Decoder + Data Aug.	52.0	34.0	47.9	44.3	23.3	41.5	60.1	47.0	56.3	77.5%
Soft-shared Decoder + Data Aug.	53.2	35.6	48.9	44.8	22.8	40.9	60.7	48.3	57.8	79.4%
Task-specific Decoder + Data Aug.	50.8	31.7	45.4	46.0	25.1	43.4	61.8	47.5	56.9	81.8%
OUR MODEL										
Gradually Soft MTL + Data Aug.	54.5	37.9	50.2	46.9	24.8	43.2	62.3	48.7	58.5	82.1%

written CHQs. This results in significantly higher dev set results than for test sets, as has been observed during the 2019 MEDIQA shared task.

In addition, we use two pretraining datasets. We use the XSum dataset [Narayan et al., 2018], an abstractive summarization benchmark, for question summarization. For the RQE task, we use the Recognizing Textual Entailment (RTE) dataset [Dagan et al., 2005, Haim et al., 2006, Giampiccolo et al., 2007, Bentivogli et al., 2009] from the GLUE benchmark [Wang et al., 2018].

3.4.2 Setup and Training Settings

All of our models use the BART large architecture. Unless otherwise noted, all experiments on the 3 question summarization datasets are made using a checkpoint pre-trained on the XSum dataset using only the summarization objective, and all experiments on the RQE dataset are made using a checkpoint pre-trained on the RTE dataset, only optimizing the cross-entropy loss.

We report ROUGE F1 scores for the question summarization datasets, and accuracy for the RQE dataset, as it is a binary classification task with two labels: entailment and not entailment.

The learning rate for RQE experiments is 1×10^{-5} and for the question summarization experiments, it is 3×10^{-5}. We use an Adam optimizer where the betas are 0.9 and 0.999 for summarization, and 0.9 and 0.98 for RQE. In all experiments, the Adam epsilon is 10^{-8}, and the

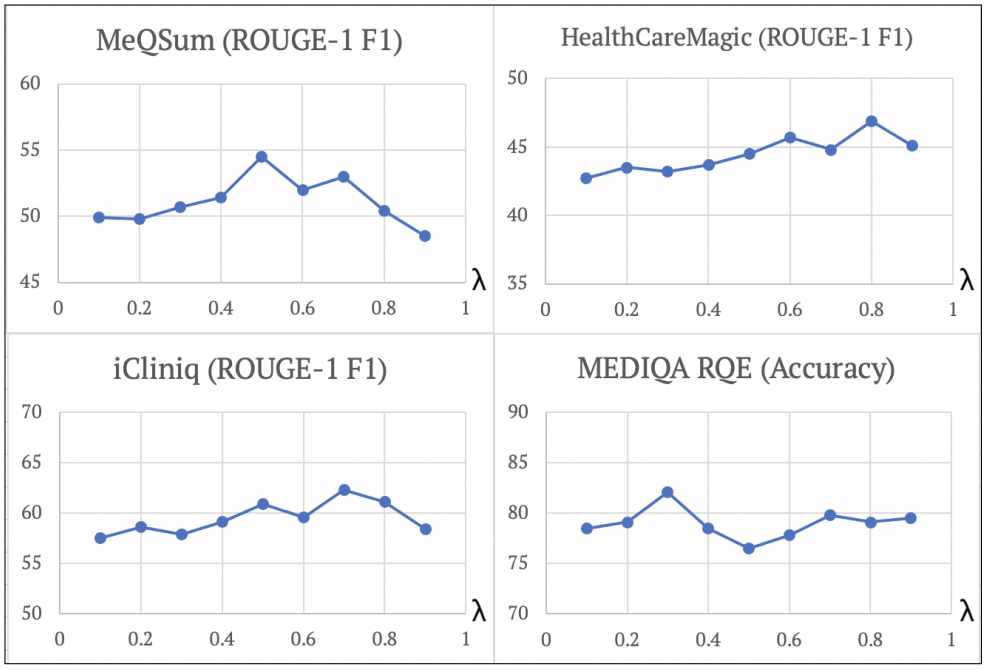


Figure 3.3: Dev set performance of multi-task learning as a function of the loss hyperparameter λ. The closer λ is to 0, the more the loss focuses on the RQE objective, and vice-versa for the question summarization objective.

dropout is 0.1. We set the γ hyperparameter to 1×10^{-7}.

3.4.3 Balancing between the Objectives

Our loss function as defined in equation 3.1 has a hyperparameter λ to balance between the question summarization objective and the RQE objective. We run experiments where λ varies from 0.1 to 0.9 in 0.1 increments. The results are in Figure 3.3. The best λ values are 0.5 for MeQSum, 0.7 for iCliniq, 0.8 for HealthCareMagic and 0.3 for MEDIQA RQE. For the question summarization datasets, we notice that the smaller the dataset, the more it benefits from data-augmented MTL with RQE.

3.4.4 Ablation Studies

We perform two ablation studies to show the added value of our main novelties: our equivalence-inspired data augmentation algorithm and our gradually soft parameter-sharing algorithm.

Data Augmentation. We compare our data augmentation algorithm against the following alternative: instead of training using a synthetic dataset for the auxiliary task, we choose a separate, existing dataset for abstractive summarization or recognizing textual entailment. This follows the approach taken by most MTL models. For the question summarization task, we optimize the cross-entropy objective using the RTE dataset. For the RQE task, we optimize the summarization objective using the XSum dataset. For the sake of fair comparison, we use the simultaneous MTL objective and the same architecture. Results in Table 3.2 show a consistent increase in performance across all datasets when using our data augmentation method, suggesting that in-domain MTL is more efficient.

Comparing Parameter-Sharing Configurations. We compare our gradually soft parameter-sharing method with 3 other parameter-sharing configurations. For all configurations, we keep using our data augmentation method, and sharing encoder parameters entirely.

1. Hard-shared decoder: decoder parameters are shared using hard parameter-sharing.

2. Soft-shared decoder: we apply soft parameter-sharing on decoder parameters across all N layers using the following, unweighted loss term:

$$\mathcal{L}_{\mathrm{S}}(\theta) = \gamma * \sum_{n=1}^{N} \left\| \theta_{\mathrm{dec},n}^{\mathrm{sum}} - \theta_{\mathrm{dec},n}^{\mathrm{ent}} \right\|^2 \tag{3.3}$$

3. Task-specific decoder: we train two task-specific decoders.

Our ablation study results in Table 3.2 show that our gradually soft parameter-sharing method exceeds all 3 of the other parameter-sharing configurations in RQE accuracy, and in the sum of ROUGE F1 scores. These results show our proposed smoother parameter-sharing

Table 3.3: Test set results on the 3 question summarization datasets.

DATASET	MeQSum			HealthCareMagic			iCliniq		
METRIC	R1	R2	RL	R1	R2	RL	R1	R2	RL
BASELINES									
Seq2seq Attentional Model [Nallapati et al., 2016]	24.8	13.8	24.3	-	-	-	-	-	-
Pointer-Generator Networks (PG) [See et al., 2017]	35.8	20.2	34.8	-	-	-	-	-	-
PG + Data Augmentation [Ben Abacha and Demner-Fushman, 2019b]	44.2	27.6	42.8	-	-	-	-	-	-
PG + Coverage Loss [See et al., 2017]	39.6	23.1	38.5	-	-	-	-	-	-
PG + Coverage Loss + Data Augmentation [Ben Abacha and Demner-Fushman, 2019b]	41.8	24.8	40.5	-	-	-	-	-	-
MODELS USING BART									
BART [Lewis et al., 2019]	45.7	26.8	40.8	44.5	22.3	39.7	48.7	28.0	43.5
BART + Entailment Generation + MTL of Pasunuru et al. [2017]	46.5	27.7	42.3	42.2	20.6	38.1	49.6	29.3	43.8
BART + Entailment Generation & Question Generation + MTL of Guo et al. [2018]	47.2	28.1	42.0	44.7	23.5	41.9	51.4	32.3	46.5
BART + Recognizing Question Entailment + Gradually Soft MTL + Data Augmentation (Ours)	49.2	29.5	44.8	45.9	24.3	42.9	54.2	36.9	49.1

Table 3.4: Human Evaluation results on 120 samples from the question summarization datasets. The percentages indicate the added value of our method.

DATASETS	Fluency	Coherence	Informative	Correct
MeQSum	+11.25%	+2.50%	+7.50%	0%
HealthCareMagic	+6.25%	-2.50%	+12.50%	+1.25%
iCliniq	+2.50%	0%	+3.75%	+5.00%

transition between encoder and decoder layers brings about higher performance.

3.4.5 Results and Discussion

Summarization Results

Baselines. We consider three main baselines. The first one is BART [Lewis et al., 2019], where we only train on the summarization task. The second baseline trains BART on the same MTL settings as Pasunuru et al. [2017], using alternative training with entailment generation on the Stanford Natural Language Inference (SNLI) corpus [Bowman et al., 2015] and having a

shared decoder and task-specific encoders. The third baseline trains BART on the same MTL settings as Guo et al. [2018], where, on top of the entailment generation task, we add the question generation task using the Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al., 2016], and all parameters are soft-shared, except for the task-specific first encoder layer and last decoder layer.

In addition, we also report the baselines assessed by Ben Abacha and Demner-Fushman [2019b] for MeQSum. For data augmentation, they use semantically-selected relevant question pairs from the Quora Question Pairs dataset [Iyer et al., 2017]. Their results show that coverage loss [See et al., 2017] diminishes the added value of data augmentation in pointer-generator networks. Our summarization-only BART baseline exceeds all of the reported MeQSum baselines in ROUGE-1 F1.

Summarization Results. We report our summarization results in Table 3.3. Compared to the single-task BART baseline, our gradually soft multi-task and data-augmented method performs better across all three ROUGE metrics, and achieves increases ranging from 1.4 to 5.5 points in ROUGE-1 F1. This differences shows that our method is consistently more efficient compared to training only on summarization.

The other two MTL baselines are generally performing better than the single-task BART baseline, except for the larger HealthCareMagic dataset. We observe that the different parameter-sharing configurations and tasks used in the MTL baselines are scoring about 1 to 4 points below our method in terms of ROUGE-1 F1 scores. This shows that our choice of tasks, simultaneous MTL loss, data augmentation and gradually soft parameter-sharing method work consistently better than existing MTL methods.

Human Evaluation. Given that ROUGE is notoriously unreliable, we hire 2 annotators to judge 120 randomly selected summaries from the summarization test sets, generated from the single-task BART baseline and our own method in Table 3.3. We ask the annotators to judge the Fluency, Coherence, Informativeness and Correctness of each generated summary, using

Best-Worst scaling, with the possibility of ranking both summaries equally. The annotators are presented with 2 generated summaries, in a randomized order at each evaluation, such that they cannot identify which method generated which summary.

Our human evaluation results are in Table 3.4. Scores generally favor our method, more strongly so in the abstractive datasets – HealthCareMagic and MeQSum. However, we note an increase in correctness for the more extractive iCliniq dataset. On average, our gradually soft multi-task and data-augmented method outputs summarized questions that are more fluent and more informative than the single-task BART baseline.

RQE Results and Discussion

Baselines. We compare our method to three baselines. The first one trains a single-task BART on RQE, with a classification head pre-trained on RTE. The second baseline is a feature-based SVM from Ben Abacha and Demner-Fushman [2016] who introduced the MEDIQA RQE dataset. The third baseline [Zhou et al., 2019] is an adversarial MTL method combining medical question answering and RQE. The architecture consists of a shared transformer encoder using BioBERT embeddings [Lee et al., 2020], separate classification heads for RQE and medical QA, and a task discriminator for adversarial training. A separate dataset is used for medical QA [Ben Abacha et al., 2019].

RQE Results. We show our RQE results in Table 3.5. We see a 12% increase on the test set compared to optimizing only on the RQE objective, and 10% increase. Without a separate dataset or embeddings trained on large-scale biomedical data, our method is able to exceed the performance of Zhou et al. [2019] by 0.7%. This confirms the strength of our method, and shows our method can increase performance in both RQE and Question Summarization in the medical domain.

Table 3.5: Accuracy results on MEDIQA RQE test set.

METHOD	Accuracy
BART [Lewis et al., 2019]	52.1%
Feature-based SVM [Ben Abacha and Demner-Fushman, 2016]	54.1%
BioBERT + Adversarial MTL with Medical QA [Zhou et al., 2019]	63.6%
BART + Summarization + Gradually Soft MTL + Data Aug. (Ours)	64.3%

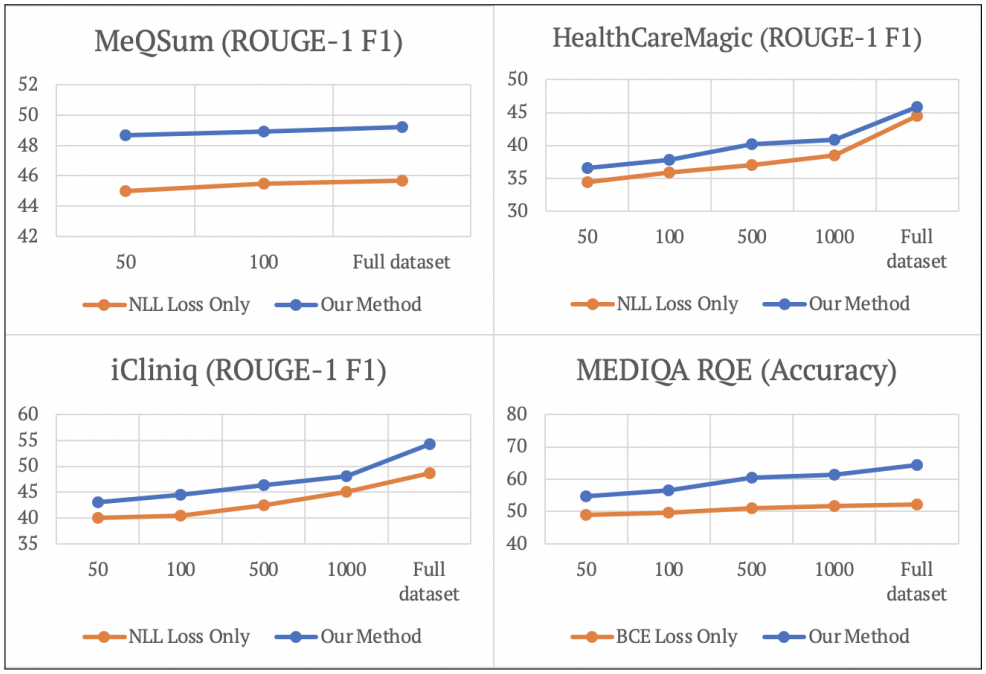


Figure 3.4: Test set 4-run average performance of our method compared to single-task BART in low-resource settings. Full dataset results are shown for comparison.

3.4.6 Performance in low-resource settings

We compare our gradually soft MTL and data-augmented method with the single-task BART baseline on four low-resource settings. For each dataset, we limit the training data to a subset of 50, 100, 500 or 1000 datapoints, and keep the same training settings. To avoid selection bias, we select four random and distinct subsets per low-resource setting, and show average ROUGE-1 F1 scores in Figure 3.4.

The results show that our approach is able to perform much better in low-resource settings. We notice in particular that, on all 4 datasets, the scores of the single-task BART baseline for 100 and 1000 datapoints are lower than or roughly equal to the scores of our method for a training subset of half the size (50 and 500 datapoints respectively). This suggests that our method's performance increase is not only related to additional datapoints, but also its gradually soft MTL setting.

3.5 Conclusions

We propose a novel multi-task learning approach for medical question understanding. Our approach trains on the tasks of RQE and question summarization in a simultaneous, weighted MTL loss function, where we add a loss term to constrain the decoder layers to be close, and we loosen the constraint gradually as we move higher up the layers. We show using the definitions of both tasks in the medical domain that we can augment datasets, such that we only need one dataset for MTL. Our two ablation studies show that our gradually soft parameter-sharing and our data augmentation algorithm each increase performance individually. We compare our method to single-task learning and existing MTL work, and show improvements across 3 medical question summarization datasets and 1 medical RQE dataset. Finally, we test our approach under low-resource settings: we find that it is able to efficiently leverage small quantities of data, and that these performance increases do not only depend on additional data from augmentation.

Acknowledgements

This chapter is a reformatted version of the material as it appears in "A Gradually Soft Multi-Task and Data-Augmented Approach to Medical Question Understanding," Khalil Mrini, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, and Ndapa Nakashole [Mrini et al., 2021c]. The material has been submitted, accepted and published at the 59th Annual Meeting of the Association for Computational Linguistics, ACL 2021. The dissertation author was the primary investigator and first author of this paper.

We gratefully acknowledge the award from NIH/NIA grant R56AG067393. Khalil Mrini is additionally supported by Adobe Research Unrestricted Gifts. This work is part of the VOLI project [Mrini et al., 2021a, Johnson et al., 2020]. We thank Naba Rizvi for the annotation work.

Chapter 4

Question Understanding and Answering

In the two previous chapters, we study the problem of selecting relevant answers, and the problem of understanding the intent of users when they write long questions. In this chapter, we bring those two problems together: we propose a system that understands a long user question, in order to select relevant answers from a knowledge base. We discover that question understanding and answering is not just a matter of combining tasks, as it is a problem with its own challenges and limitations.

4.1 Introduction

Motivation. Users of medical question answering systems often write long questions, called Consumer Health Questions (CHQs). Several aspects of CHQs hinder the capacity of current question answering (QA) systems to process them: long medical questions may contain peripheral information like patient history [Roberts and Demner-Fushman, 2016] that are not necessary to retrieve relevant answers. Consumer health questions may also use a distinct vocabulary from the one used by medical providers to describe the same health concepts [Ben Abacha and Demner-Fushman, 2019b].

A growing number of approaches attempt to enhance the processing of consumer health

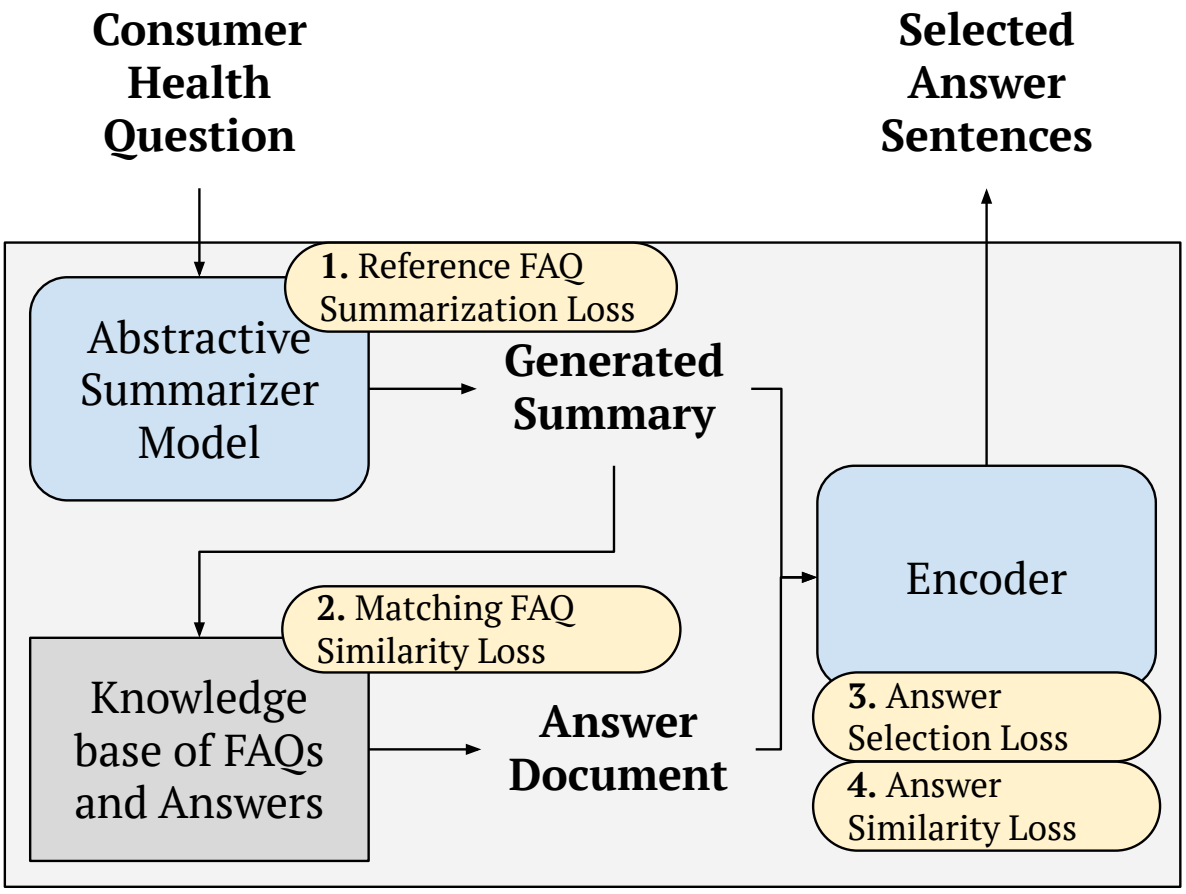


Figure 4.1: Overview of our proposed Consumer Health Question Understanding and Answering model. The input is a user question, called *Consumer Heath Question* (CHQ). The goal is to match the CHQ to relevant answer sentences associated with a *Frequently Asked Question* (FAQ) from a medical knowledge base.

questions – or medical question understanding. These approaches include query relaxation [Ben Abacha and Zweigenbaum, 2015, Lei et al., 2020], question entailment [Ben Abacha and Demner-Fushman, 2016, 2019a, Agrawal et al., 2019], question summarization [Ben Abacha and Demner-Fushman, 2019b], and question similarity [Ben Abacha and Demner-Fushman, 2017, Yan and Li, 2018].

However, the above medical question understanding approaches stop short of retrieving answers after processing consumer health questions. The Medical Question Answering Task at TREC 2017 LiveQA [Ben Abacha et al., 2017] attempts to fill the gap by proposing the task of Consumer Health Question Answering. The goal is to retrieve relevant answers obtained using online search for the corresponding CHQ. As part of their participation in this task, Yang et al. [2017] find that online search engine queries introduce noise in performance, and that even collected and curated medical knowledge available offline can fare better.

Contributions. To enable the use of a curated medical knowledge base for answering long user questions, we introduce a novel, knowledge-grounded and semantically self-supervised system for Consumer Health Question Understanding and Answering (CHQUA). We tackle a challenging aspect of CHQUA: providing answers when no relevance labels are available. Our contributions are as follows:

(1) We propose an end-to-end pipeline, as shown in Figure 4.1, that takes as input a consumer health question, and trains a summarizer model to generate a short, formally worded question. We optimize a summarization training objective using the medical question summarization datasets.

(2) The medical knowledge base we use is separate from the question summarization datasets, and therefore we have no labels to indicate which knowledge base question matches a given consumer health question. We design a novel, semantically-guided self-supervised loss function to ground the generated summary with knowledge base FAQs, using semantic similarity as proxy to question matching. The Matching FAQ similarity loss helps the encoder pick the most

semantically similar knowledge base question.

(3) The large medical knowledge base we use has no answer sentence relevance labels. We adapt to this scenario by designing two complementary self-supervised losses on the same encoder, and by considering semantic similarity as a proxy to relevance. The Answer Similarity loss pushes the model to distinguish between relevant and irrelevant answer sentences, whereas the Answer Selection loss works in a complementary way to push the model to select a given number of sentences.

Finally, we conduct an evaluation to compare the relevance of our system with two strong baselines of retrieval-based question answering. We ask evaluators to ask their own questions, and then perform a blind evaluation of the retrieved answers by each system. Seven evaluators find that our system retrieves more relevant answers compared to the two baselines, while achieving significantly faster processing speeds. We also find that the self-supervised losses help achieve better scores in ROUGE and human evaluation metrics. However, we find that the task remains challenging, with room for improvement. We release our code, model, and matched datasets to encourage further research in consumer health question understanding and answering.

4.2 Related Work

Consumer Health Question Answering. Ben Abacha et al. [2017] introduce the Medical QA shared task at TREC 2017 LiveQA, where the goal is to develop a consumer health question answering system. The training data is comprised of question-answer pairs. The questions are informally worded CHQs received by the U.S. National Library of Medicine (NLM). The answers are formally worded and come from websites of the U.S. National Institutes of Health or manually collected by librarians. The evaluation scores are given by humans, using a test set of CHQs and reference answers.

The team of Wang and Nyberg [2017] scores the highest evaluation score by far. Their

approach starts first by searching for the full CHQ on Bing Web Search and Yahoo! Answers to find relevant answers. Then, they retrieve relevant answer sentences using an ensemble model comprised of non-parametric IR techniques, a question paraphrase identification model, and a question-answer relevance prediction model.

Many participating teams adopt a question matching approach, and train their models on question similarity datasets like the Quora question pair dataset [Iyer et al., 2017], or other datasets collected from community question answering websites.

In the MEDIQA 2019 Shared task, Ben Abacha and Demner-Fushman [2019b] introduce a differently defined consumer health question answering task. Here, the goal is to rank a given list of answers according to their relevance with regard to a CHQ. He et al. [2020] introduce a new disease knowledge infusion training procedure for BERT [Devlin et al., 2019b] that scores well in this task.

Medical Question Answering. Medical QA approaches include translating questions to SPARQL queries [Ben Abacha and Zweigenbaum, 2012], semantic similarity between questions and candidate answers [Hao et al., 2019], knowledge representations [Terol et al., 2007, Goodwin and Harabagiu, 2017], ranking candidate answers [Ben Abacha et al., 2017, 2019], and multi-answer summarization [Ben Abacha et al., 2021].

There is a variety of definitions for the task of medical QA and related sub-tasks in the literature. Hao et al. [2019] define medical QA as the task of finding the correct answer from a set of candidates and a body of evidence documents. They propose to work on two datasets: the National Medical Licensing Examination of China (NMLEC) [Shen et al., 2020], and Clinical Diagnosis based on Electronic Medical Records (CD-EMR), where the goal is to predict the correct diagnosis based on patient history.

Sharma et al. [2018] propose to tackle three kinds of medical questions found in the BioASQ challenge [Balikas et al., 2015]: factoid questions where answers are single entities, list-type questions where answers are a set of entities, and yes/no questions.

Retrieval-based Question Answering. Recent methods for retrieval-based QA systems use contextual text embeddings to evaluate a candidate answer's relevance to a given question.

Tay et al. [2018b] propose to use Multi-Cast Attention Networks (MCAN), a new attention mechanism, to model question-answer pairs. The resulting model predicts a relevance score. At the time of publication, MCANs scored a new state of the art on the TrecQA benchmark dataset [Wang et al., 2007].

Mrini et al. [2021e] introduce a recursive, tree-structured model that models sentences according to their syntactic tree. Their results show that tree structure sets a new state of the art in conventional, formally worded QA benchmarks like TrecQA and WikiQA [Yang et al., 2015b], but does not fare well in informally worded, user-written datasets.

Karpukhin et al. [2020] introduce Dense Passage Retrieval (DPR): a dual-encoder based on BERT [Devlin et al., 2019b], that predicts relevance scores of passages with regard to a question. DPR encoders are trained on the relevance of passages from datasets containing such labels, using a supervised negative log-likelihood loss based on the semantic similarity of questions and relevant passages.

Mao et al. [2021] modify the *query* part of retrieval-based QA: they propose to use language models to generate context for queries. They then feed the extended queries to retrieval systems, such as DPR or BM-25.

4.3 Problem Definition

We define knowledge-grounded Consumer Health Question Understanding and Answering (CHQUA) as the problem of retrieving a fixed number of answer sentences from a medical knowledge base that are the most relevant given a long and informal user question – called a Consumer Health Question (CHQ). There are three steps in CHQUA: question summarization, matching the summarized user question with a relevant FAQ from the knowledge base, and

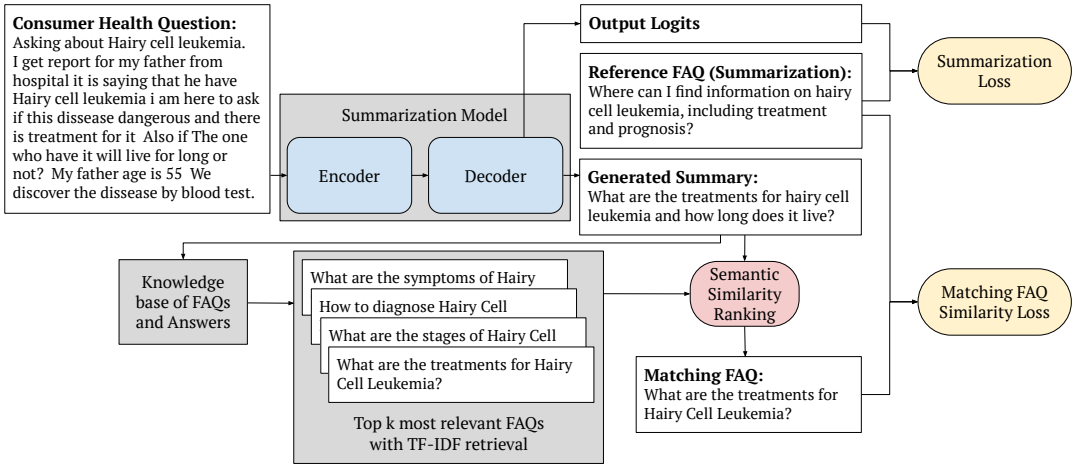


Figure 4.2: The Consumer Health Question (user question) is first summarized, and we then retrieve a relevant question from the knowledge base using the generated summary. The **top** half of the figure illustrates the first step: question understanding through summarization (§4.4.1). The **bottom** half of the figure illustrates the second step: question matching through self-supervised knowledge grounding (§4.4.2).

retrieval of the relevant answer sentences from the corresponding answer document.

Knowledge-grounded CHQUA is comprised of three elements used for training. First, the CHQ is the input of the task. Second, the Reference FAQ (Frequently Asked Question) is the golden or expert-written summary corresponding to the CHQ. Whereas the CHQ is a long and informally worded question, the reference FAQ is the corresponding short, one-sentence, formally worded question. At inference time, the reference FAQ is not available, and we will therefore use a summary generated by the model. Third, the medical knowledge base is comprised of FAQs, where each FAQ has a corresponding answer document with at least one sentence. FAQs in the knowledge base are also short, one-sentence, formally worded questions.

The goal of knowledge-grounded CHQUA is to find a set $\mathcal{R}$ of n relevant answer sentences, from a document comprised of answer sentences $\mathcal{A}_i$, such that $\mathcal{A}_i$ corresponds to question q_i from the knowledge base. We call q_i the retrieved or matching FAQ, such that q_i is the most similar question to the user's summarized question q_u:

$$\mathrm{q}_i = \arg\max_{\mathrm{q} \in \mathcal{Q}} f(\mathrm{q}, \mathrm{q}_\mathrm{u}) \tag{4.1}$$

where $\mathcal{Q}$ is the set of questions (FAQs) in the knowledge base, and f is a given similarity scoring function. q_u is the reference FAQ (during training) or a generated summary (during inference).

We find the set $\mathcal{R}$ of n relevant answer sentences such that it maximizes the relevance score with the user's summarized question q_u:

$$\mathcal{R} = \arg\max_{\mathcal{R}' \subset \mathcal{A}_i} \sum_{\mathrm{a} \in \mathcal{R}'} g(\mathrm{a}, \mathrm{q}_\mathrm{u}) \tag{4.2}$$

where a is an answer sentence, and g is a given relevance scoring function.

4.4 Our Pipeline

Our proposed pipeline for Consumer Health Question Understanding and Answering has three main components.

In the first step, our approach learns to *understand* the intent of user questions (CHQs) by summarizing them. We use an encoder-decoder-based summarization model for this step.

The second step is question matching, or the retrieval of the relevant FAQ from the knowledge base: we *ground* the generated summary to a medical knowledge base of FAQs and corresponding answer documents. As there are no question matching labels, we consider semantic similarity as a proxy to question matching, and we optimize a self-supervised similarity loss.

The third step is the retrieval of the relevant answer sentences: our model learns to *select* the top-k most relevant answer sentences from the matching answer document. To achieve this task in the absence of answer relevance labels, we consider semantic similarity as a proxy for relevance, and we optimize two novel, semantically-guided, and self-supervised loss functions. The first pushes the model to discriminate between relevant and irrelevant sentences, and the

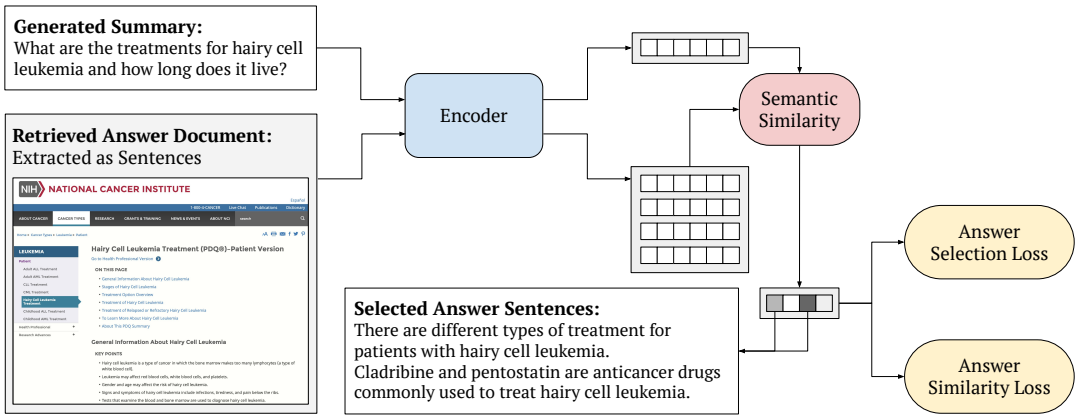


Figure 4.3: Illustration of the third step of our pipeline: answer retrieval through self-supervised similarity and selection losses (§4.4.3). Following the same example as Figure 4.2, our model encodes sentences from the retrieved answer document from the knowledge base, and compares them to the FAQ generated by the summarization model. We use the encoder of the summarization model to embed sentences.

other pushes the model to consider only a fixed number of sentences as relevant.

We show an overview of the model and learning objectives in Figure 4.1. The entire pipeline is trained together, as the summarizer encoder is re-used to encode the questions and answer sentences.

4.4.1 Question Understanding through Summarization

Our work aims to flip the burden of question understanding on the question answering model. Instead of asking the user to shorten or reformulate their question, we train an encoder-decoder abstractive summarizer to shorten user questions. Figure 4.2 illustrates this part of the model.

At training time, we input a Consumer Health Question (CHQ) to the summarization model. The reference Frequently Asked Question (FAQ) is the corresponding shorter and formal question. Given a CHQ embedding $\mathbf{x}$ and the corresponding reference FAQ embedding $\mathbf{y_{ref}}$, the summarization loss is defined as the following negative log-likelihood objective:

$$\mathcal{L}_{\text{sum}} = -\log p(\mathbf{y}_{\text{ref}}|\mathbf{x};\theta) \tag{4.3}$$

4.4.2 Question Matching through Self-Supervised Knowledge Grounding

In the next step, we match the summarized user question with the most relevant FAQ from the medical knowledge base. We use semantic similarity as a proxy for question matching, in the absence of such labels.

The knowledge-grounding process is comprised of two steps. First, we use TF-IDF-weighted bag-of-word and n-gram vectors to get the top k most relevant FAQs from the knowledge base. This first step acts as a fast filter to extract a small subset of candidate FAQs. Our retrieval approach follows the retrieval methods commonly used in question answering systems [Chen et al., 2017a, Dinan et al., 2018]. Dinan et al. [2018] note that the retriever is a potentially learnable part of the model. In our case, using TF-IDF retrieval is computationally optimal and scalable given a large knowledge base with thousands of FAQs. We use a TF-IDF embedder fitted on all the FAQs of the knowledge base, as well as reference FAQs from the training set of the question summarization dataset.

The second step of knowledge-grounding is to rank the top k FAQs using semantic similarity. To get semantic embeddings of the generated summary and the corresponding top k most relevant FAQs from the knowledge base, we use the encoder of the summarization model. We take inspiration from the precision formula of BERTSCORE [Zhang et al., 2019], and compute the weighted semantic similarity score as follows:

$$\text{Sim}(\text{q}_\text{u}, \text{q}_i) = \sum_{w \in \mathcal{W}_\text{u}} \max_{w' \in \mathcal{W}_i} \frac{\text{idf}(w) \cdot \text{CosSim}(\mathbf{x}_w, \mathbf{x}_{w'})}{\sum_{w'' \in \mathcal{W}_\text{u}} \text{idf}(w'')} \tag{4.4}$$

where q_u is the reference FAQ (during training) or the generated summary (during inference), q_i is the i-th question from the top k most relevant FAQs, $\mathcal{W}_\text{u}$ and $\mathcal{W}_i$ are the corresponding sets of words, CosSim is the cosine similarity function, and $\text{idf}(w)$ is the inverse document frequency of

the word *w*.

The matching FAQ is the knowledge base FAQ with the highest similarity score with q_u, as shown in the example in Figure 4.2. During training, the summarization model may produce low-quality or degenerate FAQs. For this reason, at training time, we choose to use the reference FAQ instead to compute the semantic similarity scores and find the matching FAQ. At test time, we only use the generated summary.

Since we are using different datasets for the question summarization and for the knowledge base, we have to reconcile the questions from the knowledge base and the reference questions. We propose to force the model to learn a representation space that does not distinguish between the reference FAQ and the most similar knowledge base FAQ. To accomplish this, we compute the matching FAQ similarity loss. Given the embedding of a summarization reference FAQ q_{sum} and the embedding of a matching FAQ q_{mat}, the matching FAQ similarity loss is defined as:

$$\mathcal{L}_{mat} = 1 - \text{ReLU}\left(\text{Sim}\left(q_{sum}, q_{mat}; \theta\right)\right) \tag{4.5}$$

4.4.3 Answer Retrieval through Self-Supervised Similarity and Selection Losses

After summarizing the user question and retrieving a relevant FAQ from the knowledge base, the next step is to retrieve relevant sentences from the corresponding answer document. In our setting, we need to retrieve a fixed number of sentences relevant to the user question. However, we have no labels for the answer sentences indicating relevance to the user question. We propose two complementary self-supervised learning objectives, that use semantic similarity as a proxy to relevance scoring, and satisfy the constraint of selecting a fixed number of answer sentences.

We show an overview of our answer retrieval approach in Figure 4.3. In the example of the figure, we show for simplicity a relatively short answer document with four sentences, from

which the model chooses the two most relevant ones. In practice, there are close to ten sentences in answer documents.

We compute semantic similarity scores between the generated summary (for inference) or the reference FAQ (for training), and each of the sentences of the retrieved answer document. We obtain the semantic embeddings of each sentence using the encoder of the summarization model. We then compute semantic similarity scores as shown in equation 4.4. Cosine similarity scores have values in the $[-1;1]$ range. For a pair of sentences, a cosine similarity value closer to -1 means that the corresponding sentence embeddings are negatively correlated, or that the sentences have opposite meanings. A value closer to 0 means that the embeddings are not correlated, and that there is no particular semantic relation between the sentences. A value closer to 1 means that the sentence embeddings are positively correlated, and the sentences are close semantically. We consider that a sentence is relevant when the values are closer to 1, and irrelevant otherwise. For this reason, we apply a ReLU activation on the cosine similarity scores before feeding them to the loss functions.

We propose two learning objectives to achieve the self-supervised selection of relevant answer sentences. The semantic similarity loss pushes the model to increase its confidence in the relevance of answer sentences, whereas the answer selection loss pushes the model to select only a fixed number of sentences. The intuition for sharing the encoder with the summarization model, is that these two losses will enable the summarizer to absorb notions of relevance and semantic similarity.

Given the summarization reference FAQ $\mathrm{q}_{\mathrm{sum}}$ and the i-th sentence of the retrieved answer document a_i, we compute the ReLU-activated semantic similarity score as follows:

$$S(\mathrm{q}_{\mathrm{sum}}, \mathrm{a}_i; \theta) = \mathrm{ReLU}\left(\mathrm{Sim}\left(\mathrm{q}_{\mathrm{sum}}, \mathrm{a}_i; \theta\right)\right) \tag{4.6}$$

We then define the semantic similarity loss $\mathcal{L}_{\mathrm{sim}}$ and the answer selection loss $\mathcal{L}_{\mathrm{sel}}$ as follows:

$$\mathcal{L}_{\text{sim}} = \sum_{i=1}^{|\mathcal{A}|} S(\text{q}_{\text{sum}}, \text{a}_i; \theta) * (1 - S(\text{q}_{\text{sum}}, \text{a}_i; \theta)) \tag{4.7}$$

$$\mathcal{L}_{\text{sel}} = \left| \min(n, |\mathcal{A}|) - \sum_{i=1}^{|\mathcal{A}|} S(\text{q}_{\text{sum}}, \text{a}_i; \theta) \right| \tag{4.8}$$

where $\mathcal{A}$ is the set of sentences in the retrieved answer document, and n is the fixed number of sentences to be retrieved.

The semantic similarity loss $\mathcal{L}_{\text{sim}}$ pushes the semantic similarity values to be either 1 (relevant) or 0 (irrelevant). In combination with $\mathcal{L}_{\text{sim}}$, the answer selection loss pushes the model to only select up to n sentences to have semantic similarity values close to 1. Our system then outputs the sentences with the highest semantic similarity values in the order in which they appear in the answer document. Therefore, the particular semantic similarity ranking of the relevant sentences does not matter – it only matters that relevant sentences have the n highest values.

Finally, the learning objective $\mathcal{L}$ is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{sum}} + \lambda * \mathcal{L}_{\text{mat}} + \gamma * (\mathcal{L}_{\text{sim}} + \mathcal{L}_{\text{sel}}) \tag{4.9}$$

where λ and γ are hyperparameters. We use only one weight for $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{sel}}$ as these two losses are complementary.

4.5 Experiments and Results

In this section, we evaluate our proposed pipeline for Consumer Health Question Understanding and Answering, and we propose to compare our proposed pipeline against two strong baselines. Seven medical experts judge the performance of our system and baselines by asking their own questions, and rating the relevance of the answers retrieved. Then, we analyze the results through the lens of summarization metrics, human evaluation, and computational speed.

Table 4.1: Statistics of the medical dataset splits.

DATASET SPLIT	TRAIN	DEV	TEST
MeQSum	405	50	50
HealthCareMagic	1,314	164	165

4.5.1 Datasets

We use one medical knowledge base, MedQuAD [Ben Abacha and Demner-Fushman, 2019a], and two medical question summarization datasets: MeQSum [Ben Abacha and Demner-Fushman, 2019b] and HealthCareMagic [Chen et al., 2020]. All datasets are in English. We show dataset statistics in Table 4.1.

MedQuAD is a large-scale Medical Question Answering Dataset. Ben Abacha and Demner-Fushman [2019a] collect trusted medical question-answer pairs by crawling them from 12 websites of the U.S. National Institutes of Health (NIH). Each web page contains information about a health-related topic, like a disease or a drug. The authors automatically collect the question-answer pairs by composing handcrafted patterns adapted to each website based on document structure and section titles. They manually evaluate 1,721 CHQs to come up with automatic wording patterns for each of 36 question types. Therefore, even though answers are curated and written by medical experts, questions are automatically formulated and may have some noise.

We collect the publicly available (e.g. not copyrighted) question-answer pairs from the MedQuAD dataset[1]. We then use the NLTK sentence tokenizer [Bird, 2006] to split answer documents into sentences. We get 16,423 questions and 157,592 answer sentences, making for an average of 9.6 answer sentences for each question.

MeQSum [Ben Abacha and Demner-Fushman, 2019b] is a medical question summarization dataset released by the U.S. National Institutes of Health (NIH). It contains 1,000 consumer health questions summarized into FAQ-style single-sentence questions by medical experts.

[1] https://github.com/abachaa/MedQuAD

HealthCareMagic is a medical dialogue dataset issued as part of the MedDialog dataset [Chen et al., 2020][2]. It is crawled from `HealthCareMagic.com`, an online healthcare service platform. This dataset includes first a formally worded, one-sentence question describing the intent of the patient question, followed by 2 long utterances: a CHQ from the patient that includes a description of the problem and a question, and then an answer from the doctor. To form a medical question summarization dataset, we consider the single-sentence descriptions as summaries of the patient's CHQ. We collect 226,405 question pairs.

4.5.2 Knowledge-based Filtering of Datasets

We conduct experiments for each of the two question summarization datasets, and we use MedQuAD as the underlying knowledge base in all experiments. For this reason, we decide to filter each of the question summarization datasets to reconcile their differences with MedQuAD.

We first fit a TF-IDF embedding model, similar to the one of [Dinan et al., 2018], on the reference FAQs of each question summarization dataset and the questions of MedQuAD. We then compute the dot products of the TF-IDF-weighted vectors for all possible pairs of summarization FAQs and MedQuAD questions. We assign a matching score $m(\mathrm{q}_{\mathrm{sum}})$ to each summarization reference FAQ:

$$m(\mathrm{q}_{\mathrm{sum}}) = \max_{\mathrm{q}' \in \mathrm{Q}_{\mathrm{MedQuAD}}} \mathrm{tfidf}(\mathrm{q}_{\mathrm{sum}}) \cdot \mathrm{tfidf}(\mathrm{q}') \tag{4.10}$$

We manually evaluate the matching scores for each summarization dataset to set a cutoff matching score of filtering. This way, we obtain question summarization datasets where reference FAQs have matches in the medical knowledge base. Finally, we perform a random and rough 80/10/10 split for the train/dev/test sets. The dataset statistics are in the main paper.

[2] `https://github.com/UCSD-AI4H/Medical-Dialogue-System`

4.5.3 Baselines

We propose the two following baselines in retrieval-based question answering: Dense Passage Retrieval (DPR) [Karpukhin et al., 2020], and Generation-Augmented Retrieval (GAR) [Mao et al., 2021]. We adapt these two baselines to our case, and adopt BART-based pre-trained encoders.

Similarly to our own pipeline, we create a two-stage retrieval to get answers. The first stage encodes questions from the knowledge base, and retrieves the question that is most relevant to the query. The second stage encodes the corresponding answer document, and retrieves the three sentences that are most relevant to the query.

For DPR, the query is simply the user question. For GAR, we need to generate a context to add to the user question: we choose to add the summary of the user question as the context. We train a BART encoder to summarize user question, using the question summarization datasets.

Whereas our system's retrieval encoder is trained on our proposed self-supervised objectives, the retrieval encoders of the baselines are trained on Wikipedia for the task of retrieval-based question answering.

4.5.4 Training Settings

We adopt the BART encoder-decoder model [Lewis et al., 2019], as it set a state of the art in abstractive summarization benchmarks. We train our model using the HuggingFace implementation [Wolf et al., 2020], on a learning rate of $2 \cdot 10^{-6}$. The question matching pool retrieved by TF-IDF is comprised of $k = 32$ knowledge base FAQs. Our answer selection loss $\mathcal{L}_{\text{sel}}$ is optimized to select up to $n = 3$ sentences. We use $\lambda = 0.01$ and $\gamma = 0.01$ as weights for the self-supervised losses. The BART encoder is used for embedding sentences for question matching and answer selection.

We train for 50 epochs for MeQSum, and 20 epochs for HealthCareMagic. Each training

epoch takes about 10 minutes for MeQSum, and about 35 minutes for HealthCareMagic. Inference takes 1 minute for the MeQSum test set and 3 minutes for the HealthCareMagic test set. The best checkpoint is selected based on the lowest loss value $\mathcal{L}$ on the dev set.

We use BART Large pre-trained on the CNN-Dailymail dataset, and each BART Large model contains 406 million parameters, as per the HuggingFace implementation.

Our OS is 16.04.1-Ubuntu/ x86 64. We used a single GPU for experiments with our system: one GeForce GTX 1080 Ti with 11 GB of memory. For the baselines, we use four 16GB GPUs, available as a single p3.8xlarge EC2 instance on Amazon Web Services.

4.5.5 Do we retrieve relevant answers?

Evaluation Strategy

We hire seven annotators: four of which are medical doctors, and the remaining three hold degrees related to healthcare or immunology.

We ask the evaluators to first write user questions, and then evaluate the answers retrieved by our system and the two existing systems. Given that our medical knowledge base has limited questions, we ask the evaluators to limit their questions to the topics covered by the nine sources from which the knowledge base was extracted. The sources of questions and answers in MedQuAD are as follows:

- National Cancer Institute
- Genetic and Rare Diseases Information Center: various aspects of genetic/rare diseases
- Genetics Home Reference (GHR): consumer-oriented information about the effects of genetic variation on human health
- MedlinePlus Health Topics: information on symptoms, causes, treatment and prevention for diseases, health conditions, and wellness issues

- National Institute of Diabetes and Digestive and Kidney Diseases
- National Institute of Neurological Disorders and Stroke: neurological and stroke-related diseases
- NIHSeniorHealth: health and wellness information for older adults
- National Heart, Lung, and Blood Institute (NHLBI): diseases, tests, procedures, and other relevant topics on disorders of heart, lung, blood, and sleep
- Centers for Disease Control and Prevention (CDC)

Then, we ask the evaluators to rate the relevance of the answers retrieved by each system independently, according to the following criteria:

- Score of 1/5: The system's answer is completely irrelevant to the question, and does not even contain any concept related to the question.
- Score of 2/5: The system's answer mentions notions that are related to the question, but does not contain a word or concept mentioned in the question.
- Score of 3/5: The system's answer mentions one or more words or concepts from the question, but does not actually answer the question.
- Score of 4/5: The system's answer partially answers the question, mentions one or more words or concepts from the question, but does not fully answer the question.
- Score of 5/5: The system's answer fully answers the question.

Each of the seven annotators wrote 20 questions, and each question gets three answers (one per system). We assign three annotators to the models trained on MeQSum, and four to the models trained on HealthCareMagic. The annotators rate answers only for the questions that they wrote themselves.

Table 4.2: Evaluation of the relevance (out of 5) of answers retrieved by our proposed system and two strong baselines for questions asked by seven evaluators. The systems trained on MeQSum are evaluated on 60 questions by 3 evaluators, and the ones trained on the larger HealthCareMagic dataset are evaluated on 80 questions by 4 evaluators. The column on the right shows the number of seconds it takes for a loaded system to retrieve the answer to a query.

SYSTEM	MeQSum	HealthCareMagic	Time/Query
DPR [Karpukhin et al., 2020]	1.42	1.73	47 seconds
GAR [Mao et al., 2021]	1.40	1.64	48 seconds
Ours	2.13	2.35	2 seconds

Results and Discussion

We show the results of the evaluations in Table 4.2. The first three columns show the averages of relevance scores that were given by annotators for all systems.

The results show that the evaluators have preferred our system's answers over the answers retrieved by the two baselines. Our system gets relevance scores that are 0.6 to 0.7 points higher, out of 5 on the relevance scale.

The two baselines seem to perform similarly to each other. This is likely due to the fact that the main difference between them is that the query is generation-augmented for GAR, whereas the query is simply the user question for DPR.

Overall, the relevance scores are on the lower side, as no system exceeds an average score of 2.5/5. This shows that consumer health question answering and understanding is a challenging task, especially since there are no labels to indicate whether an answer is relevant to a particular question, or which FAQ matches the user's intent.

In addition, the challenges of the task are also due to the limitations of the knowledge base. Some annotators noted that the retrieved answers were often not appropriate, or close to the topic but not answering the question. This is due to the fact that MedQuAD does not cover all possible illnesses and medical conditions that the users could ask about. Whereas a larger database would potentially solve coverage problems, it could be at the expense of the quality or verifiability of the answers. The MedQuAD dataset is at times noisy, and contains generic sentences that may not

Table 4.3: Question Understanding evaluation: summarization results on test set (reference FAQs). The R1, R2 and RL metrics refer to the F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L.

DATASET	MeQSum			HealthCareMagic		
METRIC	R1	R2	RL	R1	R2	RL
GAR [Mao et al., 2021]	45.72	30.43	42.02	31.04	13.68	27.90
Ours	46.74	30.10	42.81	33.13	14.71	30.18

answer any question, or generic templates related to percentages of symptoms and how frequent they are.

4.5.6 Computational Speed

We run our system on a single 11GB GPU, whereas the two baselines are each run on four 16GB GPUs. We show the average duration required to retrieve answers for a single query in the right column of Table 4.2.

This is done at the beginning when loading the models, but the query similarity computation is done at each run, thereby lengthening the processing time.

4.5.7 Analysis of Question Understanding

An additional way that our system outperforms the two baselines could be through summarization. We evaluate the summarization of consumer health questions using the ROUGE metric [Lin, 2004]. Our GAR baseline uses a BART model trained on the summarization loss only. We show the results in Table 4.3. We notice that sharing encoder parameters between the summarization loss and our proposed self-supervised losses generally increases ROUGE F1 scores across both datasets. For HealthCareMagic, score increases exceed 2 points in ROUGE-1 and ROUGE-L.

Given that ROUGE is notoriously unreliable, we hire two additional annotators on Upwork who are healthcare workers to judge the fluency, coherence, informativeness and correctness of generated summaries. We define these criteria for the two healthcare worker annotators as

Table 4.4: Question Understanding evaluation: blind evaluation by 2 annotators of the generated summaries for the test set CHQs. A "Win" evaluation means that our model generates a better summary than the baseline summarizer.

CRITERIA	Fluency			Coherence			Informativeness			Correctness		
EVALUATION	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie
MeQSum	11	5	28	10	6	28	12	3	29	12	4	28
HealthCareMagic	45	17	42	44	19	41	46	18	40	44	18	42

follows:

- Fluency: which generated FAQ is more grammatically correct, and easier to read and to understand?
- Coherence: which generated FAQ is better structured and more organized?
- Informativeness: which generated FAQ captures the most out of the concern of the patient who wrote the CHQ?
- Correctness: which generated FAQ is more factually correct given the CHQ?

We show the annotators the consumer health question (source text), the reference FAQ (target text) and two generated summaries. The annotators do not know which system generated which summary. We show the evaluation scores in Table 4.4. We remove repetitions of reference FAQs in the test sets put up for evaluation. The results confirm that our self-supervised losses increase the quality of generated summaries. Summaries generated with our model score more wins more often than losses on all four metrics, and score more wins than ties with the summarization-only baseline for HealthCareMagic.

4.6 Conclusions

We introduce an end-to-end pipeline for knowledge-grounded consumer health question answering and understanding (CHQUA). Our challenge is that we have no labels for question

matching or answer relevance. We propose to use semantic similarity as a proxy for those labels, and we design three novel self-supervised losses: one works to match the user's summarized question to a knowledge base question, and the other two losses work complementarily to teach our model to select a fixed number of relevant answer sentences.

We compare our proposed system against two strong baselines of retrieval-based question answering. We hire seven medical experts to ask their questions, and they find that our system provides more relevant answers. Our system also achieves processing times that are more than 20 times faster. Finally, we find that our proposed self-supervised losses enable the summarizer model to achieve higher scores in ROUGE and human evaluation metrics, compared to a summarization-only baseline. However, we find that this task remains challenging and that there is still room for improvement.

Acknowledgements

This chapter is a reformatted version of the material as it appears in "Medical Question Understanding and Answering with Knowledge Grounding and Semantic Self-Supervision," Khalil Mrini, Harpreet Singh, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas, and Ndapa Nakashole. The material has been submitted to the 29th International Conference on Computational Linguistics, COLING 2022. The dissertation author was the primary investigator and first author of this paper.

Khalil Mrini is supported by an Amazon Research Award won with his AWS AI Proposal "*Learning Representations for Voice-Based Conversational Agents for Older Adults*" with his advisor Ndapandula Nakashole. This work is part of the VOLI project [Mrini et al., 2021a, Johnson et al., 2020]. We thank Mike Hogarth, Allison Moore, and Nadir Weibel for fruitful discussions.

Ethical Considerations

Our model is for medical question answering, but should be used with caution as it does not claim to provide medical advice. Potential users of our system should be warned to not blindly trust the answers given to their medical questions. Potential users should always consult their physician for medical advice.

Each of our annotators spent between two and four hours on the task we gave them. Each annotator was compensated fairly (100 USD) for their work. We answered all of the annotators' questions about the task before they started. Hiring platform Upwork guarantees the payment, fair treatment and informed consent of our nine hired annotators through a mutually agreed-upon contract. The platform fee for Upwork was paid by us, and not deducted from the compensation of the annotators.

Chapter 5

Text Understanding as Entity Linking

The two previous chapters deal with understanding the intent of long user questions, in order to improve recall in answer selection. However, not all user-written text is a question. This chapter focuses on understanding what entities users are referencing in their utterances. Similarly to the earlier chapters, our study focuses on bridging the gap between informal, user-written vocabulary and formal, technical vocabulary.

5.1 Introduction

Entity linking [Zhang et al., 2010, Han et al., 2011] is the task of linking entity mentions in a text document to concepts in a knowledge base. It is a basic building block used in many NLP applications, such as question answering [Yu et al., 2017, Dubey et al., 2018, Shah et al., 2019], word sense disambiguation [Raganato et al., 2017, Uslu et al., 2018], text classification [Basile et al., 2015, Scharpf et al., 2021], and social media analysis [Liu et al., 2013, Yamada et al., 2015].

Early definitions decompose the task of entity linking (EL) into two subtasks: Mention Detection (MD) and Entity Disambiguation (ED). Many statistical and LSTM-based methods propose to cast EL as a two-step problem, and optimize for both MD and ED [Guo et al., 2013,

Source Text

SOCCER - Japan Get Lucky Win, China In Surprise Defeat. Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday. But China saw their luck desert them [...]

GENRE (De Cao et al., 2021)

SOCCER - Japan Get Lucky Win, China national football team In Surprise Defeat. Japan national football team began the defence of their AFC Asian Cup title with a lucky 2-1 win against Syria national footballer team in a Group C championship match on Friday. But China Chinese Super League [...]

Our Multi-Task Model

SOCCER - Japan national football team Get Lucky Win, China national football team In Surprise Defeat. Japan national footballer team began the defence of their AFC Asian Cup title with a lucky 2-1 win against Syria national football teams in a Group C championship match on Friday. But China national Football team saw their luck desert them [...]

Figure 5.1: Example of an Entity Linking (EL) source text and generated outputs. Entity mentions to be recognized and disambiguated are denoted in **blue** in the source text. In the outputs, **red** denotes errors, **green** denotes correct answers, **yellow** denotes close matches.

Luo et al., 2015, Cornolti et al., 2016, Ganea and Hofmann, 2017].

Recent entity linking methods based on language models propose to cast entity linking as a single, end-to-end trained task [Broscheit, 2019, Poerner et al., 2020, El Vaigh et al., 2020], rather than a two-subtask problem. An example is autoregressive entity linking [Petroni et al., 2021, De Cao et al., 2021b], which formulates entity linking as a language generation problem using an encoder-decoder model. A more recent approach [De Cao et al., 2021a] increases performance, and is instead based on a two-step architecture: first mention detection with a transformer encoder, and then autoregressive candidate selection with an LSTM. However, this candidate selection module needs a predefined set of candidate mentions.

Methods based on word embedding models [Basaldella et al., 2020] propose to learn entity disambiguation by mapping embedding spaces. Their high accuracy at 10 results show that re-ranking could increase entity linking performance.

Contributions. In this paper, we propose an autoregressive entity linking method, that is trained jointly with two auxiliary tasks, and learns to re-rank generated samples at inference time. Our proposed novelties address two weaknesses in the literature.

First, instead of the two-step method [De Cao et al., 2021a] that learns to detect mentions and then to select the best entity candidate from a predefined set, we propose to add mention detection as an *auxiliary* task to encoder-decoder-based autoregressive EL. By using encoder-decoder-based autoregressive EL, we bypass the need for a predefined set of candidate mentions, while preserving the benefit of the knowledge learned from mention detection for the main EL task.

Second, previous work suggests that re-ranking could correct prediction errors [Basaldella et al., 2020]. We propose to train a second, new auxiliary task, called *Match Prediction*. This task teaches the model to re-rank generated samples at inference time. We define match prediction as a classification task where the goal is to identify whether entities in a first sentence were correctly disambiguated in the second sentence. We train this second task with samples generated by the

model at each training epoch. At inference time, we then rank the generated samples using our match prediction scores.

Our multi-task learning model outperforms the state of the art in two benchmark datasets of entity linking across two domains: COMETA [Basaldella et al., 2020] from the biomedical and social media domain, and AIDA-CoNLL [Hoffart et al., 2011] from the news domain. We show through three ablation study experiments that each auxiliary task provides improvements on the main task. Then, we show that using our model's match prediction module to re-rank generated samples at inference time plays an important role in increasing performance. Finally, we devise three experiments where we train auxiliary tasks with a smaller dataset. Results suggest that our model's performance is not only due to more training datapoints, but also due to our auxiliary task definition.

5.2 Related Work

Entity Linking (EL). Entity Linking is often [Hoffart et al., 2011, Steinmetz and Sack, 2013, Piccinno and Ferragina, 2014, De Cao et al., 2021a] trained as two tasks: Mention Detection (MD) and Entity Disambiguation (ED). Mention detection is the task of detecting entity mention spans, such that an entity mention m is represented by start and end positions. A mention m refers to a concept in a given knowledge base. Entity disambiguation is the task of finding the right knowledge base concept for an entity mention, thereby *disambiguating* its meaning.

Early EL methods [Hoffart et al., 2011, Steinmetz and Sack, 2013, Daiber et al., 2013] rely on probabilistic approaches. Hoffart et al. [2011] propose a probabilistic framework for MD and ED, based on textual similarity and corpus occurrence. They test their framework using the entity candidate sets available in the AIDA-CoNLL dataset.

More recently, neural methods propose to train end-to-end EL models. Francis-Landau et al. [2016] propose a convolutional neural EL model to take into account windows of context.

Kolitsas et al. [2018] propose a neural model for joint mention detection and entity disambiguation. They use a bidirectional LSTM [Hochreiter and Schmidhuber, 1997] to encode spans of entities. They then embed candidate entities and train layers to score the likelihood of a match.

Sil et al. [2018] introduce an LSTM-based model that uses multilingual embeddings for zero-shot transfer from English-language knowledge bases.

EL as Language Modeling. Language modeling approaches have enabled new, end-to-end definitions of the entity linking task. These new settings enable to bypass the two-step MD-then-ED setting for entity linking, and propose to cast entity linking as a single task.

Broscheit [2019] propose to reformulate end-to-end EL problem as a token-wise classification over the entire set of the vocabulary. Their model is based on BERT [Devlin et al., 2019b]. The training combines mention detection, candidate generation, and entity disambiguation. If an entity is not detected, then the prediction is *O*. If an entity is detected, the classification head has to classify it as the corresponding particular entity within the vocabulary.

De Cao et al. [2021b] propose an autoregressive setting for EL. They use BART [Lewis et al., 2020b] and cast entity linking as a language generation task. In this setting, the input is the source sentence with the entity mention. The goal is to generate an annotated version of the input sentence, such that the entity mention is highlighted and mapped to a knowledge base concept. Brackets and parentheses are used to annotate the entity mention and concept: "*I took the [flu shot] (influenza vaccine).*". They then introduce a constrained beam search to force the model to annotate. De Cao et al. [2021c] is a multilingual extension of this work.

EL as Embedding Space Mapping. Language models like BERT, as well as embedding models like FastText [Bojanowski et al., 2017], enable to retrieve context-aware representations of entities and knowledge base concepts.

Basaldella et al. [2020] propose to map the embeddings of entity mentions to the embeddings of knowledge base concepts. They find that the right mapping is more often found

among the ten closest concept embeddings (accuracy at 10) rather than being the closest concept embedding (accuracy at 1). Their results suggest that generated sample re-ranking could improve entity linking systems.

Concurrently, Wu et al. [2020b] propose a method that uses re-ranking for zero-shot retrieval of entities. They use entity definition embeddings to find candidate entities from a knowledge base, and then train a cross-encoder to re-rank the candidates.

Basaldella et al. [2020] also introduce the COMETA dataset: an entity linking benchmark based on social media user utterances on medical topics, and linked to the SNOMED-CT biomedical knowledge base [DONNELLY, 2006]. The dataset has four splits, based on whether the dev/test set entities are seen during training (stratified) or not (zeroshot), and on whether the entity mapping is context-specific (specific) or not (general). Liu et al. [2021a] propose a self-alignment pre-training scheme for entity embeddings, and show that it benefits the context-free splits (stratified general and zeroshot general). Liu et al. [2021b] propose MirrorBERT: a data-augmented approach for masked language models. Lai et al. [2021] and Kong et al. [2021] propose convolution-based and graph-based methods, respectively, for embedding mapping between entities and knowledge base concepts.

All of the above methods use knowledge base concepts. In our biomedical entity linking setting, we choose the harder zeroshot specific split. We propose to use the language modeling task setting instead of the embedding mapping method. We therefore bypass the need to embed each and every knowledge base concept, whereas only a small portion (¡10%) of the SNOMED-CT knowledge base concepts are used in the COMETA dataset.

5.3 Multi-Task Learning for Autoregressive Entity Linking

We propose an autoregressive entity linking model, that is trained along with two auxiliary tasks, and uses re-ranking at inference time.

In this section, we first describe the main entity linking task. Then, we define the two auxiliary tasks: Mention Detection and a new task, called *Match Prediction*. Third, we train our multi-task learning architecture with a weighted objective. Finally, we propose to use the match prediction module for re-ranking during inference. An overview of our architecture is in Figure 5.2.

5.3.1 Autoregressive Entity Linking

We train autoregressive entity linking as a language generation task. We follow the setting of the encoder-decoder model of De Cao et al. [2021b]. They train their model to generate the input sentence containing both the entity mention *and* the target entity, annotated with parentheses and brackets. For simplicity, we omit these annotations from the examples in the figures.

For entity linking (EL), we optimize the following negative log-likelihood loss:

$$\mathcal{L}_{\text{EL}} = -\sum_{i=1}^{N} \log P(y_i | y_1, ..., y_{i-1}, \mathbf{x}) \tag{5.1}$$

where $\mathbf{x}$ is the input sentence, and $\mathbf{y}$ is the output sentence of length N.

5.3.2 Entity Mention Detection

We introduce mention detection (MD) as an auxiliary task to encoder-decoder autoregressive EL, in order for the knowledge learned from MD to benefit the main EL task, while bypassing the need for predefined candidate sets. MD teaches the model to distinguish tokens that are part of entities from tokens that are not part of any entity. As a result, this task is in essence a token-wise binary classification task. Broscheit [2019] propose a similar task definition, but combine entity detection with entity disambiguation. Their task definition is a classification task over the entire knowledge base vocabulary, rather than our binary setting.

In this task, we train the model to predict where the tokens of the entities are in the input

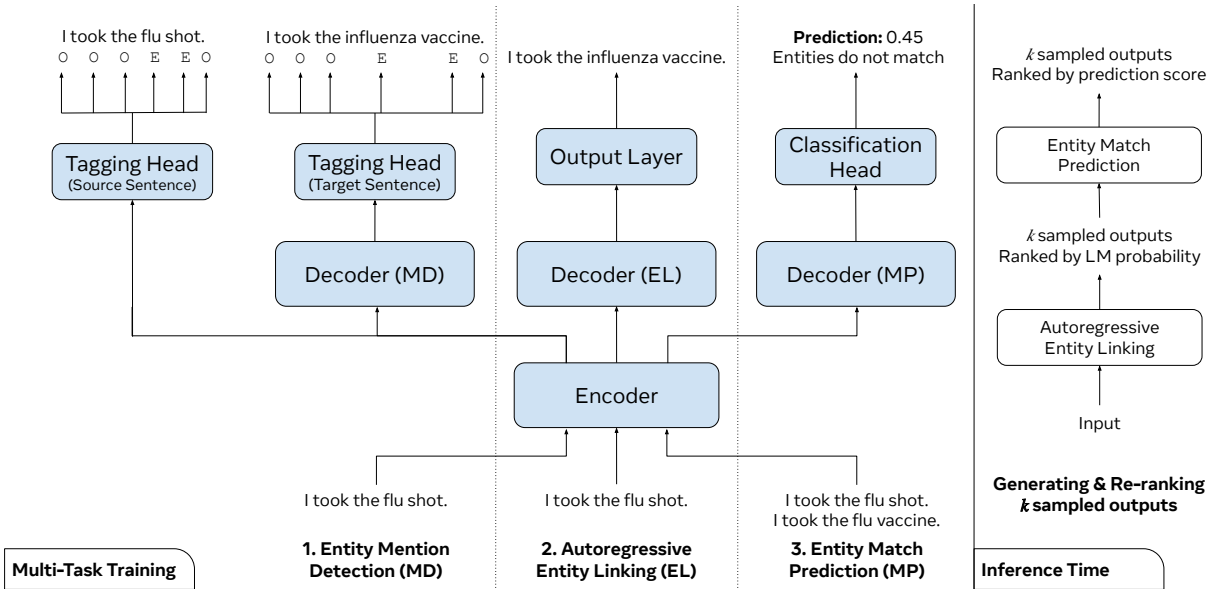


Figure 5.2: Architecture of our proposed multi-task autoregressive entity linking model. Each task is trained using a shared encoder and a task-specific decoder and output layer. The auxiliary mention detection task uses datasets derived from one entity linking dataset, whereas the match prediction task uses sampled outputs. At inference time, we use the match prediction module to re-rank generated samples.

sentence and in the target (annotated) sentence. Therefore, this auxiliary task has to output two sequences of entity indicators: "*E*" for entity mention or concept tokens, and "*O*" for all other tokens. To train our model to generate sequences for the input and target sentences, we augment our existing dataset. We create two datasets of the same size: the first has sequences of entity indicators for the input sentences, and the second has sequences of entity indicators for the target sentences.

As shown at the left of Figure 5.2, we use two different tagging heads for mention detection: one for the input sentence, and one for the output sentence. We use two tagging heads as the model learns different mappings from two different kinds of input. For the input sentence, we feed the encoder embeddings to the first tagging head. We cast this as a classification problem. For mention detection on the output sentence, we use a separate decoder, and feed this decoder's embeddings to the second tagging head. We cast this task as a generation task. For both tasks, we optimize a cross entropy (CE) loss. In summary, we optimize the following loss function for

mention detection (MD):

$$\begin{aligned}\mathcal{L}_{\text{MD}} =& \text{CE}\left(Enc(\mathbf{x}), Ent(\mathbf{x})\right) \\ &+ \text{CE}\left(Dec(Enc(\mathbf{x})), Ent(\mathbf{y})\right)\end{aligned} \tag{5.2}$$

where $Enc(\cdot)$ is the encoder representation, $Dec(\cdot)$ is the decoder representation, and $Ent(\cdot)$ indicates the corresponding sequence of entity indicators.

The method of De Cao et al. [2021a] has two steps, where the first step is to detect mentions. Here, mention detection is an auxiliary task rather than a main part of the pipeline. We employ encoder-decoder autoregressive EL as our main end-to-end pipeline.

5.3.3 Entity Match Prediction

In their biomedical entity linking experiments using word embedding space mapping, Basaldella et al. [2020] find that accuracy at 10 is often more than double the accuracy at 1. They then suggest that re-ranking could significantly improve performance. We build on this observation to introduce the second auxiliary task: entity match prediction (MP). The goal of this task is to teach the model to re-rank generated samples based on the input sentence, with the aim to help narrow the gap with the accuracy at 10 scores.

The input to this task is composed of two sentences: the first one is the input sentence, and the second is a sentence where entity mentions are replaced by entities that may or may not be the matching target entities. We train the model to predict whether the entities match (score of 1) or not (score of 0) between both sentences. The entity match must be complete – all target entities must be generated – for a score of 1.

At regular intervals during training, we generate k samples for each input sentence using beam search on the autoregressive entity linking part of the trained model. We then form k sentence pairs. The corresponding ground truth label for a given sentence pair indicates whether

the entities match or not. This data generation setting exposes the model to its own successes and failures in the main entity linking task.

It may be the case that no generated sample contains entities that match the input sentence, and therefore that all labels for a pair are 0. In this case, the model would not be shown what an example of matching entities looks like. To mitigate this issue, we decide to add one additional sentence pair, where the second sentence is the target sentence used in the autoregressive entity linking training. We add this additional sentence pair to all datapoints for consistency.

We train entity match prediction using a mean squared error loss:

$$\begin{aligned}\mathcal{L}_{\text{MP}} = &\left(P^{\text{MP}}(\hat{\mathbf{y}}|\mathbf{x}) - 1\right)^2 \\ &+ \sum_{i=1}^{k} \left(P^{\text{MP}}(\mathbf{y}_i^s|\mathbf{x}) - \hat{y}_i^{\text{MP}}\right)^2\end{aligned} \tag{5.3}$$

where $\hat{\mathbf{y}}$ is the target sentence, $\mathbf{y}_i^s$ is the i-th generated sample, $P^{\text{MP}}(\cdot|\cdot)$ is the probability that the entities in the left-hand sequence match the ones in the right-hand sequence, and $\hat{y}_i^{\text{MP}}$ is the ground truth label for entity match prediction for the i-th generated sample.

De Cao et al. [2021a] propose to rank candidate concepts from a predefined set after the detecting entity mentions. In our case, we do not learn to rank predefined sets of candidates, nor do we rank concepts. Instead, we generate sentences using beam search, and propose to learn to re-rank them.

5.3.4 Multi-Task Learning

We propose to optimize simultaneously for all three tasks using a single loss function. We set one weight for each auxiliary task. We discuss the task weight hyperparameter tuning in §5.4.3.

Given the losses defined in equations 5.1, 5.2, and 5.3, our loss function for multi-task

learning is as follows:

$$\mathcal{L}_{\text{MTL}} = \mathcal{L}_{\text{EL}} + \lambda_{\text{MD}}\mathcal{L}_{\text{MD}} + \lambda_{\text{MP}}\mathcal{L}_{\text{MP}} \tag{5.4}$$

where λ_{MD} and λ_{MP} are the auxiliary task weights for mention detection and match prediction, respectively.

As shown in Figure 5.2, we use three separate decoders for training: one for each task. We use two separate tagging heads for mention detection. For the match prediction task, we feed the last decoder output to the classification head. This follows the training scheme of BART [Lewis et al., 2020b] for sentence classification tasks.

Our proposed multi-task definition is inspired by our prior work [Mrini et al., 2021b,c]. In our prior research papers, we introduce multi-task learning architectures for biomedical question summarization and entailment. We find that closely related tasks benefit each other during learning, through either multi-task learning or transfer learning [Mrini et al., 2021d].

Our model architecture is also inspired by MT-DNN [Liu et al., 2019a], a multi-task model that obtained state-of-the-art results across many NLP tasks involving sentence representation. In the MT-DNN architecture, the encoder is shared across tasks, and prediction heads are task-specific. Nonetheless, other multi-task architectures remain compatible with our auxiliary tasks and re-ranking, which are the novelties we focus on in this work.

5.3.5 Inference-time Re-ranking

In order to bridge some of the gap between accuracy at 1 and accuracy at 10 [Basaldella et al., 2020], we propose to use the entity match prediction module to re-rank generated samples. The right side of Figure 5.2 illustrates the process.

At inference time, we first generate k samples ranked by their language modeling probability. We then use the separate entity match prediction (MP) decoder to predict an entity match

Table 5.1: Statistics of Entity Linking benchmark datasets.

	AIDA-CoNLL		COMETA
Split	**Documents**	**Mentions**	**Mentions**
Train	942	18,540	13,714
Dev	216	4,791	2,018
Test	230	4,485	4,283

probability. To do so, we input the source sentence and a generated sample to the MP decoder. We use the resulting MP probabilities to re-rank the k generated samples. We select the sample with the highest MP probability to compute the evaluation metrics.

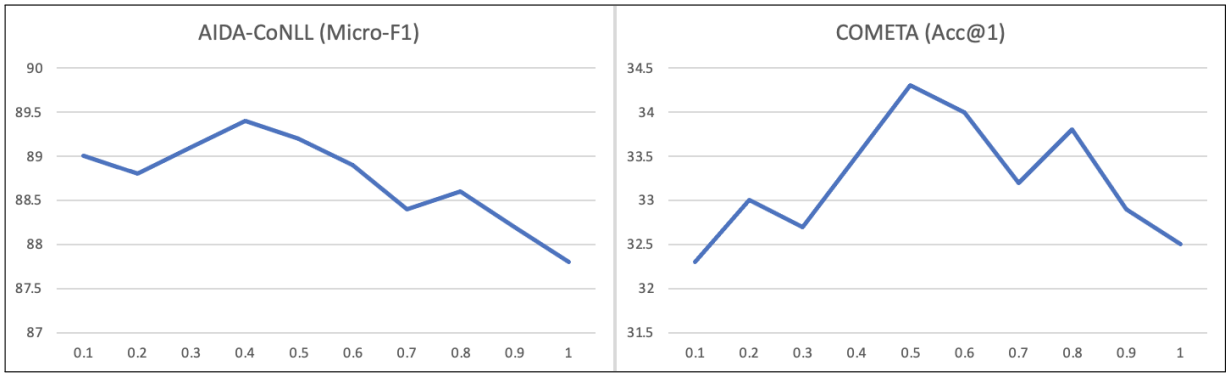
5.4 Experiments

5.4.1 Datasets and Setup

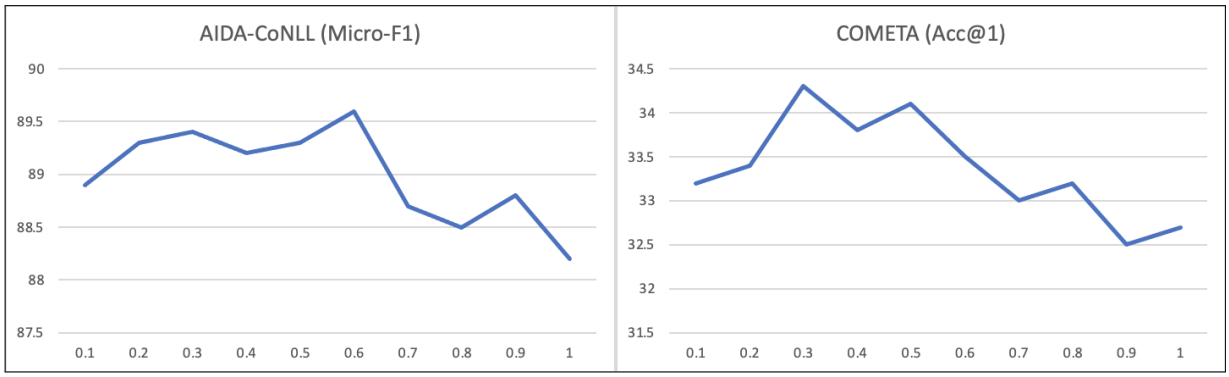
We use two benchmark datasets for English-language entity linking. We use the standard data splits for both datasets, as detailed in Table 5.1.

AIDA-CoNLL [Hoffart et al., 2011] is a dataset consisting of annotated news articles from the Reuters Corpus [Lewis et al., 2004]. The knowledge base concepts come from the titles of the English-language Wikipedia. Each news article contains multiple entity mentions. Articles are sometimes too long for the maximum sequence length of our model. We follow De Cao et al. [2021a] and cut the articles into separate chunks. We use the Micro-F1 metric for evaluation. We only evaluate mentions present in the knowledge base, following the *In-KB* setting [Röder et al., 2018], in line with previous work [De Cao et al., 2021b,a]. This dataset contains candidates for each entity mention. We do not use entity candidates, although several baselines do [Kolitsas et al., 2018, Martins et al., 2019, De Cao et al., 2021a].

COMETA [Basaldella et al., 2020] is a dataset of biomedical entity mentions from social media (Reddit) utterances. In this dataset, each user-written utterance contains exactly one entity mention. The metric used to evaluate this dataset is accuracy at 1 (Acc@1). We measure Acc@1



(a) Choosing the optimal λ_{MD}, setting $\lambda_{MP} = 0.3$.



(b) Choosing the optimal λ_{MP}, given the optimal λ_{MD}.

Figure 5.3: Task weight tuning on the dev set for Mention Detection (MD) and Match Prediction (MP). We first optimize for λ_{MD} (a), and then λ_{MP} (b).

by checking whether the correct knowledge base concept is present in the top generated sample. We use the zeroshot specific split, where the entity mention and disambiguation pairs in the test set are not seen during training, and the entity linking is context-specific.

5.4.2 Training Details

We use BART Large [Lewis et al., 2020b] as our base model. We use three decoders, all initialized from the same checkpoint decoder. We found in initial experiments that separate decoders for all tasks benefit the main EL task. We train for 100 epochs on AIDA-CoNLL, and for 10 epochs on COMETA.

5.4.3 Task Weight Tuning

For each dataset, we optimize the auxiliary task weights λ_{MD} for mention detection, and λ_{MP} for match prediction. We select these hyperparameters based on the highest performance in Micro-F1 (AIDA-CoNLL) or accuracy at 1 (COMETA) on the dev set.

We trial all values from 0.1 to 1.0 with 0.1 increments, for both task weights. We start by optimizing λ_{MD} given $\lambda_{\text{MP}} = 0.3$, and then optimize λ_{MP} given the optimal λ_{MD} weights. The results are in Figure 5.3. The graphs show that performance on the main entity linking task can vary visibly when the weights of the auxiliary tasks change. The variation is likely due to the large auxiliary task datasets, which could dominate training. Moreover, the optimal task weights are different for every dataset and domain: we find that the optimal auxiliary task weights are $\lambda_{\text{MD}} = 0.4$ and $\lambda_{\text{MP}} = 0.6$ for AIDA-CoNLL, and $\lambda_{\text{MD}} = 0.5$ and $\lambda_{\text{MP}} = 0.3$ for COMETA. We use these task weights for the next experiments.

5.4.4 Results and Discussion

AIDA-CoNLL. The test results for the AIDA-CoNLL dataset are on Table 5.2. Our model establishes a new state of the art for this task.

Compared to the state-of-the-art encoder-decoder autoregressive EL model on AIDA-CoNLL [De Cao et al., 2021b], our method shows a 2.0-point improvement in Micro-F1 score. This increase shows that our model is able to correct some errors with the re-ranking at inference time, and that our multi-task setting benefits the main entity linking task.

Our model scores a Micro-F1 0.2 higher than the model of De Cao et al. [2021a]. However, De Cao et al. [2021a] use a predefined candidate set of concepts, whereas the encoder-decoder autoregressive EL models – including our own – do not. This shows that our model is able to bypass the knowledge base, and that our method leverages language modeling to gain knowledge of the news domain.

Table 5.2: Results on the AIDA-CoNLL test set.

Method	Micro-F1
Hoffart et al. [2011]	72.8
Steinmetz and Sack [2013]	42.3
Daiber et al. [2013]	57.8
Moro et al. [2014]	48.5
Piccinno and Ferragina [2014]	73.0
Kolitsas et al. [2018]	82.4
Peters et al. [2019]	73.7
Broscheit [2019]	79.3
Martins et al. [2019]	81.9
van Hulst et al. [2020]	80.5
Févry et al. [2020]	76.7
Kannan Ravi et al. [2021]	83.1
De Cao et al. [2021a]	85.5
Encoder-Decoder Autoregressive EL Models	
De Cao et al. [2021b]	83.7
Our model	**85.7**

COMETA. There are no predefined sets of candidate concepts in the COMETA dataset. In this task, there is a knowledge base of biomedical concepts from which the model can choose. Similarly to our AIDA-CoNLL setting, our model does not use the knowledge base.

We consider three baselines for our biomedical entity linking benchmark. The first baseline is the embedding mapping method of Basaldella et al. [2020]. They use BioBERT and a max-margin loss with negative target embeddings. The second baseline is the BERT- and classification-based method of Broscheit [2019]. We train this baseline by classifying tokens into the concepts present in the COMETA dataset, as opposed to the entire vocabulary of $350K$ knowledge base concepts. This is for computational purposes, as a $350K$-way classification would be difficult to train. The third baseline is the autoregressive, single-task model of De Cao et al. [2021b]. We train this baseline as a reference point for our model. We do not include De Cao et al. [2021a] as a baseline, as their method uses predefined sets of candidate concepts, and COMETA does not include them.

The test results of the COMETA dataset experiments are on Table 5.3. Our model is able

Table 5.3: Results on the COMETA test set.

Method	Acc@1
Basaldella et al. [2020]	27.0
Broscheit [2019]	24.5
Encoder-Decoder Autoregressive EL Models	
De Cao et al. [2021b]	30.9
Our model	**32.4**

to exceed over five percentage points the baselines that use the knowledge base concepts. This shows that our method can efficiently generalize without the need for a knowledge base, but only through learning about the biomedical domain. Note that we use the zeroshot specific split here, where the entity mention and disambiguation pairs in the test set are not seen during training. Moreover, our model exceeds the autoregressive single-task baseline by 1.5%. This increase shows that our multi-task setting and re-ranking can generalize, and increase performance under zeroshot settings.

5.4.5 Ablation Studies

We perform two types of ablation studies to analyze the added value of our novelties. First, we evaluate how do the two auxiliary tasks and the re-ranking impact entity linking performance. Second, we implement a low-resource scenario for the auxiliary tasks, as we ask whether the main task benefits more from the knowledge learned the auxiliary tasks, or from the additional training data.

Auxiliary Tasks and Re-ranking. Our main novelties are multi-task learning with two auxiliary tasks, and the re-ranking of generated samples at inference time. The first auxiliary task, mention detection, aims to preserve the knowledge learned from detecting mentions of entities, while allowing the encoder-decoder model to bypass the need for predefined sets of entity candidates. The second auxiliary task, match prediction, aims to teach the model how to predict whether entities were correctly disambiguated given an input sentence and a generated sample.

Table 5.4: Results of the ablation studies on the test sets. We perform ablation studies on Mention Detection (**MD**), Match Prediction (**MP**), and the re-ranking of generated samples (**Rk**).

			AIDA-CoNLL	COMETA
MD	**MP**	**Rk**	**Micro-F1**	**Acc@1**
Ablation of Auxiliary Tasks and Re-ranking				
✗	✗	✗	83.7	30.9
Ablation of Auxiliary Tasks				
✓	✗	✗	84.3	31.2
✗	✓	✓	85.4	32.1
Ablation of Re-ranking				
✓	✓	✗	84.8	31.5
MD, MP and Re-ranking (Ours)				
✓	✓	✓	85.7	32.4

We perform ablation studies to gauge the added value of each task and re-ranking. We perform three additional experiments, keeping the same number of model parameters. First, we remove the match prediction training objective ($\lambda_{\text{MP}} = 0.0$), and therefore also remove the re-ranking, but we keep the optimally weighted mention detection objective. Second, we remove the mention detection training objective by setting $\lambda_{\text{MD}} = 0.0$, but we keep the optimally weighted mention prediction objective, along with the re-ranking. Third, we keep both optimally weighted auxiliary tasks, but remove the inference-time re-ranking of generated samples. Finally, we compare our results to De Cao et al. [2021b] as it does not have both auxiliary tasks nor the re-ranking.

We show the results of all ablation experiments on the dev sets in Table 5.4. The lowest scores are obtained when both auxiliary tasks and re-ranking are ablated. This shows the added value of all of our main novelties on the main entity linking task. In addition, each auxiliary task individually increases performance, as shown on the second and third row of results. The auxiliary match prediction task along with re-ranking provide a larger performance increase than the auxiliary mention detection task alone. This could be due to the fact that the match prediction task gets a larger number of samples to train on. Finally, the difference in performance between

our model and the re-ranking ablation study shows that re-ranking of generated samples is an important contribution to the final performance. This result backs the suggestion of Basaldella et al. [2020] that re-ranking can bridge some of the gap between Acc@1 and Acc@10.

Impact of additional training data. In this subsection, we ask whether the main task benefits more from the knowledge learned by the auxiliary tasks, or from the large sizes of the auxiliary task datasets. The mention detection task has two datapoints for every EL datapoint, while the match prediction task has $k+1 = 11$ datapoints for every EL datapoint. Therefore, in a given training epoch, there are more datapoints to train on for the auxiliary tasks in comparison with the main task.

We devise three experiments to gauge whether a lower amount of training datapoints for auxiliary tasks impacts the main task results. We propose a low-resource regimen of training for auxiliary tasks, such that we bring the ratio of training datapoints down to 1:1 between the auxiliary tasks and the main task. We train on one out of every two MD datapoints, and on one of out every 11 MP datapoints. In other words, we skip 50% of the training data of the MD task, and 91% of the training data of the MP task. We spread out the input such that, at each training step, the model sees one EL input sentence, one MD input sentence, and one MP input sentence pair. In each epoch, we skip the same datapoints so that the model only sees a reduced number of training datapoints.

In the first experiment, we train for both auxiliary tasks on a train set ratio of 1:1 with the main task. In the second and third experiments, we apply the low-resource setting only to the mention detection task, and only to the match prediction task, respectively. In all three experiments, we keep the same selection of skipped datapoints for each task, and we keep re-ranking.

We show the results of the low-resource experiments in Table 5.5. For reference, we add the results from our model and the model without auxiliary task nor re-ranking of De Cao et al. [2021b]. The results show that globally, there is a slight decrease in performance when the

Table 5.5: Results on the test sets of the low-resource experiments. We reduce the training datasets of the auxiliary mention detection **MD** and match prediction **MP** tasks to measure the benefit of multi-task learning.

% of Train Set		AIDA-CoNLL	COMETA
MD	**MP**	**Micro-F1**	**Acc@1**
Ablation of Auxiliary Tasks and Re-ranking			
0%	0%	83.7	30.9
Low-Resource Experiments			
50%	9%	84.5	32.0
50%	100%	85.4	31.4
100%	9%	84.5	31.8
No Low-Resource (Ours)			
100%	100%	85.7	32.4

training set is smaller, compared to our model. However, the low-resource experiments show a significant increase in performance compared to the ablation experiment of the first row. This shows that our proposed method's edge does not only come from the additional training data, but also from our formulation of the auxiliary tasks, and the re-ranking of generated samples.

5.5 Conclusions

We propose a multi-task learning and re-ranking approach to autoregressive entity linking. Our main two novelties address two weaknesses in the literature. First, whereas the two-step method of De Cao et al. [2021a] improves performance, it relies on predefined sets of entity candidates. We propose to instead train mention detection as an auxiliary task to autoregressive EL, in order to bypass the need for entity candidate sets, and to preserve the knowledge learned by mention detection. Second, previous work suggests that a sizeable portion of errors could be corrected with re-ranking. We propose to use samples generated at training time to teach the model to re-rank outputs.

Our model establishes a new state of the art in both COMETA and AIDA-CoNLL. The increases in performance across both datasets show that our model can learn and leverage domain-

specific knowledge, without using a candidate set or a knowledge base. To analyse our model, we devise three ablation study experiments, and show that our model benefits from both auxiliary tasks and re-ranking. In particular, we show that re-ranking plays a major role in increasing entity linking scores. Then, we propose three low-resource experiments for auxiliary tasks. The results show that our model's performance is not only due to additional training datapoints, but also due to how we defined our auxiliary tasks.

Acknowledgements

This chapter is a reformatted version of the material as it appears in "Detection, Disambiguation, Re-ranking: Autoregressive Entity Linking as a Multi-Task Problem," Khalil Mrini, Shaoliang Nie, Jiatao Gu, Sinong Wang, Maziar Sanjabi, Hamed Firooz [Mrini et al., 2022]. The material has been submitted, accepted and published at the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022. The dissertation author was the primary investigator and first author of this paper.

The dissertation author performed this work during an internship at Meta AI (previously Facebook AI). We thank Jingbo Shang and Ndapa Nakashole for insightful discussions. We thank the anonymous reviewers for their feedback.

Ethical Considerations

This work deals with user-generated text in the medical domain. However, our work and models should not be used as text understanding tools for real-life medical systems without human supervision and verification. Our system is not error-free, and using it could lead to a misunderstanding of the true intentions of people seeking medical care.

Chapter 6

Conclusions

In this dissertation, I presented my research on text understanding and question answering for user-written utterances in the low-resource medical domain. I shed light on lesser known benchmark datasets that contain user-written text. In answer selection and summarization, user-written text datasets have been ignored in favor of traditional benchmarks consisting of formally written and encyclopedia-like text. I showed that transformers, currently the most popular model architecture in NLP, produce mixed results for user-written text, even when there is consistent improvement for the traditional, formally written benchmarks. Throughout my time working with consumer health datasets and traditional benchmark datasets, I observed that there are far fewer baselines to work with in user-written text datasets. In many chapters, I spent a considerable time determining which baselines to consider, and then trained them on our own for consumer health text datasets. A recommendation would be that more work needs to be done for marginalized forms of writing, and in particular users who have non-traditional or underrepresented writing styles. To achieve this, a human-centered approach to text representation learning should be preferred over the predominant benchmark-based one.

My dissertation focuses as well on developing new task formulations that can help alleviate the problem of scarce datasets. In consumer health question understanding, the particular

definition of question summarization and entailment enabled us to design data-augmented multi-task learning architectures. In consumer health question understanding and answering, self-supervised losses enable to find alternatives for labels, that would otherwise be costly and difficult to obtain due to the technical knowledge required. In consumer health entity linking, I find that defining new auxiliary tasks with synthetically created datasets enables us to take achieve better performance on the main task, even under low-resource settings. Through this work, I aim to highlight that less popular, technical domains require methods that adapt to their specificities. As such, this opens the door for many opportunities to develop domain-specific methods, and enable those domains to also benefit from tremendous advances in human language technology.

References

A. Agrawal, R. A. George, S. S. Ravi, S. Kamath, and A. Kumar. Ars_nitk at mediqa 2019: analysing various methods for natural language inference, recognising question entailment and medical question answering system. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 533–540, 2019.

M. Ahmed, M. R. Samee, and R. E. Mercer. You only need attention to traverse trees. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 316–322, 2019.

E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, and M. McDermott. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

G. Balikas, A. Krithara, I. Partalas, and G. Paliouras. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In H. Müller, O. A. Jimenez del Toro, A. Hanbury, G. Langs, and A. Foncubierta Rodriguez, editors, *Multimodal Retrieval in the Medical Domain*, pages 26–39, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24471-6.

M. Basaldella, F. Liu, E. Shareghi, and N. Collier. Cometa: A corpus for medical entity linking in the social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3122–3137, 2020.

P. Basile, V. Basile, M. Nissim, and N. Novielli. Deep tweets: from entity linking to sentiment analysis. In *Proceedings of the Italian Computational Linguistics Conference (CLiC-it 2015)*, 2015.

I. Beltagy, K. Lo, and A. Cohan. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611, 2019.

A. Ben Abacha and D. Demner-Fushman. Recognizing question entailment for medical question answering. In *AMIA Annual Symposium Proceedings*, volume 2016, page 310. American Medical Informatics Association, 2016.

A. Ben Abacha and D. Demner-Fushman. Nlm_nih at semeval-2017 task 3: from question entailment to question similarity for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 349–352, 2017.

A. Ben Abacha and D. Demner-Fushman. A question-entailment approach to question answering. *BMC bioinformatics*, 20(1):511, 2019a.

A. Ben Abacha and D. Demner-Fushman. On the summarization of consumer health questions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2228–2234, 2019b.

A. Ben Abacha and P. Zweigenbaum. Medical question answering: Translating medical questions into sparql queries. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, page 41–50, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450307819. doi: 10.1145/2110363.2110372. URL https://doi.org/10.1145/2110363.2110372.

A. Ben Abacha and P. Zweigenbaum. Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information processing & management*, 51(5): 570–594, 2015.

A. Ben Abacha, E. Agichtein, Y. Pinter, and D. Demner-Fushman. Overview of the medical question answering task at trec 2017 liveqa. In *TREC*, 2017.

A. Ben Abacha, C. Shivade, and D. Demner-Fushman. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 370–379, 2019.

A. Ben Abacha, Y. Mrabet, Y. Zhang, C. Shivade, C. Langlotz, and D. Demner-Fushman. Overview of the mediqa 2021 shared task on summarization in the medical domain. In *Proceedings of the 20th SIGBioMed Workshop on Biomedical Language Processing, NAACL-BioNLP 2021*. Association for Computational Linguistics, 2021.

L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

W. Bian, S. Li, Z. Yang, G. Chen, and Z. Lin. A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1987–1990, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3133089. URL https://doi.org/10.1145/3132847.3133089.

S. Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

S. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

S. Broscheit. Investigating entity knowledge in bert with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, 2019.

R. Cai, B. Zhu, L. Ji, T. Hao, J. Yan, and W. Liu. An cnn-lstm attention approach to understanding user query intent from online health communities. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 430–437. IEEE, 2017.

D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, 2017a.

L. Chen, D. Zhang, and L. Mark. Understanding user intent in community question answering. In *Proceedings of the 21st international conference on world wide web*, pages 823–828, 2012.

Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An. Enhancing recurrent neural networks with positional attention for question answering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–996, 2017b.

Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada, July 2017c. Association for Computational Linguistics. doi: 10.18653/v1/P17-1152. URL https://www.aclweb.org/anthology/P17-1152.

Q. Chen, Q. Hu, J. X. Huang, and L. He. Can: Enhancing sentence similarity modeling with collaborative and adversarial network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 815–824, 2018a.

Q. Chen, Q. Hu, J. X. Huang, and L. He. Ca-rnn: using context-aligned recurrent neural networks for modeling sentence similarity. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

S. Chen, Z. Ju, X. Dong, H. Fang, S. Wang, Y. Yang, J. Zeng, R. Zhang, R. Zhang, M. Zhou, P. Zhu, and P. Xie. Meddialog: a large-scale medical dialogue dataset. *arXiv preprint arXiv:2004.03329*, 2020.

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, 2018.

M. Cornolti, P. Ferragina, M. Ciaramita, S. Rüd, and H. Schütze. A piggyback system for joint entity mention detection and linking in web queries. In *Proceedings of the 25th International Conference on World Wide Web*, pages 567–578, 2016.

I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer, 2005.

I. Dagan, D. Roth, M. Sammons, and F. M. Zanzotto. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220, 2013.

J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124, 2013.

N. De Cao, W. Aziz, and I. Titov. Highly parallel autoregressive entity linking with discriminative correction. *arXiv preprint arXiv:2109.03792*, 2021a.

N. De Cao, G. Izacard, S. Riedel, and F. Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=5k8F6UU39V.

N. De Cao, L. Wu, K. Popat, M. Artetxe, N. Goyal, M. Plekhanov, L. Zettlemoyer, N. Cancedda, S. Riedel, and F. Petroni. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*, 2021c.

D. Demner-Fushman, Y. Mrabet, and A. Ben Abacha. Consumer health information and question answering: helping consumers find answers to their health-related information needs. *Journal of the American Medical Informatics Association*, 27(2):194–201, 2020.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019b.

B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1168. URL https://www.aclweb.org/anthology/P17-1168.

E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*, 2018.

K. DONNELLY. Snomed-ct: The advanced terminology and coding system for ehealth. *Medical and Care Compunetics 3*, 121:279, 2006.

M. Dubey, D. Banerjee, D. Chaudhuri, and J. Lehmann. Earl: joint entity and relation linking for question answering over knowledge graphs. In *International Semantic Web Conference*, pages 108–126. Springer, 2018.

S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045. URL https://www.aclweb.org/anthology/D18-1045.

C. B. El Vaigh, F. Torregrossa, R. Allesiardo, G. Gravier, and P. Sébillot. A correlation-based entity embedding approach for robust entity linking. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 949–954. IEEE, 2020.

T. Falke, L. F. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, 2019.

T. Févry, L. Baldini Soares, N. FitzGerald, E. Choi, and T. Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.400. URL https://aclanthology.org/2020.emnlp-main.400.

M. Francis-Landau, G. Durrett, and D. Klein. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261, 2016.

O.-E. Ganea and T. Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, 2017.

S. Garg, T. Vu, and A. Moschitti. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *arXiv preprint arXiv:1911.04118*, 2019.

D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9, 2007.

T. R. Goodwin and S. M. Harabagiu. Knowledge representations and inference techniques for medical question answering. *ACM Trans. Intell. Syst. Technol.*, 9(2), Oct. 2017. ISSN 2157-6904. doi: 10.1145/3106745. URL https://doi.org/10.1145/3106745.

A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

J. A. G. Groenendijk and M. J. B. Stokhof. *Studies on the Semantics of Questions and the Pragmatics of Answers*. PhD thesis, Univ. Amsterdam, 1984.

H. Guo, R. Pasunuru, and M. Bansal. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697, 2018.

S. Guo, M.-W. Chang, and E. Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030, 2013.

A. Gupta, M. Kaur, S. Mirkin, A. Singh, and A. Goyal. Text summarization through entailment-based minimum vertex cover. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)*, pages 75–80, 2014.

R. B. Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774, 2011.

Y. Hao, X. Liu, J. Wu, and P. Lv. Exploiting sentence embedding for medical question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 938–945, 2019.

J. Harer, C. Reale, and P. Chin. Tree-transformer: A transformer-based method for correction of tree-structured data. *arXiv preprint arXiv:1908.00449*, 2019.

Y. He, Z. Zhu, Y. Zhang, Q. Chen, and J. Caverlee. Infusing disease knowledge into bert for health question answering, medical inference and disease name recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4604–4614, 2020.

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

J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, 2011.

K. Huang, J. Altosaar, and R. Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

S. Iyer, N. Dandekar, and K. Csernai. First quora dataset release: Question pairs. 2017.

A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

J. Johnson, K. Mrini, A. Moore, E. Farkas, N. Nkashole, M. Hogarth, and N. Weibel. Voice-based conversational agents for older adults. In *Proceedings of the CHI 2020 Workshop on Conversational Agents for Health and Wellbeing, Honolulu, Hawaii*, 2020. URL `http://voli.ucsd.edu/pdfs/CHI2020_Workshop_VOLI.pdf`.

S. Kamath, B. Grau, and Y. Ma. Predicting and integrating expected answer types into a simple recurrent neural network model for answer sentence selection. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*, 2019.

M. P. Kannan Ravi, K. Singh, I. O. Mulang', S. Shekarpour, J. Hoffart, and J. Lehmann. CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 504–514, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.40. URL `https://aclanthology.org/2021.eacl-main.40`.

V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL `https://aclanthology.org/2020.emnlp-main.550`.

E. Kiperwasser and Y. Goldberg. Easy-first dependency parsing with hierarchical tree lstms. *Transactions of the Association for Computational Linguistics*, 4:445–461, 2016.

N. Kolitsas, O.-E. Ganea, and T. Hofmann. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, 2018.

L. Kong, C. Winestock, and P. Bhatia. Zero-shot medical entity retrieval without annotation: Learning from rich knowledge graph semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2401–2405, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.212. URL https://aclanthology.org/2021.findings-acl.212.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

T. Lai, Q. H. Tran, T. Bui, and D. Kihara. A gated self-attention memory network for answer selection. 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 5955–5961, 2019.

T. Lai, H. Ji, and C. Zhai. Bert might be overkill: A tiny but effective biomedical entity linker based on residual convolutional neural networks. *arXiv preprint arXiv:2109.02237*, 2021.

M. T. R. Laskar, X. Huang, and E. Hoque. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5505–5514, 2020.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

C. Lei, V. Efthymiou, R. Geis, and F. Ozcan. Expanding query answers on medical knowledge bases. In *EDBT*, pages 567–578, 2020.

D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting*

of the Association for Computational Linguistics, pages 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020b.

H. Li, J. Zhu, J. Zhang, and C. Zong. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, 2018.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

F. Liu, E. Shareghi, Z. Meng, M. Basaldella, and N. Collier. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, 2021a.

F. Liu, I. Vulić, A. Korhonen, and N. Collier. Fast, effective and self-supervised: Transforming masked languagemodels into universal lexical and sentence encoders. *arXiv preprint arXiv:2104.08027*, 2021b.

X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1304–1311, 2013.

X. Liu, P. He, W. Chen, and J. Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, 2019a.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

E. Lloret, O. Ferrández, R. Munoz, and M. Palomar. A text summarization approach under the influence of textual entailment. In *NLPCS*, pages 22–31, 2008.

E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and*

Computational Linguistics - Volume 1, ETMTNLP '02, page 63–70, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118117. URL https://doi.org/10.3115/1118108.1118117.

G. Luo, X. Huang, C.-Y. Lin, and Z. Nie. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, 2015.

H. T. Madabushi, M. Lee, and J. Barnden. Integrating question classification and deep learning for improved answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3283–3294, 2018.

Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, 2021.

M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, page 114–119, USA, 1994. Association for Computational Linguistics. ISBN 1558603573. doi: 10.3115/1075812.1075835. URL https://doi.org/10.3115/1075812.1075835.

P. H. Martins, Z. Marinho, and A. F. Martins. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196, 2019.

C. McCreery, N. Katariya, A. Kannan, M. Chablani, and X. Amatriain. Domain-relevant embeddings for medical question similarity. *arXiv preprint arXiv:1910.04192*, 2019.

Y. Mehdad, G. Carenini, F. Tompa, and R. Ng. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, 2013.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119, 2013.

G. A. Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

A. Moro, A. Raganato, and R. Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 05 2014. ISSN 2307-387X. doi: 10.1162/tacl_a_00179. URL https://doi.org/10.1162/tacl_a_00179.

K. Mrini, C. Musat, M. Baeriswyl, and M. Jaggi. Structure tree-lstm: Structure-aware attentional document encoders. *arXiv preprint arXiv:1902.09713*, 2019.

K. Mrini, F. Dernoncourt, Q. H. Tran, T. Bui, W. Chang, and N. Nakashole. Rethinking self-attention: Towards interpretability in neural parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 731–742, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.65. URL https://www.aclweb.org/anthology/2020.findings-emnlp.65.

K. Mrini, C. Chen, N. Nakashole, N. Weibel, and E. Farcas. Medical question understanding and answering for older adults. *The 3rd Southern California (SoCal) NLP Symposium*, 2021a. URL http://voli.ucsd.edu/pdfs/2021_VOLI_SoCal_NLP.pdf.

K. Mrini, F. Dernoncourt, W. Chang, E. Farcas, and N. Nakashole. Joint summarization-entailment optimization for consumer health question understanding. In *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*, pages 58–65, Online, June 2021b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.nlpmc-1.8.

K. Mrini, F. Dernoncourt, S. Yoon, T. Bui, W. Chang, E. Farcas, and N. Nakashole. A gradually soft multi-task and data-augmented approach to medical question understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1505–1515, Online, Aug. 2021c. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.119. URL https://aclanthology.org/2021.acl-long.119.

K. Mrini, F. Dernoncourt, S. Yoon, T. Bui, W. Chang, E. Farcas, and N. Nakashole. UCSD-adobe at MEDIQA 2021: Transfer learning and answer sentence selection for medical summarization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 257–262, Online, June 2021d. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.bionlp-1.28.

K. Mrini, E. Farcas, and N. Nakashole. Recursive tree-structured self-attention for answer sentence selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4651–4661, Online, Aug. 2021e. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.358. URL https://aclanthology.org/2021.acl-long.358.

K. Mrini, S. Nie, J. Gu, S. Wang, M. Sanjabi, and H. Firooz. Recursive tree-structured self-attention for answer sentence selection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, May 2022. Association for Computational Linguistics.

P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, and B. Randeree. SemEval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 269–281, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2047. URL https://www.aclweb.org/anthology/S15-2047.

P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, A. A. Freihat, J. Glass, and B. Randeree. Semeval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545, 2016.

P. Nakov, D. Hoogeveen, L. Màrquez, A. Moschitti, H. Mubarak, T. Baldwin, and K. Verspoor. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, 2017.

R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://www.aclweb.org/anthology/K16-1028.

S. Narayan, S. B. Cohen, and M. Lapata. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

X.-P. Nguyen, S. Joty, S. Hoi, and R. Socher. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*, 2019.

R. Pasunuru and M. Bansal. Multi-reward reinforced summarization with saliency and entailment. 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 646–653, 2018.

R. Pasunuru, H. Guo, and M. Bansal. Towards improving abstractive summarization via entailment generation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 27–32, 2017.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL https://aclanthology.org/D19-1005.

F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Plachouras, T. Rocktäschel, and S. Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200.

F. Piccinno and P. Ferragina. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62, 2014.

N. Poerner, U. Waltinger, and H. Schütze. E-bert: Efficient-yet-effective entity embeddings for bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 803–818, 2020.

A. Raganato, J. Camacho-Collados, and R. Navigli. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, 2017.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

C. Roberts. Information structure in discourse: Towards an integrated formal theory of pragmatics. 1996.

K. Roberts and D. Demner-Fushman. Interactive use of online health resources: a comparison of consumer and professional questions. *Journal of the American Medical Informatics Association*, 23(4):802–811, 2016.

M. Röder, R. Usbeck, and A.-C. Ngonga Ngomo. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625, 2018.

A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

P. Scharpf, M. Schubotz, and B. Gipp. Towards explaining stem document classification using mathematical entity linking. *arXiv preprint arXiv:2109.00954*, 2021.

A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

L. Sha, X. Zhang, F. Qian, B. Chang, and Z. Sui. A multi-view fusion neural network for answer selection. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

S. Shah, A. Mishra, N. Yadati, and P. P. Talukdar. Kvqa: Knowledge-aware visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8876–8884, 2019.

V. Sharma, N. Kulkarni, S. Pranavi, G. Bayomi, E. Nyberg, and T. Mitamura. Bioama: towards an end to end biomedical question answering system. In *Proceedings of the BioNLP 2018 workshop*, pages 109–117, 2018.

S. Shen, Y. Li, N. Du, X. Wu, Y. Xie, S. Ge, T. Yang, K. Wang, X. Liang, and W. Fan. On the generation of medical question-answer pairs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8822–8829, 2020.

A. Sil, G. Kundu, R. Florian, and W. Hamza. Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011.

R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211, 2012.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

N. Steinmetz and H. Sack. Semantic multimedia information retrieval based on contextual descriptions. In *Extended Semantic Web Conference*, pages 382–396. Springer, 2013.

S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

Z. Sun, Q. Zhu, Y. Xiong, Y. Sun, L. Mou, and L. Zhang. Treegen: A tree-based transformer architecture for code generation. In *AAAI*, pages 8984–8991, 2020.

M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online qa collections. In *Proceedings of ACL-08: HLT*, pages 719–727, 2008.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL https://www.aclweb.org/anthology/P15-1150.

Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui. Learning to rank question answer pairs with holographic dual lstm architecture. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 695–704, 2017.

Y. Tay, L. A. Tuan, and S. C. Hui. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 583–591, 2018a.

Y. Tay, L. A. Tuan, and S. C. Hui. Multi-cast attention networks for retrieval-based question answering and response prediction. *arXiv preprint arXiv:1806.00778*, 2018b.

Z. Teng and Y. Zhang. Head-lexicalized bidirectional tree lstms. *Transactions of the Association for Computational Linguistics*, 5:163–177, 2017.

R. M. Terol, P. Martínez-Barco, and M. Palomar. A knowledge based method for the medical question answering problem. *Computers in biology and medicine*, 37(10):1511–1521, 2007.

Q. H. Tran, G. Haffari, and I. Zukerman. A generative attentional neural network model for dialogue act classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 524–529, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2083. URL `https://www.aclweb.org/anthology/P17-2083`.

Q. H. Tran, N. Dam, T. Lai, F. Dernoncourt, T. Le, N. Le, and D. Phung. Explain by evidence: An explainable memory-based neural network for question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5205–5210, 2020.

K. Tymoshenko and A. Moschitti. Cross-pair text representations for answer sentence selection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2162–2173, 2018.

T. Uslu, A. Mehler, D. Baumartz, and W. Hemati. fastsense: An efficient word sense disambiguation classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

J. M. van Hulst, F. Hasibi, K. Dercksen, K. Balog, and A. P. de Vries. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200, 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

D. Wang and E. Nyberg. Cmu oaqa at trec 2017 liveqa: A neural dual entailment approach for question paraphrase identification. In *TREC*, 2017.

M. Wang, N. A. Smith, and T. Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, 2007.

Y. Wang, H.-Y. Lee, and Y.-N. Chen. Tree transformer: Integrating tree structures into self-attention. 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 1060–1070, 2019.

J. Weizenbaum. Computer power and human reason: From judgment to calculation. 1976.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

C. Wu, G. Luo, C. Guo, Y. Ren, A. Zheng, and C. Yang. An attention-based multi-task model for named entity recognition and intent analysis of chinese online medical questions. *Journal of Biomedical Informatics*, 108:103511, 2020a.

L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, 2020b.

I. Yamada, H. Takeda, and Y. Takefuji. An end-to-end entity linking approach for tweets. In *5th Workshop on Making Sense of Microposts: Big Things Come in Small Packages,# Microposts 2015, at the 24th International Conference on the World Wide Web, WWW 2015*, pages 55–56. CEUR-WS, 2015.

G. Yan and J. Li. Medical question similarity calculation based on weighted domain dictionary. In *Proceedings of the 2018 International Conference on Big Data and Computing*, pages 104–107, 2018.

Y. Yang, W.-t. Yih, and C. Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal, Sept. 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1237. URL https://www.aclweb.org/anthology/D15-1237.

Y. Yang, W.-t. Yih, and C. Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018, 2015b.

Y. Yang, J. Yu, Y. Hu, X. Xu, and E. Nyberg. Cmu livemedqa at trec 2017 liveqa: A consumer health question answering system. *arXiv preprint arXiv:1711.05789*, 2017.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

M. Yu, W. Yin, K. S. Hasan, C. dos Santos, B. Xiang, and B. Zhou. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581, 2017.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

W. Zhang, J. Su, C. L. Tan, and W. T. Wang. Entity linking leveraging automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1290–1298, 2010.

H. Zhou, X. Li, W. Yao, C. Lang, and S. Ning. Dut-nlp at mediqa 2019: an adversarial multi-task network to jointly model recognizing question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 437–445, 2019.

W. Zhu, X. Zhou, K. Wang, X. Luo, X. Li, Y. Ni, and G. Xie. Panlp at mediqa 2019: Pre-trained language models, transfer learning and knowledge distillation. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 380–388, 2019.