Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Tomasz Jurczyk Date

Improving Question Answering by Bridging Linguistic Structures and Statistical Learning

By

Tomasz Jurczyk
Doctor of Philosophy

Mathematics and Computer Science

Jinho D. Choi, Ph.D.
Advisor

Eugene Agichtein, Ph.D.
Committee Member

Li Xiong, Ph.D.
Committee Member

Marsal Gavalda, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the Graduate School

Date

Improving Question Answering by Bridging Linguistic Structures and Statistical Learning

By

Tomasz Jurczyk
M.S., Wroclaw University of Technology, 2011

Advisor: Jinho D. Choi, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2017

Abstract

Improving Question Answering by Bridging Linguistic Structures and Statistical Learning
By Tomasz Jurczyk

Question answering (QA) has lately gained lots of interest from both academic and industrial research. No matter the question, search engine users expect the machines to provide answers instantaneously, even without searching through relevant websites. While a significant portion of these questions ask for concise and well known facts, more complex questions do exist and they often require dedicated approaches to provide robust and accurate systems.

This thesis explores linguistically-oriented approaches for both factoid and non-factoid question answering and cross-genre text applications The contributions include new annotation schemes for question answering oriented corpora, extracting linguistic structures and performing matching, and early exploration of conversation dialog text applications.

For sentence-based factoid question answering, a multi-stage crowdsourcing annotation scheme is presented. Next, a subtree matching algorithm for two sentences that aims to extract semantic similarity in open-domain texts is introduced and combined with a neural network architecture. Then, various factoid question answering corpora are thoroughly analyzed and cross-tested to improve the performance of QA systems. This thesis explores two complex scenarios of non-factoid question answering. In the first, a semantics-graph knowledge graph that is build on the top of linguistic structures is presented and applied on arithmetic questions using verb polarity classification. In the second, a system that combines lexical, syntactic and semantic text representations with statistical learning is presented and evaluated on event-based question answering. The last part of this thesis is focused on the cross-genre aspect of text in which the misalignment between the dialog and formal writings is the main challenge. First, an approach that combines semantic structure extraction with statistical learning is presented and used to improve the performance in the document retrieval task. Next, an exploration for the passage completion task is presented. A crowdsourcing annotation scheme is executed and a new corpus is created. A multi-gram convolutional neural network with the attention is compared to several state-of-the-art approaches for reading comprehension applications.

Improving Question Answering by Bridging Linguistic Structures and Statistical Learning

By

Tomasz Jurczyk
M.S., Wroclaw University of Technology, 2011

Advisor: Jinho D. Choi, Ph.D.

A dissertation submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2017

Acknowledgments

First, I would like to thank my advisor, Jinho D. Choi. I met Jinho during my third year at Emory University. His passion and insight were motivating me to become a better researcher. I'm thankful for countless discussions which we have had, which helped me to find the direction of my research. Jinho encouraged me to participate in multiple conferences and to seek industrial opportunities. I thank Jinho for working with me days and nights to improve my publications.

I would like to thank to my thesis committee members, Prof. Eugene Agichtein, Prof. Li Xiong, and Dr. Marsal Gavalda. Their comments and suggestions helped me to improve the work in this thesis. Thanks to Dr. Marsal Gavalda for coming from San Francisco to attend my defense.

Thanks to the friends and collaborators from the NLP Group at Emory University: Henry Chen, Gary Lai, Timothy Lee, Kaixin Ma, Allen Nie, Bonggun Shin, Sayyed Zahiri, and Michael Zhai. I also would like to thank to all of my friends in the Department of Mathematics and Computer Science at Emory University: Aji Abulimiti, Ali Ahmadvand, Sapoonjyoti DuttaDuwarah, Daniel Garcia ulloa, Reza Karimi, Ameen Kazerouni, Pooya Mobadersany, Derek Onken, Denis Savenkow, Mani Sotoodeh, Farnaz Tahmasebian, and Safoora Yousefi. I want to thank them for all the insightful discussions which we have had during my 5.5 years at Emory University.

To my wife Magdalena

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

For decades, humans have dreamed of constructing a system that could maintain a natural language conversation and answer questions. The first generation of question answering systems from the early 1960s was based on the simple paradigm that every combination of characters and words can be represented by some Boolean query [129]. Systems designed during these times often lacked the ability to perform any logical reasoning or advanced semantic analysis. One of the first attempts in developing semantically advanced systems was the *ELIZA* project in the 1960s. It was capable of providing answers to single questions using the pattern-matching techniques [55]. The most famous chatbot implemented during the *ELIZA* period was *DOCTOR* that is a simulation of the person-centered therapy that formed an illusion that a machine is capable of engaging in discourse. Despite the great progress made by the project, it has never reached a step where it could perform rea-

soning from the text.

With the spike of natural language processing and machine learning, text applications have become a popular trend among researchers. Currently, all major search engines try to predict the user's intent from the query and if possible, they serve the answer content right away. In order to provide meaningful answers, the models must be capable of recognizing the type of question, expected answer or even performing an inference.

Despite the recent progress in the field of question answering, there is still an increasing need to build systems that would comprehend natural language. More recently, conversational assistant approaches have emerged as the potential breakthrough in question answering enabled systems. Unfortunately, such systems often require a deep and contextual analysis to understand an abstract representation given any text.

1.2 Research Challenges

Question answering is defined as a field where the main challenge consists of providing automatic answers to questions posed by humans in natural language. In order to develop a system capable of performing such actions, the system has to first understand the logic behind reasoning, understanding and extracting information from text.

Open-domain *factoid* question answering consists of questions regarding well-known and concise facts. Consider the question *"When was Barack Obama born?*, for which the answer is *August 4th, 1961.* Current systems are able to provide answers to such questions using already existing knowledge

graphs. Such extraction is rather a straightforward process, in which the relation is extracted first, and then it is matched against the structure of the graph.

On the other hand, a large number of questions asked today on search engines, approximately 70% of them[1], still require users to perform a manual search through provided search results. Consider the question *Who lead the polish army in the Siege of Warsaw?*, for which the answer is *Walerian Czuma*. The information that supports this query can be directly extracted from one of the Wikipedia pages[2]: *The siege lasted until September 28, when the Polish garrison, commanded under General Walerian Czuma, officially capitulated*. Unfortunately, lexicon-based approaches would likely fail in locating the correct sentence among the ones from the Wikipedia page. The entire article is highly correlated with the words from the query, therefore more advanced syntactic and semantic analysis is needed.

Solving more complex factoid questions is a great challenge that is often approached by designing and training statistical models. Unfortunately, more advanced the model is, it requires a vast amount of data to be trained. The source of such data could be search engines with their click data. Sadly, this type of information is almost impossible to access by researchers. Therefore, the manual creation process has to be developed to generate diverse, challenging and realistic datasets for machine learning models.

Question answering is one of the most unsolved problems in natural language processing and has already attracted a large number of researchers.

[1] `https://www.act-on.com/blog/how-and-why-to-set-your-site-up-for-googles-rich-answers/`

[2] `https://en.wikipedia.org/wiki/Siege_of_Warsaw_(1939)`

In recent times, several corpora for various tasks have been published. It is reasonable to look how these corpora differ and how they could be combined to improve the quality and performance of question answering systems. The main reason to combine the corpora is that it increases the amount of training data for statistical models. On the other hand, the combination of various datasets increases diversity which possibly can reduce the amount of data that is enough to train robust models.

Questions that do not fall into the factoid questions category are often classified as *non-factoid* questions. A significant portion of them is lexically and semantically complex in a way that they require more advanced systems compared to the factoid question answering systems. An example of a non-factoid question can be: *how to boil rice?*. While many potential answers to such question exist, the task is to extract the most precise and relevant one. *Arithmetic questions* are another example of the non-factoid, complex branch that requires the abstract understanding of the text. An answer to these questions is impossible to be extracted from a single sentence and the answer is rarely mentioned in the text. In arithmetic questions, the system must first obtain the meaning representation from given text which is later used to extract the actual answer from. Unfortunately, computer systems are not primarily designed to infer meanings from natural language text. Therefore, due to the structural uniqueness and abstract complexity of the complex non-factoid questions, the approaches that worked for factoid questions, would likely fail for non-factoid.

On the other hand, *event-based* questions pose a different challenge. Consider a set of sequential sentences describing a simulation of characters and

objects that are moving within various locations. Then, given this sequence and a question, the task is to locate the context followed by extracting the actual answer from it. Already existing lexicon-oriented systems would perform poorly due to the fact that these sentences are lexically similar to each other. Moreover, the system must perform reasoning and semantic understanding of natural language such as basic deduction or yes/no questions.

The source data that the answer to the question is supposed to be answered from is not always of the same genre. While the *cross-genre* aspect has been well studied in the past for various text applications such as dialog summarization, several challenges still need to be addressed for the question answering domain. Consider a natural language fact or question regarding a popular TV show and a set of scripts from all episodes. In the *cross-genre document retrieval* task, the correct episode that describes the events from given query must be located. Multiple dialogue disfluencies and the rhetorical devices such as irony or sarcasm make this task extremely difficult. On the other hand, in order to perform question answering in the cross-genre setup, it is important and necessary to perform the actual text comprehension of the human dialogue. *Passage completion* is a task in which given a dialogue and a query with masked entities such as persons, organizations *etc.*, the system must recognize which entity from the dialogue should be placed on the masking position. To achieve this goal, the semantic understanding of the human dialog will be crucial. Also, systems designed for this task will have to address the misalignment between various text genres.

1.3 Contributions

This thesis explores various methods and techniques dedicated to improving the performance of question answering systems. Its focus is on presenting methods for crowdsourcing data generation and combining linguistic structures to extract meaning representations. Moreover, the dissertation provides methods dedicated to the non-factoid aspect of question answering. The lexical and semantic complexity in these questions is often significantly higher than in the factoid questions. Finally, this thesis examines the natural language aspect focused on conversational data. Two tasks are presented: document retrieval and passage completion. New methods are introduced that set a solid groundwork in this area. The work in this thesis consists of the following contributions:

- **A crowdsourcing annotation framework for generating question answering corpora**: This thesis introduces a scalable, multi-stage crowdsourcing annotation scheme for creating open-domain corpora that are both diverse and challenging. The scheme consists of four annotation tasks that are performed in sequence and the fifth automatic task that is designed to generate an answering triggering corpus. This annotation scheme is one of the first works designed to provide a framework for providing a scalable and challenging open-domain corpus with low-cost. As a proof of concept, the framework is executed on one of the crowdsourcing services and results in the new dataset for open-domain, sentence-based question answering called SelQA. The cross-evaluation with already existing corpora shows the power of the

corpus as the universal and generic benchmark for sentence-based question answering systems.

- **A subtree matching mechanism for measuring the contextual similarity between two sentences**: The dissertation develops an algorithmic approach for extracting semantic similarity between two sentences. The designed algorithm, unlike the previous tree-edit approaches, is based on the slice-by-slice tree matchings and thus is computationally less expensive. Moreover, the algorithm supports the matching that is performed using the distributional semantics models. Features based on extracted similarities are used with statistical learning, which shows a significant improvement in the sentence-based question answering and sets a new state of the art for answer triggering.

- **Exploration of neural network architectures for open-domain question answering**: This thesis investigates various deep learning approaches for handling sentence classification in the sentence-based question answering tasks. The convolutional neural architectures presented in this thesis as one of the first attempts of building a convolutional neural network for sentence-based question answering paved the way for future advancements to deep learning architectures in this area. Researchers have already started extending this framework and providing another improvements in this area. Also, the developed modifications in the convolutional neural network provide a new state-of-the-art network architecture for the answer sentence selection task.

- **Cross-evaluation of multiple sentence-based question answer-**

ing corpora: The thesis presents intrinsic and extrinsic analyses of the four latest corpora that are based on Wikipedia. The presented work is a major contribution to the cross-use of independent question answering corpora. The analysis provides essential knowledge of how these corpora differ. Also, the work in this dissertation is the first step towards applying the transfer learning methods for independent sentence-based corpora. The experiments show that the combination of datasets can significantly reduce the amount of training data that is required to train statistical models to achieve similar results.

- **A semantics-based graph approach designed for non-factoid question answering**: This thesis develops a semantics-based graph approach to complex question answering. The approach is evaluated using a publicly available dataset of arithmetic questions. The graph presented in this thesis is one of the first attempts to build an abstract representation of natural text using already existing natural language processing tasks. This graph does not require an external annotation, unlike other approaches. The evaluation dataset consists of math problems that are dedicated to elementary and middle school children. By using the grounded knowledge included in the constructed graph, the system shows a promising result and proves to be an effective approach to construct meaning representations of the text that can be used to solve complex questions.

- **Multi-field structural decomposition for question answering**: The dissertation develops a precursory yet novel approach to the ques-

tion answering task using structural decomposition. Documents are decomposed into multiple fields that are grouped in three categories: lexical, syntactic and semantic. The questions are decomposed in the same way as documents. Unlike recent approaches that often consist mostly of neural architectures, it tries to provide an easy to comprehend representations of natural text. The final model gives an absolute improvement of over 40% from the baseline approach that uses a simple search for detecting documents containing answers. The evaluation proves that the decomposition presented in this thesis can be used to successfully solve non-factoid questions.

- **The cross-genre document retrieval task for conversational and formal writings**: Motivated by the recent spike of conversational agents and personal assistants, this thesis provides early solutions to several challenges in the cross-genre aspect. The dissertation introduces a structural approach for improving the document retrieval task in the case when the source and target texts are of a different type. The developed approach consists of extracting relations from the dialog and formal writings. Then, it matches these relations to improve the initial ranking. A significant performance improvement is observed when the structure matching features are combined with a neural network architecture. The structure extraction presented in this thesis is the first work that addresses matching of the conversational and formal writings. It paves the way for the future developments of systems that will have to understand and extract the knowledge from the human conversation and then match it against the information that comes from

formal writings.

- **Exploration of the passage completion task in cross-genre text domains**: This thesis explores the task of passage completion where the challenge involves predicting the masked entity in a query when a dialogue is given. A crowdsourcing annotation scheme is performed first, and a new dataset dedicated to this task is built. A scalable multi-gram convolution neural network with attention is presented and compared to the wide spectrum of already existing approaches. This work sets an early groundwork for the reading comprehension tasks in the conversational and formal text domains. The experimentation and analysis performed in this dissertation address the structure uniqueness of the human dialogue which is the combination of speakers and their spoken utterances. The dataset created during this work is a first publicly available corpus dedicated to the reading comprehension task of multi-party conversations.

This dissertation contributes to the field of question answering from several angles. The experimentation and development of early neural architectures for factoid question answering tasks lay the early groundwork for recent advancements in convolution neural networks. The presented convolutional neural network has already been extended and improved, and the researchers will likely keep developing another improvements. Next, this work results in several corpora released that are dedicated to several aspects of question answering. One of the most recent developments in question answering has been evaluated using the SelQA dataset, and the evaluation has shown the power and uniqueness of the dataset. The researchers have already started

using this source as the benchmark for their selection-based question answering approaches. The thesis also addresses the challenges related to the non-factoid branch in question answering. The work provides an early work on building abstract representations of text which can be extended in the future to handle more non-factoid branches. Finally, motivated by the recent growth of popularity in conversational agents, the work in this thesis contributes to the major challenges in the cross-genre text tasks. The cross-genre document retrieval task is introduced along with the corpus that can be used as a benchmark. The presented structure matching provides a new state-of-the-art approach for this task, leaving room for improvement in future work. The convolutional neural network-based approaches are explored in the passage completion task in the cross-genre domain. This work sets the groundwork for the reading comprehension tasks in cross-genre domain.

Chapter 2

Related Work

This section provides an overview of several topics that are relevant to this dissertation. Question answering emerged in the early 1960s, when one of the first approaches were developed [14, 130, 36]. In the 1970s and 1980s, researchers continued working in this field and started looking into natural language processing aspects that could potentially impact this field [143, 78, 156, 15, 87]. At the end of 20th century, the TREC [1] set of workshops has dominated the field of information retrieval including questions answering [142, 141]. While this workshop has been discontinued in 2007, it established strong baselines in many question answering tasks including answer retrieval [75] and answer extraction [147]. The early approaches to the question answering tasks from the information retrieval perspective were focused on the surface-level information. One group of approaches tried to explore and learn the pattern extraction process automatically and then extract them from retrieved documents [109, 110, 67]. Another group of

[1] Text Retrieval Conference, `http://trec.nist.gov/`

approaches tried to use bag-of-words representations to classify the question-answer pairs [173]. More recently, the TREC family of question answering tasks have divided into several various directions including sentence-based answer sentence selection [149].

Modern question answering covers a wide spectrum of tasks such as reading comprehension [113, 20, 112], passage retrieval [137, 38, 114] or answer extraction [41, 1, 140]. This chapter presents background and related work that is consistent with this thesis and gives the important context to presented contributions.

2.1 Sentence-based Factoid Question Answering

Approaches designed for factoid question answering can be categorized into two groups: *knowledge-based* and *text-based*. The knowledge-based approaches use various techniques to extract relations and entities from the question and then matches these elements with pre-built knowledge bases. Several knowledge bases have been constructed over the last decade: DBPedia [4], Freebase [16], Yago [133], to name a few. Knowledge bases are usually constructed in the entity-relationship manner, providing an easy access and allowing to extract precise information. For instance, consider the question: *"What is LeBron James height?"*. A single entity and relation can be easily extracted and then the answer can be directly extracted from the knowledge base. This extraction process can be performed using two main approaches: *information extraction* and *semantic parsing*. In the former, the task is to locate

all related nodes to the main topic (*LeBron James* in the example above) and their neighbors. Then, each neighbor is binary classified whether it is an answer or not using the combination of various lexical and syntactic features. This approach has been widely used in past work [163, 65, 92]. In *semantic parsing*, lexicons are used to map natural language to the knowledge base predicates. It is followed by the process of combining predicates that will form a logical representation. SEMPRE is a well-known semantic parser which adopts this approach [8]. The knowledge-based question answering is relevant to the work presented in this dissertation although this thesis focuses on the text-based approaches.

2.1.1 Text-based Question Answering Tasks

Knowledge-based approaches are often common choices when the query asks for a specific relation of entities. In the previous subsection, the question asked for the height of LeBron James. In such case, it is relatively easy to extract the entity and its relation, which can be immediately mapped with a knowledge base. Unfortunately, despite the fact that the largest knowledge bases contain millions of nodes and billions of edges, several limitations exist. Often, these approaches fail on more difficult queries or when the query requires more advanced linguistic analysis [69]. Also, knowledge bases sometimes lack the actual entity links which cause problems [174].

Text-based approaches, on the other side, use lexical, syntactic and semantic analyses to perform a set of tasks based on the natural language text. At first, *passage retrieval* is performed, where for a given query and a set of passages (documents), the most relevant ones to the query are located. The

foundation of early approaches in this task is built on the lexicon-based approaches [120, 12, 37]. Salton and Buckley provide a comprehensive overview of term-weighting methods for passage retrieval [120]. More recently, machine learning has been successfully applied in this task by learning to rank the potential candidates [82]. Statistical learning methods such as SVM [18], ListNet [19], decision trees [22], neural networks and genetic algorithms [24] pushed the state of the art forward.

Answer sentence selection is a core task in open-domain question answering. Given a question and a set of candidate sentences, the task is to rank the sentences from the most to least likely containing the answer or supporting it. Early approaches to this task were based on syntactic tree manipulations such as tree edit distances and semantic kernels [104, 38, 149]. These approaches are described further in Subsection 2.1.3. Unfortunately, the tree-edit based approaches have a major limitation of being computationally expensive and are often based on the outputs from specific parsers. Quasi-synchronous grammars [149] and deeper semantic analysis such as semantic roles [125, 58] have also been shown successful in the sentence selection task. Recent growth of neural network architectures and distributional semantics contributed to an increased number of approaches using these methods [171, 68, 43, 168, 166, 13]. Currently, researchers are investigating different attention approaches using neural architectures [160, 136, 138, 135]. Their main goal is to explicitly expose the network to the fragments of text that most likely contain the answer based on the query words. The current state of the art in neural network architectures dedicated to question answering is further described in Section 2.2.

Answer triggering is a more complex version of answer sentence selection in which the model has to first decide whether there is at least a single sentence that supports the query. If there is, it should be returned, otherwise, the model should make a prediction that there is not enough context to support the query. Because of this fact, this scenario is more realistic than answer sentence selection. The initial list of sentences might or might not have a single sentence that is an answer or supports the query. Answer triggering was proposed by Wang et al. [161] along with the first dataset, WikiQA. Since this task is substantially more difficult, neural network architectures combined with manually-crafted lexical features have achieved the highest scores so far [161, 70].

2.1.2 Question Answering Corpora

With the current growth of question answering, several corpora have been published and can be used to evaluate the robustness of question answering systems. Corpus generation has been pioneered by the QASent dataset that consists of 277 questions. It has been widely used for benchmarking the answer sentence selection task [149]. While the corpus has been popular, its extremely small scale makes it impossible to be used in advanced neural architectures. Over the past few years, several more corpora have been released focusing on different aspects of the text-based question answering tasks. InsuranceQA is an automatically generated dataset containing 16K+ questions selected from the insurance library[2] [50]. WikiQA is a dataset com-

[2]http://www.insurancelibrary.com/

prising almost two thousand cleaned Bing[3] search engine queries [161]. These queries have been manually associated with their corresponding Wikipedia pages' abstracts. Morales et al. [93] extracted information boxes from various Wikipedia pages and released the InfoboxQA dataset that consists of over 15K questions with the associated infoboxes. More recently, the SQuAD dataset has been published [106]. It is a massive corpus of over 100K questions designed for the answer extraction task. Majority of these corpora have been created using crowdsourcing methods thus are relevant to this dissertation. Unfortunately, creating a diverse and realistic dataset is still a difficult task for researchers. In this thesis, a new multi-stage annotation scheme is presented that addresses several challenges when building a corpus for open-domain question answering.

2.1.3 Syntactic and Semantic Matching

Text-based methods perform syntactic and semantic matching between natural language texts in several ways. Early tree-based methods for text applications emerged in the early 1980s [134]. This family of approaches is based on a distance paradigm: two syntactic trees are similar to each other if the minimum cost of required sequence modifications (add, delete, change) is relatively small. Punyakanok et al. [104] were one of the firsts who used syntactic tree matching applications to question answering. This idea has been further extended to fuzzy relation matching based on statistical models [38]. Several other extensions to the tree-based approaches have been presented in the current literature [148, 164, 10].

[3]http://www.bing.com/

Tree kernels for syntactic structures have been well studied in the application of question answering. Heilman and Smith [59] developed a heuristic method based on a tree kernel to find shorter and more intuitive sequences. These approaches are often more compact than the initial tree-edit distances, because they are based on prebuilt tree structural cores that define possible tree modifications. Tree kernels have been further researched in [94, 97, 95].

Semantic parsing is another field that has attracted a lot of attention in question answering and has been widely used. Two semantic roles corpora are commonly known: PropBank [102] and FrameNet [5]. After parsing natural language texts and obtaining their semantic annotations, it is easier to understand the semantic meaning of the text. One of the first attempts focused on structured probabilistic inference using predicate-arguments pairs [99]. More recently, Shen et al. have shown significant performance gains when semantic roles are combined with syntax-based systems. [125]. All the aforementioned work describes the background in using the syntactic and semantic information to improve the question answering systems. This dissertation presents several new techniques that are built on the foundation of research in this field and provide significant performance boosts in various tasks.

2.2 Neural Architectures

Statistical learning has been widely used in numerous text applications including question answering. The recent growth of neural architectures has provided several strong and bold improvements to already existing systems.

Convolution neural network along with distributional semantics has been widely adopted in the question answering field. A bi-gram convolution model with logistic regression and a few manually crafted lexical features was successfully applied in answer sentence selection [171] and in answer triggering [161, 70]. Yin et al. developed a hierarchical convolution neural network without any manually crafted features and successfully applied it to the machine comprehension task [167]. More configurations of convolutional neural network have been developed for answer sentence selection [123, 73, 168, 139] and machine comprehension [167], to name a few.

On the other side, the main goal of recurrent neural networks is to extract encoded representation given a sequence of natural language text. Wang and Nyberg presented an approach of stacked bi-directional long-short term memory network that sequentially reads words from a question and candidate answer and then calculates their relevance score [145]. Two direction pass lets the network preserve information from past to future, and vice versa. When more than a single layer of a recurrent cell is used, the output of one becomes the input to another layer which helps the network to catch hidden sequential information from the input. The long-short term memory based approaches have been widely adopted in answer sentence selection [45, 138, 107, 70] and machine comprehension [60, 23, 127, 122, 100, 27]

More recently, various attention mechanisms have emerged as a new trend in the neural network architectures designed for text applications. This idea has originated from the image processing field [86, 172, 77], but recently have been found useful in the text domain as well. In convolutional neural networks, one of a few approaches consists of developing an attention matrix

and applying it to the dot product of feature maps [45]. As the result, two vectors are computed that represents the importance of each word from candidate answer with respect to given query, and vice versa. Yin et al. proposed an extension where the network receives an additional input that consists of word similarities from the query and candidate sentence [168]. This input is then used to perform an attention mechanism by incorporating it with the feature maps. Attention in convolutional neural networks has been further studied in the recent years [162, 25]. Similar to convolutional neural networks, the same paradigm was found useful in recurrent neural networks, especially in machine comprehension tasks [122, 150, 151, 39].

Neural architectures have been proven effective in the question answering field. In this dissertation, one of the earliest applications of convolutional neural network is extended and paired with additional hand-crafted features that leads to a new state-of-the-art result in answer triggering.

2.3 Non-factoid Question Answering

Non-factoid question answering is an umbrella term for almost all questions that cannot be classified as factoid. Several tasks have emerged in this area: community question answering [98], real-time question answering [81], visual question answering [2], math word and logic questions (arithmetic) [76], event-based question answering [154], and more. This dissertation presents two approaches that are evaluated in two areas of non-factoid question answering: arithmetic questions and event-based questions.

Arithmetic Question Answering

Math and logic problems primarily designed for elementary and middle school students are a good example of an abstract problem that requires context understanding and performing logical operations. The model has to infer the actual question intent, merge multiple layers of information and provide an answer. Consider this example: *Sara had 7 red and 8 blue balls. She received 3 red balls and gave away 2 blue balls. How many blue balls does Sara have?*. The answer to this question (*6 blue balls*) is not mentioned in the context. Therefore, the algorithm has to reason the answer given the context rather than extracting it from the text.

Recently, these problems have gained lots of interest among researchers. Kushman et al. collected a set of algebra problems from a crowdsourced tutoring website[4] and published a dataset comprising 1024 algebraic questions with their linear equations and answers [76]. Researchers partnered with Allen Institute for Artificial Intelligence[5] have released several corpora for math and logical problems[6]: a set of life science, 6,952 real science exam questions, 100 geometry questions, a set of arithmetic questions, and many more. This dissertation explores the last set, which is focused on the arithmetic problems.

Deep semantic understanding of human language and text comprehension is required to solve an arithmetic question. While lexical approaches have shown their potential for different factoid question tasks, it is likely they would fail here. A state transition method designed to map natural language

[4] `www.algebra.com`

[5] `http://allenai.org/`

[6] `http://allenai.org/data.html`

to an equation based on predefined templates was one of the first approaches designed for this problem [76]. This approach has been later extended to learning to categorize verbs [66]. More recently, algorithmic approaches have been proposed that try to directly build the equations based on tree and graph structures [118, 119]

Several approaches to build generic semantic representations of text have already been proposed. Abstract Meaning Representation (AMR)[7] [7] is an attempt to generate a semantic graph of meaning based on syntactic and semantic structures. Wong et al. proposed a novel statistical approach based on a syntax-based translation model that constructs a meaning representation of a sentence [157]. While some of these representations have been applied to text problems [42, 108], most of the today's work in this field focuses on building robust and reliable parsers. This dissertation presents a novel approach to create a semantics-based graph approach that is constructed on top of the natural language processing tasks.

Event-based Question Answering

Question answering often requires text understanding in several ways: deduction, multiple supporting facts from different sentences, coreference resolution, *etc.* Weston et al. released a dataset called bAbI[8] that consists of a set of proxy tasks that are focused on the natural text understanding [154]. The dataset consists of twenty artificially-crafted small tasks. The goal is to select a supporting sentence and to extract an answer given context.

[7] https://amr.isi.edu/

[8] https://research.fb.com/downloads/babi/

There have already been several approaches proposed for tasks that are related to the event-based type of questions. Pizzato et al. proposed a question prediction language model based on indexing of semantic roles and achieved a promising result [103]. Athira et al. presented a modular architecture that consists of four basic modules [3]. Ontology-based domain knowledge is used to reformulate questions and identify relations. More syntactic and semantic-based approaches that have been developed can be found in [11, 105, 83, 49] More recently, IBM has released Watson[9], a hybrid approach between NLP and IR [52] that has significantly advanced the field of question answering.

This thesis presents a novel approach to combine lexical, syntactic and semantic features that are extracted automatically given a corpus of text.

2.4 Applications to Cross-genre Tasks

Due to the spike of applications that are required to maintain the conversation, dialog data has recently become a popular target among researchers. The work in this field concerns problems such as learning facts through conversation [51, 155, 62] or dialog summarization [101, 91]. More recent work in this field has focused on several inter-dialogue tasks [158, 72, 57]. Several corpora in the conversation domain have already been released. *Cornell Movie Dialog Corpus* contains a large collection of fictional conversations that have been extracted from movie scripts [40]. *The Ubuntu Dialogue Corpus* is a large dataset comprising 1M multi-turn dialogues with a massive number of 7M utterances [84]. *The Character Mining* project provides transcripts of all

[9] `https://www.ibm.com/watson/`

ten seasons of the TV show, *Friends*. [26].

Document retrieval has been a central task in natural language processing and information retrieval for a long time. The goal is to match a query against a set of documents. Previous work provided strong baselines for unstructured text retrieval and ranking problems [12, 17, 18, 6, 159, 175]. However, these systems usually assume a homogeneous domain for queries and target documents. Information extraction for the dialogue data has also been explored. Yoshino et al. presented a spoken dialogue system that extracts predicate-argument structures and uses them to extract facts from news documents [170]. Flycht-Eriksson et al. developed a dialog interaction process of accessing textual data from a bird encyclopedia [53]. An unsupervised technique for a meeting summarization using decision-related utterances has been presented by Wang et al. [146]. Gorinski and Lapata studied movie script summarization [54]. Rosset et al. developed the RITEL PROJECT, a system designed to integrate spoken language and open-domain information retrieval [117]. All the aforementioned work uses the syntactic and semantic relation extraction in conversation domain and is relevant to this thesis. This dissertation lays the early groundwork for the document retrieval task that is focused on conversation and formal writings.

Passage completion can be defined as a subtask in machine comprehension. Given a passage of text with a set of entities and a query with the masked placeholder, the goal is to select the entity from the passage that matches the query context. The CNN/DAILY NEWS dataset is a massive data set that consists of over 1M queries with their corresponding passages [60] and is dedicated for this task. Existing approaches to this task consist mainly

of memory networks [61], recurrent neural networks [60] and the attention-based models [71, 39, 132]. This thesis explores the passage completion task in when the cross-genre aspect of text domains must be addressed.

Chapter 3

Sentence-based Factoid Question Answering

Factoid question answering is a field that consists of any questions regarding well-known and concise facts. Two main directions have emerged in this field: knowledge-based and text-based approaches. Usually, the former is applied to the questions that do not pose a significant syntactic or semantic challenge. The latter concerns approaches focused on extracting semantic similarity between any open-domain texts.

This chapter explores the tasks and methods that are related to the text-based methods. More precisely, it is focused on the sentence-based approaches, in which a single sentence is considered an answer to a question. It is different from the TREC evaluation where the answer key must be returned with the supporting document ID. However, the sentence-based aspect does not simplify the task [104]. It has been shown that the hardest part of the answer selection tasks is to find the correct supporting document. The an-

swer key may be extracted using the heuristic rules in the last step. Also, it is reasonable to provide a sentence to a user who will be able to extract the answer having the selected sentence. Moreover, when a user is provided with a sentence, it has better resolution of the context to the question.

Factoid questions often have only one correct answer unlike several non-factoid types such as recommendation questions. Therefore, it is reasonable to perform text-based techniques on large unstructured collections of information such as Wikipedia. One might consider a question: *"Who lead the polish army during the Siege of Warsaw?"* and two different sentences: *"The siege lasted until September 28, when the Polish garrison, commanded under General Walerian Czuma, officially capitulated"* and *"Johannes Blaskowitz was a commander of the German army who were invaders, while the defending army was lead by Walerian Czuma"*. The first sentence does not mention explicitly the main object of this question (*"The Siege of Warsaw"*) and uses a synonym of *"garrison"* instead of the word *"army"*. The second sentence requires more semantic understanding. The word *"defenders"* with respect to the German invaders is the Polish army. Most of the knowledge base approaches would likely fail in this question. Despite the size and impact of the knowledge bases, they often lack some entities. More importantly, even if the knowledge bases are extended, it becomes extremely difficult to scale up the matching process [153]. On the other hand, a text-based approach would likely work for this question. Semantic similarity can be applied as a way to resolve semantic relatedness [111]. In the example above, a model should be able to recognize and properly measure the relatedness of the words: *army*, *garrison*, *commanded*, *lead*, and *commander*. Obviously, such

matching must be performed beyond the lexical level. More precisely, the syntactic and semantic structures of given question and sentence candidates must be measured and compared.

Researchers from the distributional semantics field have recently developed several techniques to represent the human language in a dense dimension. Thanks to this procedure, it is now possible to apply a wider spectrum of statistical learning approaches to the human language-related problems. One of the approaches is called WORD2VEC and has been broadly used [89]. The WORD2VEC model learns the dense representation that will locate the word vectors closer to each other if they often appear in the same contexts. For instance, a word *dog* will be closer to *cat* and *leash* than to *classroom*. The WORD2VEC model has been successfully applied to several text applications: sentiment analysis [44, 73], text summarization [35, 131], machine translation [28, 85], and of course question answering [25, 68]. The distributional semantics models are therefore useful in approaches which try to extract semantic similarity from natural language. Neural networks are common choices for these systems. Unfortunately, these architectures often require a huge amount of data to train meaningful models.

Researchers have recently published multiple corpora that can be used to benchmark the question answering systems. Their main goal is to evaluate how well the model does with recognizing the contextual similarity of open-domain texts. Unfortunately, these datasets often come from different sources and are generated using different tools. Also, these datasets might concern different question answering tasks. Therefore, it is challenging to train a statistical model using the combination of all datasets. However, it is likely

that this combination would improve the performance of question answering system by increasing the diversity and uniqueness of the training data.

In this chapter, multiple contributions to the factoid question answering field are presented. First, a multi-stage annotation scheme is described that provides researchers a framework to build diverse and challenging open-domain question answering corpora (Section 3.1). The designed framework uses crowdsourcing to generate, paraphrase and clean questions based on any text domains. Unlike other datasets, this framework allows researchers to create a corpus designed for passage retrieval, answer sentence selection, answer triggering and answer extraction. Next, Section 3.2 introduces a novel subtree matching algorithm based on syntactic structures. Unlike previous approaches, this algorithm has a configuration to be used with the distributional semantics models such as WORD2VEC as similarity measures. The developed algorithm is next paired with a convolutional neural network and logistic regression. The evaluation shows a significant boost in answer sentence selection and sets the state-of-the-art performance for answer triggering. Finally, Section 3.3 explores recently published question answering corpora and compare them intrinsically and extrinsically. Cross-evaluation shows that similar system performance can be achieved by using a limited amount of data for training.

Overall, the work discussed in this chapter provides a novel framework to build diverse corpora and extends the current state of the art in the field. Also, a thorough analysis of already existing corpora is presented which helps to understand their characteristics and use them better in the future work.

Type	Count
Total # of articles	486
Total # of sections	8,481
Total # of sentences	113,709
Total # of tokens	2,810,228

Table 3.1: Lexical statistics of the collected Wikipedia articles corpus.

3.1 Multi-stage Annotation Scheme for Question Answering

Researchers usually do not have access to large collections of data that would let them generate datasets on a large scale. Fortunately, with the current state of the art of crowdsourcing techniques, this goal becomes achievable. Presented annotation scheme provides a framework for any researcher to create a large, diverse, pragmatic, and challenging dataset for answer sentence selection and answer triggering while maintaining a low cost using crowdsourcing.

3.1.1 Data Collection

A total of 486 articles are uniformly sampled from the following 10 topics of the English Wikipedia, dumped on August, 2014:

Arts, *Country*, *Food*, *Historical Events*,
Movies, *Music*, *Science*, *Sports*, *Travel*, *TV*.

These are the most prevalent topics categorized by DBPedia.[1] The original data is preprocessed into smaller chunks. First, each article is divided into sections using the section boundaries provided in the original dump.[2] Each section is segmented into sentences by the open-source toolkit, NLP4J.[3] In the corpus, documents refer to individual sections in the Wikipedia articles. Table 3.1 presents a lexical analysis of the collected data.

3.1.2 Annotation Scheme

Four annotation tasks are conducted in sequence on Amazon Mechanical Turk for answer sentence selection (Tasks 1-4), and a single task is conducted for answer triggering using only Elasticsearch (Task 5; see Figure 3.1 for the overview).

Task 1

Approximately two thousand sections are randomly selected from the 486 articles in Section 3.1.1. All the selected sections consist of 3 to 25 sentences; is has been empirically found that annotators experienced difficulties accurately and timely annotating longer sections. For each section, annotators are instructed to generate a question that can be answered in one or more sentences in the provided section, and select the corresponding sentence or sentences that answer the question. The annotators are provided with the instructions, the topic, the article title, the section title, and the list of num-

[1] `http://dbpedia.org`
[2] `https://dumps.wikimedia.org/enwiki`
[3] `https://github.com/emorynlp/nlp4j`

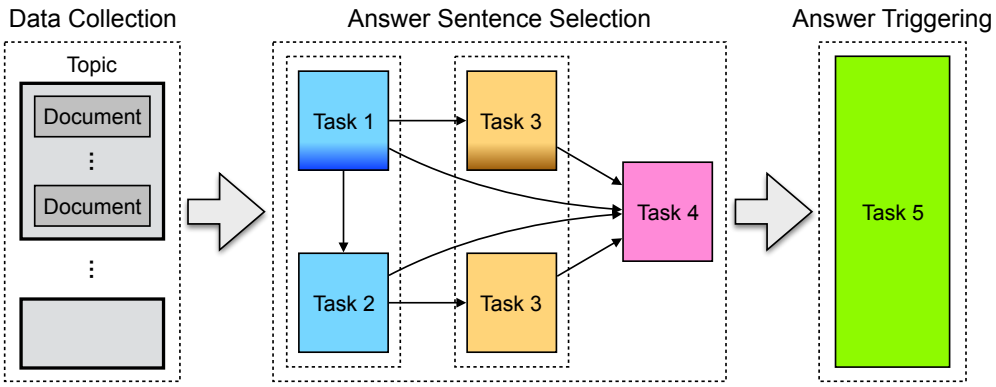

Figure 3.1: The overview of the data collection (Section 3.1.1) and annotation scheme (Section 3.1.2).

bered sentences in the section (Table 3.2).

Task 2

Annotators are asked to create another set of ≈2K questions from the same selected sections excluding the sentences selected as answers in Task 1. The goal of Task 2 is to generate questions that can be answered from sentences different from those used to answer questions generated in the Task 1. The annotators are provided with the same information as in Task 1, except that the sentences used as the answer contexts in Task 1 are crossed out (line 1 in Table 3.2). Annotators are instructed not to use these sentences to generate new questions.

Task 3

Although the annotation instruction encourages the annotators to create questions in their own words, annotators will generate questions with some

lexical overlap with the corresponding contexts. The intention of this task is to mitigate the effects of annotators' tendency to generating questions with similar vocabulary and phrasing to answer contexts. This is a necessary step in creating a corpus that evaluates reading comprehension rather than the ability to model word co-occurrences. The annotators are provided with the previously generated questions and answer contexts and are instructed to paraphrase these questions using different terms.

Task 4

Most questions generated by Tasks 1-3 are of high quality, that is they can be answered by a human when given the corresponding contexts; however, there are some questions that are ambiguous in meaning and difficult for humans to answer correctly. These difficult questions often incorrectly assume that the related sections are provided with the questions, which cannot be assumed in reality. For instance, it is impossible to answer the question from Task 3.1 in Table 3.2 unless the related section is provided with the question. These ambiguous questions are sent back to the annotators for revision.

Elasticsearch is used to find ambiguous questions,[4] a Lucene-based open-source search engine. First, an inverted index of 8,481 sections is built, where each section is considered a document. Each question is queried to this search engine. If the answer context is not included within the top 5 sections in the search result, the question is considered 'suspicious' although it may not be ambiguous. Among 7,904 questions generated by Tasks 1-3, 1,338 of them are found to be suspicious. These questions are sent to the annotators, and

[4] `www.elastic.co/products/elasticsearch`

Topic: TV, **Article**: Criminal Minds, **Section**: Critical reception

1. The premiere episode was met with mixed reviews, receiving
   a score of 42 out of 100 on aggregate review site Metacritic,
   indicating "mixed or average" reviews.
2. Dorothy Rabinowitz said, in her review for the Wall Street Journal,
   that "From the evidence of the first few episodes,
   Criminal Minds may be a hit, and deservedly"...
3. The New York Times was less than positive,
   saying "The problem with *Criminal Minds* is its many confusing maladies,
   applied to too many characters" and felt that "as a result,
   the cast seems like a spilled trunk of broken toys, with which
   the audience - and perhaps the creators - may quickly become bored."
4. The Chicago Tribune reviewer, Sid Smith, felt that the show
   "May well be worth a look" though he too criticized...

Task 1	How was the premiere reviewed?
Task 2	Who felt that Criminal Minds had confusing characters?
Task 3.1	How were the initial reviews?
Task 3.2	Who was confused by characters on Criminal Minds?
Task 4.3.1	How were the initial reviews in Criminal Minds?

Table 3.2: Given a section, Task 1 asks to generate a question regarding to the section. Task 2 crosses out the sentence(s) related to the first question (line 1), and asks to generate another question. Task 3 asks to paraphrase the first two questions. Finally, Task 4 asks to rephrase ambiguous questions.

rephrased by the annotators if deemed necessary.

Task 5

By using the previously generated answer sentence selection data, the answer triggering corpus can be automatically generated again using Elasticsearch. To generate answer contexts for answer triggering, all 14M sections from the entire English Wikipedia are indexed, and each question from Tasks 1-4 is queried. Every sentence in the top 5 highest scoring sections from Elasticsearch are collected as candidates, which may or may not include the answer context that resolves the question. Default Elasticsearch configuration is followed with tf-idf as a similarity measurement and relevance scoring. Stopwords are excluded from indexing, and BM25 [116] is used as a similarity measurement. This Lucene approach gives an efficient way of generating high-quality annotation.

3.1.3 Corpus Analysis

The entire annotation took about 130 hours, costing $770 in total; each mechanical turk job took on average approximately 1 minute and cost about ¢10. A total of 7,904 questions were generated from Tasks 1-4, where 92.2% of them found their answers in single sentences. It is clear that Task 3 was effective in reducing the percentage of overlapping words between question and answer pairs (about 4%; Ω_f in Table 3.3). The questions from Task 3 can be used to develop paraphrasing models as well, which makes the annotation scheme even more attractive to researchers. Multiple pilot studies on different tasks were conducted to analyze quality and cost and to find

	$\mathbf{Q_s}$	$\mathbf{Q_m}$	$\mathbf{Q_{s+m}}$	$\mathbf{\Omega_q}$	$\mathbf{\Omega_a}$	$\mathbf{\Omega_f}$	**Time**	**Credit**
Task 1	1,824	154	1,978	44.99	23.65	28.88	71 sec.	$ 0.10
Task 2	1,828	148	1,976	44.64	23.20	28.62	64 sec.	$ 0.10
Task 3	3,637	313	3,950	38.03	19.99	24.41	41 sec.	$ 0.08
Task 4	682	55	737	31.09	19.41	21.88	54 sec.	$ 0.08
This corpus	7,289	615	**7,904**	40.54	21.51	**26.18**	-	-
WikiQA	1,068	174	1,242	39.31	9.82	15.03	-	-

Table 3.3: $Q_{s|m}$: number of questions whose answer contexts consist of single|multiple sentences, $\Omega_{q|a}$: macro avg. of overlapping words between q and a, normalized by the length of $q|a$, $\Omega_f = {}^{(2\cdot\Omega_q\cdot\Omega_a)}/_{(\Omega_q+\Omega_a)}$, Time|Credit: avg. time|credit per mturk job. WikiQA statistics here discard questions w/o answer contexts.

the most effective and accurate annotation scheme; Tasks 1-4 were proved to be the most effective in the pilot studies. Annotators who submitted outstanding work were paid incentives, which improved the overall quality of the annotation. It has been previously shown that by incentivizing the best performing annotators, the overall quality of annotation is significantly higher[64], which has also been observed in this annotation scheme.

The newly created corpus called SelQA could be compared to WikiQA that was created with the intent of providing a challenging dataset for selection-based question answering [161]. Questions in this dataset were collected from the user logs of the Bing search engine, and associated with the specific sections in Wikipedia, namely the first sections known as the abstracts. The goal of this annotation is to provide a similar yet more ex-

haustive dataset by broadening the scope to all sections from articles. Considering a larger context adds a layer of complexity for locating the correct section than considering only the abstracts. A notable difference was found between these two corpora for overlapping words (about 11% difference), which was expected due to the artificial question generation in the annotation scheme. Although questions taken from the search queries are more natural in terms of human language, real search queries are inaccessible to most researchers. This issue has been addressed by performing paraphrase step which was aimed to complicate the questions. The new annotation scheme proposed here can prove useful for researchers needing to create a corpus for selection-based QA.

Dataset created as a proof of concept contains 5 times more answer candidates per question than WikiQA because WikiQA includes only sections clicked on by users. Manual selection is eliminated from this framework, making generated corpus more practical since finding the relevant section no longer rely on the user clicks. In WikiQA, 40.76% of the questions have corresponding answer contexts for answer triggering, as compared to 39.25% in SelQA. It shows that while these corpora differ, they still can be combined together to solve the same tasks in open-domain question answering.

3.2 Subtree Matching with Statistical Learning

Sentence-based text-domain question answering is based on the similarity paradigm. It is more likely that a sentence is an answer or supports the

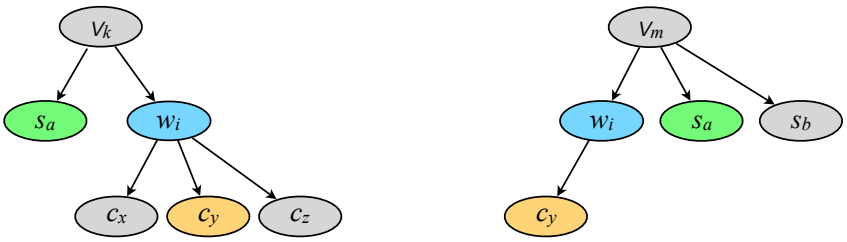

Figure 3.2: Subtree matching between D^q (left) and D^a (right). w_i is the i'th co-occurring word between q and a. The color odes imply 'match', and the grey nodes imply 'non-match'. For instance, v_k in D^q is not matched to any node in D^a, whereas c_y in D^q finds its match in D^a.

question if it is contextually more similar that others. In the beginnings of factoid question answering this paradigm was based mostly on lexical similarity but was later extended to syntactic and semantic measures. This section of this thesis presents a novel subtree matching algorithm for open-domain question answering. Unlike previous approaches, it is easily extendable to include distributional semantics models to perform similarity measurements. Along with the subtree matching mechanism, two models using convolutional neural networks are developed, one is a replication of the best model in [161], and the other is an improved model using subtree matching (Section 3.2.2).

3.2.1 Subtree Matching Algorithm

A subtree matching is now described as a mechanism for measuring the contextual similarity between two sentences. First, all sentences are automatically parsed by the NLP4J dependency parser [30]. A set of co-occurring words between q and a, say T, is created. For each $w_i^o \in T$, w_i^o's parents

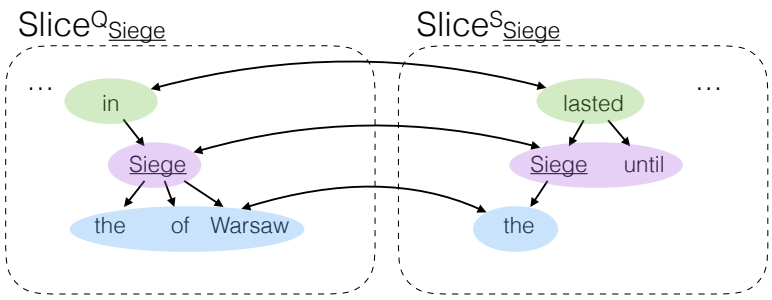

Figure 3.3: The example of subtree matching for a question and candidate sentence. Two overlapping words are first located: *polish* and *siege*, and then their slices are extracted. The matching is performed on the extracted slices.

(p_i^q, p_i^a), siblings (S_i^q, S_i^a), and children (C_i^q, C_i^a) are extracted from the dependency slices of q and a. Then, three matching scores: μ_p, μ_s and μ_c are measured as follows:

$$\mu_p = \sum_{w_i \in T} f_c(p_i^q, p_i^a)$$

$$\mu_s = \sum_{w_i \in T} f_m(\{f_c(x, y) : \forall (x, y) \in S_i^q \times S_i^a\})$$

$$\mu_c = \sum_{w_i \in T} f_m(\{f_c(x, y) : \forall (x, y) \in C_i^q \times C_i^a\})$$

It has been empirically observed that combining these features as a single score lowered the accuracy significantly. The comparator function $f_c(x, y)$ performs a comparison between two tokens. The algorithm currently sup-

Input: T: a set of co-occurring words between a question and answer.

D^q, D^a: sets of slices for a question and answer.

f_m: a metrics function.

f_c: a comparator function.

Output: S_{dep}: A triplet of dependency similarity.

$S_{dep} \leftarrow [0, 0, 0]$;

foreach *word* w_i^o *in* T **do**

- $p_i^q \leftarrow$getParent(D_i^q)
- $p_i^a \leftarrow$getParent(D_i^a)
- $S_{dep}[0] \leftarrow S_{dep}[0] + f_c(p_i^q, p_i^a)$
- $S_i^q \leftarrow$getSiblings(D_i^q)
- $S_i^a \leftarrow$getSiblings(D_i^a)
- $vals \leftarrow []$
- **foreach** *sibling* s_j^q *in* S_i^q **do**
  - **foreach** *sibling* s_k^a *in* S_i^a **do**
    - $vals$.append$(f_c(s_j^q, s_k^a))$
  - **end**
- **end**
- $S_{dep}[1] \leftarrow S_{dep}[1] + f_m(vals)$
- $C_i^q \leftarrow$getChildren(D_i^q)
- $C_i^a \leftarrow$getChildren(D_i^a)
- $vals \leftarrow []$
- **foreach** *child* c_j^q *in* C_i^q **do**
  - **foreach** *child* c_k^a *in* C_i^a **do**
    - $vals$.append$(f_c(c_j^q, c_k^a))$
  - **end**
- **end**
- $S_{dep}[2] \leftarrow S_{dep}[2] + f_m(vals)$

end

Algorithm 1: Algorithm of the subtree matching mechanism

ports three possible comparator functions: *word-form*, *lemma* and *embedding*. When first two are used, the function returns 1 if x and y have the same form; otherwise, it returns 0. When the word embedding is used as the comparator, the function returns the cosine similarity between x and y. The function f_m takes a list of scores and returns either the `sum`, `avg`, or `max` of the scores. Finally, the triplet S_{dep} is used as the additional features to the statistical model. Algorithm 1 presents the entire process in detail. Figure 3.2 and Figure 3.3 present the slicing matching and one of examples, respectively. Although the subtree matching mechanism adds just three more features, the experimentation shows significant performance gains for both the answer sentence selection and answer triggering proving that to solve open-domain question answering problems more effectively, better contextual similarity techniques are required.

The designed algorithm differs from the tree-to-tree and tree-edit-distance approaches proposed in past work [104, 58, 148, 164, 124]. First, it is based on the subtree slices that are extracted based on a heuristic method rather than on the entire tree manipulations Currently, this method uses co-overlapping words between two sentences to locate the slices. Since the algorithm does not perform any transformation nor matching based on the entire trees, it is computationally inexpensive. Next, the algorithm currently supports the WORD2VEC in the matching process. This feature could potentially be used in an attention mechanism in the neural network approach. Also, this algorithm currently does not take dependency relations into account and thus is generic in its nature.

3.2.2 Convolutional Neural Networks

The designed convolutional neural network model is motivated by [161]. First, a convolutional layer is applied to the image of text using the hyperbolic tangent activation function. The image which is an input consists of rows standing for consecutive words in two sentences, the question (q) and the answer candidate (a), where the words are represented by their embeddings [89]. In the experiments, the images of 80 rows (40 for question and answer, respectively) are used. If any of the question or answer is longer than 40 tokens, the rest is being cut from the input. Next, the max pooling is applied to the feature maps and the sentence vectors for q and a are generated. Experimentation with the average pooling as [161] has led a marginally lower accuracy thus the max pooling is used in the final version of the framework. Unlike [161] who performed the dot product between these two vectors, this work adds another hidden layer and learn their weights. The framework supports retraining word embedding to which has not been done previously. Finally, the sigmoid activation function is applied and the entire network is trained using the binary cross-entropy.

Next, a logistic regression model is applied, where the convolution neural network score is used as one of the features. Other features in the logistic regression are the number of overlapping words between q and a, say Ω, Ω normalized by the IDF, and the question length. While the logistic regression model could be merged directly with designed convolutional neural network model, it has been empirically shown that it is more effective to construct this last phase as a separate model [171]. While a neural architecture is extremely useful to extract hidden semantic structure, it lacks the skill to

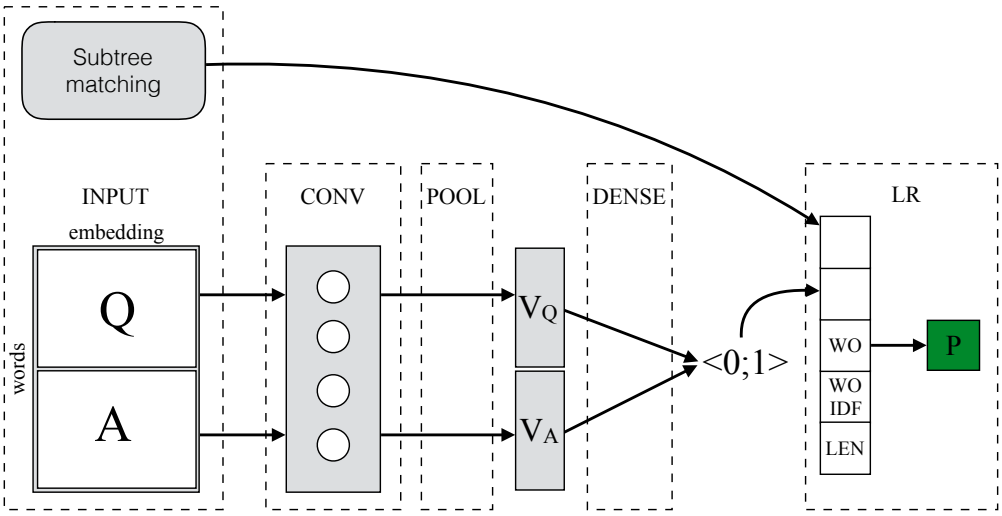

Figure 3.4: The overview of the system that uses a convolutional neural network and logistic regression.

expose a simple lexical matching, which often is an important signal for the final classifier.

The training phase is a separate process for both convolutional neural network model and logistic regression model. First, the convolutional model is trained using early stopping method on development split. After the model is trained, its prediction scores on training, development and test splits are extracted and used in a separate training for logistic regression. For the future predictions, this framework runs as a pipeline performing these steps automatically.

For the answer sentence selection task, the predictions for each question are treated as a ranking; MAP and MRR scores are being calculated and used to evaluate this task. On the other hand, in the answer triggering task, a threshold is applied to each predicted question by the logistic regression;

Set	Q	ASS		AT	
		Sec	Sen	Sec	Sen
TRN	5,529	5,529	66,438	27,645	205,075
DEV	785	785	9,377	3,925	28,798
TST	1,590	1,590	19,435	7,950	59,845

Table 3.4: Distributions of the SelQA corpus. Q/Sec/Sen: number of questions/sections/sentences.

the threshold is trained during logistic regression step on the development split. The candidate with the highest score is considered the answer if it is above the threshold; otherwise, the model assumes no existence of the answer context in this document for that question. It is a crucial difference between the answer sentence selection and answer triggering tasks where this decision is to be made by the model. Figure 3.4 shows the architecture of the entire model that consists of two separate classification algorithms.

3.2.3 Experiments

To perform a thorough analysis of designed techniques, the evaluation is performed on the answer sentence selection and answer triggering tasks on both WikiQA and newly created corpus. Since the SelQA corpus provides an extensive metadata, a thorough error analysis on each system with respect to this corpus is also provided.

Table 3.4 shows the distributions of the SelQA corpus. The dataset is split into training (70%), development (10%), and evaluation (20%) sets.

Model	Development		Evaluation	
	MAP	**MRR**	**MAP**	**MRR**
CNN_0: baseline	69.93	70.66	65.62	66.46
CNN_1: `avg` + word	**70.75**	**71.46**	67.40	69.30
CNN_2: `avg` + emb	69.22	70.18	**68.78**	**70.82**
Yang et al. [161]	-	-	65.20	66.52
Santos et al. [46]	-	-	68.86	69.57
Miao et al. [88]	-	-	68.86	70.69
Yin et al. [169]	-	-	69.21	71.08
Wang et al. [152]	-	-	70.58	72.26
Shen et al. [126]	-	-	71.07	73.04
Wang et al. [144]	-	-	**73.41**	**74.18**

Table 3.5: The answer sentence selection results on the development and evaluation sets of WikiQA.

The answer triggering dataset is significantly larger than the answer sentence selection one, due to the extra sections added by Task 5 (Section 3.1.2).

Answer Sentence Selection

First, the results on answer sentence selection are presented. Table 3.5 shows the results from the previous approaches against the approaches designed in this thesis on the WikiQA dataset. Two metrics are used, mean average precision (MAP) and mean reciprocal rank (MRR), for the evaluation of this task. CNN_0 is the replication of the best model in [161]. CNN_1 and CNN_2

Model	Development		Evaluation	
	MAP	**MRR**	**MAP**	**MRR**
CNN_0: baseline	84.62	85.65	83.20	84.20
CNN_1: `avg` + word	85.04	86.17	84.00	84.94
CNN_2: `avg` + emb	85.70	86.67	**84.66**	**85.68**
Santos et al. [47]	-	-	87.58	88.12
Shen et al. [126]	-	-	**89.14**	**89.93**

Table 3.6: The answer sentence selection results on SelQA.

are the CNN models using the subtree matching mechanism in Section 3.2.2, where the comparator of f_c is either the word form or the word embedding respectively, and f_m = `avg`. The average function is used considering the fact that in the answer sentence selection configuration, there exist at least a single sentence is an answer or supports given the question. Therefore, the given context (a set of candidate sentences) is contextually quite consistent with the question. The experimentation setup shows that the models that use subtree matching method consistently outperform the baseline model. Note that among the three metrics of f_m, `avg`, `sum`, and `max`, `avg` outperformed the others in the experiments for the answer sentence selection task although no significant differences were found.

It is interesting to see how CNN_1 outperforms CNN_2 on the development set, but not on the evaluation set. This result may be explained by the larger percentage of overlapping words in the development set, enabling the simpler models to perform more effectively. More recently, researchers have extended neural architectures based models reaching the MAP scores of 73.41%. The

Topic	**CNN$_0$**	**CNN$_2$**	**Q**
Arts	80.45	82.83	135
Country	**87.12**	**89.03**	178
Food	85.30	86.11	147
H. Events	**91.72**	**92.61**	164
Movies	84.43	86.50	164
Music	81.38	80.39	155
Science	86.37	86.50	179
Sports	81.83	83.69	168
Travel	83.78	86.03	165
TV	77.34	81.23	135

Table 3.7: MRR scores on the SelQA evaluation set for answer sentence selection with respect to topics.

score reported using the network and subtree matching algorithm in this thesis is based on introducing a novel syntactic and semantic context matching, which if extended will possibly reach the score levels of very deep neural networks. Also, it is important to remember that the interpretability of very deep neural networks is difficult although recently used attention models provided a nice platform to extract this kind of information.

To provide a consistent evaluation of the approaches designed in this thesis, the model is now tested on the SelQA data (Table 3.6). CNN$_2$ outperforms the other CNN models, indicating the power of subtree matching coupled with word embedding similarity. Unlike the results on WikiQA in Table 3.5, CNN$_2$ show the best performance on both the development and evaluation sets, implying the robustness and consistency of these models on the SelQA corpus.

Type	**CNN$_0$**	**CNN$_2$**	**Q**
Original	86.70	88.31	810
Paraphrase	81.67	83.00	789

Table 3.8: MRR scores on the SelQA evaluation set for answer sentence selection w.r.t. paraphrasing.

Table 3.7 shows the MRR scores on SelQA with respect to different topics. All models show strength on topics such as 'Country' and 'Historical Events', which is comprehensible since questions in these topics tend to be deterministic. On the other hand, most models show weakness on topics such as 'TV', 'Arts', or 'Music'. This may be due to the fact that not many overlapping words are found between question and answer pairs in these documents, which also consist of many segments caused by bullet points.

Table 3.8 shows comparisons between questions from Tasks 1 and 2 (original) and Task 3 (paraphrase) in Section 3.1.2. As expected, noticeable performance drop is found for the paraphrased questions, which have much fewer overlapping words to the answer contexts than the original questions.

Table 3.9 shows the MRR scores with respect to question types. The convolutional neural network models show strength on the 'who' type. It is due to the fact that these questions often contain crucial, unique information in a sentence that is an answer, which can be easily caught by the network. On the other hand, the models struggle the most with 'why' questions; it is likely due to unique structure of these questions, in which answers often might be sparse among multiple sentences and thus it is difficult locate a single candidate that would support the question.

Type	CNN_0	CNN_2	Q
What	84.54	85.36	678
How	81.92	84.01	233
Who	**85.46**	**88.17**	195
When	84.21	85.56	180
Where	83.78	87.44	85
Why	78.55	82.64	41
Misc.	84.17	84.80	215

Table 3.9: MRR scores on the SelQA evaluation set for answer sentence selection w.r.t. question types.

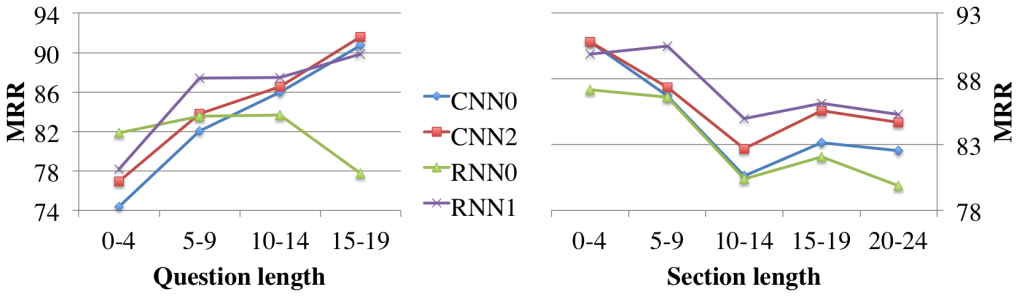

Figure 3.5: Answer sentence selection on the SelQA evaluation set w.r.t. question and section lengths.

Model	Development			Evaluation		
	P	**R**	**F1**	**P**	**R**	**F1**
CNN_0: baseline	41.86	42.86	42.35	29.70	37.45	32.73
CNN_1: `max` + word	44.53	45.24	**44.88**	29.77	42.39	34.97
CNN_2: `max` + emb	43.07	46.83	44.87	29.77	42.39	34.97
CNN_3: `max` + emb+	44.44	44.44	44.44	29.43	48.56	**36.65**
Yang et al. [161]	-	-	-	27.96	37.86	32.17

Table 3.10: Answer triggering results on WikiQA.

Finally, Figure 3.5 shows the performance differences with respect to question and section lengths. All models tend to perform better as questions become longer; this makes sense since longer questions are usually more informative and contains more details about the context. On the other hand, models generally perform worse as sections become longer, which also makes sense because the models have to select the answer contexts from larger pools.

Answer Triggering

Due to the nature of answer triggering, metrics used for evaluating answer sentence selection are not used here, because those metrics assume that models are always provided with contexts including the answers. Broadly speaking, the answer sentence selection task is a raking problem, while answer triggering is a binary classification task with additional constraints. Thus, the F1-score on the question level was proposed by [161] as the evaluation for this task, which is followed in this work.

Model	Development			Evaluation		
	P	**R**	**F1**	**P**	**R**	**F1**
CNN_0: baseline	50.63	40.60	45.07	52.10	40.34	45.47
CNN_1: `max` + word	48.15	47.99	48.07	52.22	47.30	49.64
CNN_2: `max` + emb	49.32	48.99	**49.16**	53.69	48.38	**50.89**
CNN_3: `max` + emb+	47.16	47.32	47.24	52.14	47.14	49.51

Table 3.11: Answer triggering results on SelQA.

Table 3.10 shows the answer triggering results on WikiQA. In this setting, it is more reasonable to use the `max` method for the f_m function; Unlike in the answer sentence selection task, the answer triggering setup provides a vast range of contexts including the ones that are completely irrelevant. Therefore, the `max` function will try to maximize a possible contextual match with any sentence. In fact, this is exactly what is observed where the model with f_m = `max` outperformed the other metrics for answer triggering. The convolutional neural network model with subtree matching models consistently gave over 2% improvements to the baseline model.

In addition, CNN_3 was experimented by retraining word embeddings (emb+), which performed slightly worse on the development set, but gave another 1.68% improvement on the evaluation set. It is interesting to see that while this technique did not help in the answer sentence selection task, it shows a significant improvement in the triggering configuration. While the reason of this behavior remains unclear, it might due to the fact that answer triggering combines much broader contexts with respect to queries thus the neural network is able to find new semantic relations between words.

Topic	**CNN_0**	**CNN_2**	**Q**
Arts	27.45	31.37	135
Country	43.59	**61.54**	178
Food	31.40	44.19	147
H. Events	**60.32**	**63.49**	164
Movies	37.74	45.28	164
Music	29.31	36.21	155
Science	45.00	57.50	179
Sports	50.00	58.11	168
Travel	42.68	50.00	165
TV	32.79	32.79	135

Table 3.12: Accuracies on the SelQA evaluation set for answer triggering with respect to topics.

Table 3.11 shows the answer triggering results on SelQA. Unlike the results on WikiQA (Table 3.10), CNN_2 outperforms CNN_3 on the SelQA corpus. CNN_2 using subtree matching gives over a 5% improvement to the baseline model, which is significant.

Table 3.12 shows the accuracies on SelQA with respect to different topics. The accuracy is measured on the subset of questions that contain at least one answer among candidates; the top-ranked sentence is taken and checked for the correct answer. Similar to answer sentence selection, CNN_2 stills shows strength on topics such as 'Country' and 'Historical Events', but the trend is not as clear for the other models. The worst performing topics are 'TV', 'Music' and 'Art'. Such a noticeable difference might be caused by the unusual semantic sentence constructions of the text. Sections in these categories often contain listings, bullet-pointed texts etc., which is problematic

Type	CNN_0	CNN_2	Q
Original	46.15	55.13	810
Paraphrase	31.52	38.52	789

Table 3.13: Accuracies on the SelQA evaluation set for answer triggering w.r.t. paraphrasing.

Type	CNN_0	CNN_2	Q
What	40.68	50.19	678
How	36.63	43.56	233
Who	**44.94**	50.56	195
When	33.33	43.06	180
Where	33.33	**51.85**	85
Why	42.11	47.37	41
Misc.	44.90	51.02	215

Table 3.14: Accuracies on the SelQA evaluation set for answer triggering w.r.t. question types.

for the models to properly take care of. How to correctly understand and solve a question from such context will be a challenge to the future systems.

Table 3.13 shows the accuracies on SelQA with respect to paraphrasing, which is similar to the trend found in Table 3.8 for answer sentence selection.

Table 3.14 shows the accuracies on SelQA with respect to question types. Interestingly, each model shows different strength on different types, which may suggest a possibility of an ensemble model. Finally, Figure 3.6 shows the performance difference with respect to question and section lengths for the

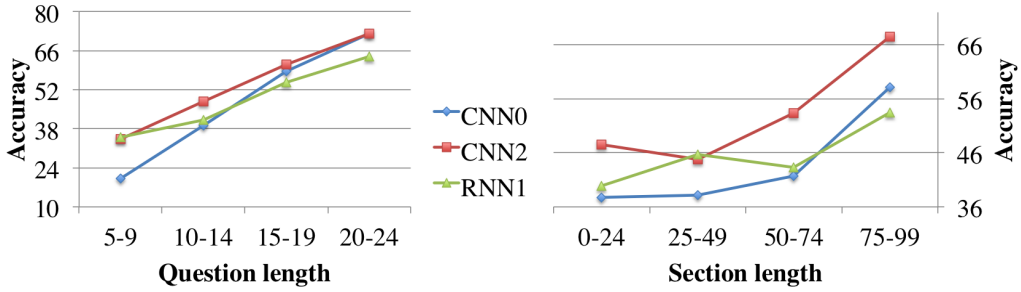

Figure 3.6: Answer triggering on the SelQA evaluation set w.r.t. question and section lengths.

answer triggering task. All the models tend to perform better as questions become longer. Similarly as in the answer sentence selection task, since longer questions are more informative, it is understandable. Interestingly, once the section becomes longer, the accuracy increases. It is likely that such a behavior might be caused by the fact that it is easier for the models to decide whether the context of the section is the same as the context of the question when there is more information (sentences) in the section. Thus, this phenomenon is related to the task of answer triggering, where the model not only choose the sentence with the answer, but must decide if the context matches first.

3.3 Cross-evaluation of Factoid Question Answering Corpora

In the last decade, researchers have devoted themselves to building and publishing multiple question answering corpora. These corpora are often built

independently and can be used to benchmark various approaches for question answering. Due to the fact of growing popularity of question answering, more datasets have started appearing in this field. Therefore, it is reasonable to evaluate these corpora themselves, so better use of them can be taken. By understanding the semantics hidden behind specific corpora, it is possible to improve current approaches. For instance, the performance boost could be achieved by transferring the knowledge from one dataset to another. Unfortunately, this task is problematic and challenging. Since these corpora are created independently, their creators often dedicate them to different tasks and thus making it impossible to easily combine them.

This section presents a thorough analysis done internally and externally on currently existing corpora in open-domain question answering. First, an intrinsic analysis is performed to understand crucial differences between them (Section 3.3.1). Next, an answer passage retrieval task is performed to map each question and their contents to the current version of English Wikipedia (Section 3.3.2) allowing all corpora to be used on the same set of tasks. Finally, an extrinsic analysis is presented through a set of experiments cross-testing these corpora using a convolutional neural network architecture. neural network architecture (Section 3.3.3)

3.3.1 Intrinsic Analysis

Four publicly available corpora are selected for the analysis. These corpora are based on Wikipedia, so more comparable than the others, and have already been used for the evaluation of several QA systems.

WikiQA [161] comprises questions selected from the Bing search queries,

where user click data give the questions and their corresponding Wikipedia articles. The abstracts of these articles are then extracted to create answer candidates. The assumption is made that if many queries lead to the same article, it must contain the answer context; however, this assumption fails on some occasions, which makes this dataset more challenging. Since the existence of answer contexts is not guaranteed in this task, it is called answer triggering instead of answer selection.

SelQA [70] is a product of five annotation tasks through crowdsourcing (Section 3.1) It consists of about 8K questions where a half of the questions are paraphrased from the other half, aiming to reduce contextual similarities between questions and answers. Each question is associated with a section in Wikipedia where the answer context is guaranteed, and also with five sections selected from the entire Wikipedia where the selection is made by the Lucene search engine. This second dataset does not assume the existence of the answer context, so can be used for the evaluation of answer triggering.

SQuAD [106] presents 107K+ crowdsourced questions on 536 Wikipedia articles, where the answer contexts are guaranteed to exist within the provided paragraph. It contains annotation of answer phrases as well as the pointers to the sentences including the answer phrases; thus, it can be used for both answer extraction and selection. This corpus also provides human accuracy on those questions, setting up a reasonable upper bound for machines. To avoid overfitting, the evaluation set is not publicly available although system outputs can be evaluated by their provided script.

InfoboxQA [93] gives 15K+ questions based on the infoboxes from 150 articles in Wikipedia. Each question is crowdsourced and associated with an

infobox, where each line of the infobox is considered an answer candidate. This corpus emphasizes the gravity of infoboxes, which summary arguably the most commonly asked information about those articles. Although the nature of this corpus is different from the others, it can also be used to evaluate answer selection.

Analysis

Table 3.15 presents the comparisons between the four analyzed corpora. Note that both WikiQA and SelQA provide separate annotation for answer triggering, which is not shown in this table. The SQuAD column shows statistics excluding the evaluation set, which is not publicly available. AE, AS and AT denotes annotation for answer extraction, answer sentence selection and answer triggering. q/c shows the average number of candidate sentence per question. w/t presents the overall number of all tokens and the size of vocabulary, respectively. $\mu_{q/c}$ shows the average length of questions and their candidates. $\Omega_{q/a}$ shows a macro average in % of overlapping words between question-answer pairs normalized by the questions/answers lengths (Ω_f: $^{(2\cdot\Omega_q\cdot\Omega_a)}/_{(\Omega_q+\Omega_a)}$).

All corpora provide datasets/splits for answer selection, whereas only (WikiQA, SQuAD) and (WikiQA, SelQA) provide datasets for answer extraction and answer triggering, respectively. SQuAD is much larger in size although questions in this corpus are often paraphrased multiple times. On the contrary, SQuAD's average candidates per question ($^c/_q$) is the smallest because SQuAD extracts answer candidates from paragraphs whereas the others extract them from sections or infoboxes that consist of bigger contexts. Al-

	WikiQA	**SelQA**	**SQuAD**	**InfoboxQA**
Source	Bing search queries	Crowdsourced	Crowdsourced	Crowdsourced
Year	2015	2016	2016	2016
(AE, AS, AT)	(O, O, O)	(X, O, O)	(O, O, X)	(X, O, X)
$(q, c, c/q)$	(1 242, 12 153, 9.79)	(7 904, 95 250, 12.05)	(**98 202**, 496 167, 5.05)	(15 271, 271 038, **17.75**)
(w, t)	(386 440, 30 191)	(3 469 015, 44 099)	(19 445 863, **115 092**)	(5 034 625, 8 323)
(μ_q, μ_c)	(6.44, 25.36)	(11.11, 25.31)	(11.33, 27.86)	(9.35, 9.22)
$(\Omega_q, \Omega_a, \Omega_f)$	(46.72, **11.05**, 16.96)	(32.79, 16.98, 20.19)	(32.27, 12.15, **16.54**)	(**26.80**, 35.70, 28.09)

Table 3.15: Comparisons between the four corpora for answer selection.

though InfoboxQA is larger than WikiQA or SelQA, the number of token types (t) in InfoboxQA is smaller than those two, due to the repetitive nature of infoboxes.

All corpora show similar average answer candidate lengths (μ_c), except for InfoboxQA where each line in the infobox is considered a candidate. SelQA and SQuAD show similar average question lengths (μ_q) because of the similarity between their annotation schemes. It is not surprising that WikiQA's average question length is the smallest, considering their questions are taken from search queries. InfoboxQA's average question length is relatively small, due to the restricted information that can be asked from the infoboxes. InfoboxQA and WikiQA show the least question-answer word overlaps over questions and answers (Ω_q and Ω_a in Table 3.15), respectively. In terms of the F1-score for overlapping words (Ω_f), SQuAD gives the least portion of overlaps between question-answer pairs although WikiQA comes very close.

Figure 3.7 shows the distributions of seven question types grouped deterministically from the lexicons. Although these corpora have been independently developed, a general trend is found, where the *what* question type

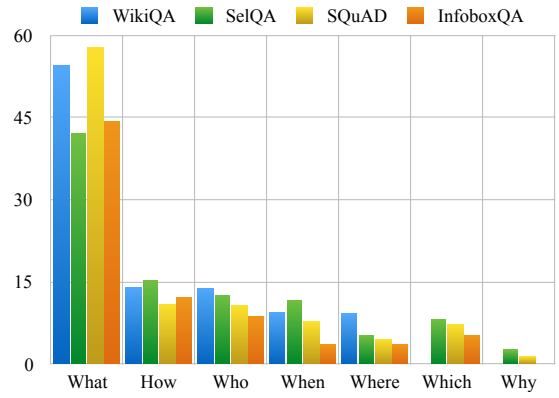

Figure 3.7: Distributions of question types in %.

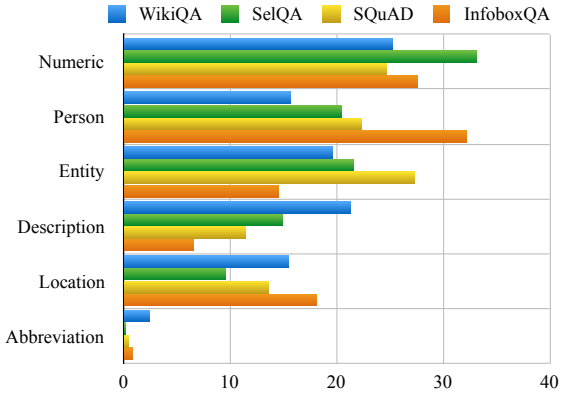

Figure 3.8: Distributions of answer categories in %.

dominates, followed by *how* and *who*, followed by *when* and *where*, and so on.

Figure 3.8 shows the distributions of answer categories automatically classified by the Convolutional Neural Network model trained on the data distributed by [79].[5] Interestingly, each corpus focuses on different categories, *Numeric* for WikiQA and SelQA, *Entity* for SQuAD, and *Person* for In-

[5]The CNN model shows 95.20% accuracy on the test set.

FOBOXQA, which gives enough diversities for statistical learning to build robust models.

Finding a paragraph that includes the answer context out of the entire Wikipedia is an extremely difficult task ($1/28.7M$). The last row of Table 3.16 shows results from answer retrieval. Given $k = 5$, SELQA and SQUAD show about 34% and 35% accuracy, which are reasonable. However, WIKIQA shows a significantly lower accuracy of 12.47%; this is because the questions in WIKIQA are about twice shorter than the questions in the other corpora such that not enough lexicons can be extracted from these questions for the Lucene search.

3.3.2 Answer Passage Retrieval

This section describes another selection-based QA task, called *answer passage retrieval*, that finds the answer context from a larger dataset, the entire Wikipedia. SQUAD provides no mapping of the answer contexts to Wikipedia, whereas WIKIQA and SELQA provide mappings; however, their data do not come from the same version of Wikipedia. An automatic way of mapping the answer contexts from all corpora to the same version of Wikipeda[6] is presented so they can be coherently used for sentence selection tasks.

Each paragraph in Wikipedia is first indexed by Lucene using {1,2,3}-grams, where the paragraphs are separated by WikiExtractor[7] and segmented by NLP4J[8] (28.7M+ paragraphs are indexed). Each answer sentence from

[6] `enwiki-20160820-pages-articles.xml.bz2`

[7] `github.com/attardi/wikiextractor`

[8] `github.com/emorynlp/nlp4j`

	WikiQA	**SelQA**	**SQuAD**
$(\rho, \gamma_c, \gamma_p), t \geq 0.3$	(92.00, 1 203, 96.86)	(90.00, 7 446, 94.28)	(100.00, 93 928, 95.61)
$(\rho, \gamma_c, \gamma_p), t \geq \mathbf{0.4}$	(**94.00**, **1 139**, **91.71**)	(**94.00**, **7 133**, **90.31**)	(**100.00**, **93 928**, **95.61**)
$(\rho, \gamma_c, \gamma_p), t \geq 0.5$	(100.00, 1 051, 84.62)	(98.00, 6 870, 86.98)	(100.00, 93 928, 95.61)
$k = (1, \mathbf{5}, 10, 20)$	$(4.39, \mathbf{12.47}, 16.59, 22.39)$	$(20.01, \mathbf{34.07}, 40.29, 46.40)$	$(19.90, \mathbf{35.08}, 40.96, 46.74)$

Table 3.16: Statistics of the silver-standard dataset (first three rows) and the accuracies of answer retrieval in % (last row).
ρ: robustness of the silver-standard in %, $\gamma_{c/p}$: #/% of retrieved silver-standard passages (coverage).

the corpora in Table 3.16 is then queried to Lucene, and the top-5 ranked paragraphs are retrieved. The cosine similarity between each sentence in these paragraphs and the answer sentence is measured for n-grams, say $n_{1,2,3}$. A weight is assigned to each n-gram score, say $\lambda_{1,2,3}$, and the weighted sum is measured: $t = \sum_{i=1}^{3} \lambda_i \cdot n_i$. The fixed weights of $\lambda_{1,2,3} = (0.25, 0.35, 0.4)$ are used for the experiments, which can be improved in the future.

If there exists a sentence whose $t \geq \theta$, the paragraph consisting of that sentence is considered the silver-standard answer passage. Table 3.16 shows how robust these silver-standard passages are based on human judgment (ρ) and how many passages are collected (γ) for $\theta = [0.3, 0.5]$, where the human judgment is performed on 50 random samples for each case. For answer retrieval, a dataset is created by $\theta = 0.4$, which gives $\rho \geq 94\%$ accuracy and $\gamma_p > 90\%$ coverage, respectively.[9] Finally, each question is queried to Lucene and the top-k paragraphs are retrieved from the entire Wikipedia.

[9]SQuAD mapping was easier than the others because it was based on a more recent version of Wikipedia.

If the answer sentence exists within those retrieved paragraphs according to the silver-standard, it is considered correct.

3.3.3 Extrinsic Analysis

Answer Sentence Selection

Answer sentence selection is evaluated by two metrics, mean average precision (MAP) and mean reciprocal rank (MRR). The bigram CNN introduced by [171] is used to generate all the results in Table 3.17, where models are trained on either single or combined datasets. Clearly, the questions in WikiQA are the most challenging, and adding more training data from the other corpora hurts accuracy due to the uniqueness of query-based questions in this corpus. The best model is achieved by training on W+S+Q for SelQA; adding InfoboxQA hurts accuracy for SelQA although it gives a marginal gain for SQuAD. Just like WikiQA, InfoboxQA performs the best when it is trained on only itself. From the analysis, it is suggested to use models trained on WikiQA and InfoboxQA for short query-like questions, whereas to use ones trained on SelQA and SQuAD for long natural questions. Even with enough similarities, training on SQuAD and testing on SelQA performs slightly worse than training and testing on SelQA because the size of answer candidates is more than twice bigger in SelQA than SQuAD. Comparing W+S+Q with W+S+Q+I, WikiQA also finds InfoboxQA useful since questions in InfoboxQA are similar to search queries on the infoboxes.

Trained on	Evaluated on											
	WikiQA			SelQA			SQuAD			InfoboxQA		
	MAP	MRR	F1	MAP	MRR	F1	MAP	MRR	F1	MAP	MRR	F1
WikiQA	**65.54**	**67.41**	13.33	53.47	54.12	8.68	73.16	73.72	11.26	30.85	30.85	-
SelQA	49.05	49.64	**24.30**	82.72	83.70	**48.66**	77.22	78.04	44.70	63.13	63.13	-
SQuAD	58.17	58.53	19.35	81.15	82.27	42.88	88.84	89.69	**44.93**	63.24	63.24	-
InfoboxQA	45.17	45.43	-	53.48	54.25	-	65.27	65.90	-	**79.44**	**79.44**	-
W+S+Q	56.40	56.51	-	**83.19**	**84.25**	-	88.78	89.65	-	62.53	62.53	-
W+S+Q+I	60.19	60.68	-	82.88	83.97	-	**88.92**	**89.79**	-	70.81	70.81	-

Table 3.17: Results for answer selection and triggering in % trained and evaluated across all corpora splits. The first column shows the training source, and the other columns show the evaluation sources. W: WikiQA, S: SelQA, Q: SQuAD, I: InfoboxQA.

Answer Triggering

The results of $k = 5$ from the answer retrieval task in Section 3.3.2 are used to create the datasets for answer triggering, where about 65% of the questions are not expected to find their answer contexts from the provided paragraphs for SelQA and SQuAD and 87.5% are not expected for WikiQA. Answer triggering is evaluated by the F1 scores as presented in Table 3.17, where three corpora are cross-validated. The results on WikiQA are pretty low as expected from the poor accuracy on the answer retrieval task. Training on SelQA gives the best models for both WikiQA and SelQA. Training on SQuAD gives the best model for SQuAD although the model trained on SelQA is comparable. Since the answer triggering datasets are about 5 times larger than the answer selection datasets, it is computationally too expensive to combine all data for training. The future work consists of finding a strong machine to perform this experiment.

3.4 Summary

This chapter described several advancements and additions to different methods in question answering. A multi-stage annotation scheme was presented that addresses the lack of access to large collections of data by researchers. It allows to generate a diverse, challenging and realistic corpus for various tasks in open-domain question answering. The scheme provides a quality control and low-cost thanks to the crowdsourcing techniques that were used. Next, the subtree matching algorithm was presented that focuses on measuring the contextual similarity. The developed method supports similarity metrics based on distributional semantics. Also, the algorithm is syntactically transparent, unlike other methods which are based on specific syntactic structures. The algorithm proved its power by outperforming several state-of-the-art results on the answer sentence selection and answer triggering tasks. Finally, a comprehensive analytical study on several open-domain question answering corpora was presented. It opens the path for the future work on the corpora combinations and transfer learning. Moreover, the analysis has shown that the dataset created during this work, SelQA, is almost identical when compared to SQuAD. Researchers can use an order of magnitude smaller dataset to train almost identical models in terms of performance. This shows that SelQAis a diverse and linguistically difficult corpus that can be used as a benchmark. In fact, more recently, researchers have already used the SelQAcorpus to evaluate their approaches to selection-based question answering [47, 126].

Chapter 4

Non-factoid Question Answering

Chapter 3 described several advancements designed for open-domain question answering. Questions that do not fall into the factoid category are classified as *non-factoid*. Non-factoid questions are highly unspecified and their expected answer type, context, *etc.* depends on the type of questions. For instance, a recommendation question might have a snippet of text as an answer. On the other hand, math and logical problems assume that an answer is a number.

Non-factoid questions such as arithmetic often require customized approaches and special treatments. Consider the arithmetic question: *"Tom found 7 seashells but 4 were broken. How many unbroken seashells did Tom find?"*. Its answer is a number that is a result of an algebraic formula:

$$x = 7 - 4$$

The answer to this question is the number 3 which does not appear in the context directly. In arithmetic problems, it is crucial that the system is able to build an abstract representation when given any text. Then, this representation can be used to recognize the question intent, match the information from the question, and finally form an answer.

On the other side, there exist questions that require systems to perform a temporal, lexical, and semantic analysis. Consider a fiction story that consists of a set of characters, themes, locations, *etc.* The story is described in a set of sequential sentences where the characters move themes within various locations, *etc.* When a question is asked, the system must be capable of merging several layers of information which is difficult.

In this chapter, two types of non-factoid question answering tasks are explored: *arithmetic* and *event-based* question answering. First, a semantic-based graph structure is presented that is built on the foundation of several core natural language processing tasks (Section 4.1). As a proof of concept, arithmetic questions are used as an experimentation setup to prove that the designed structure is valid. Next, a modular approach for non-factoid question answering is introduced (Section 4.2). The system combines good aspects of natural language processing and information retrieval. The framework is based on the structural decomposition of linguistic structures. The evaluation on event-based questions shows an improvement of over 40% compared to a lexical search.

4.1 Semantics-based Graph Approach to Complex Question Answering

Thanks to years of research on statistical parsing, several tools are available that provide rich syntactic and semantic structures from texts. The output of these tools, however, often needs to be post-processed into more complicated structures, such as graphs of knowledge, in order to retrieve answers to complex questions. These graphs consist of relations between entities found not only within a sentence but also across sentences. Vertices and edges in these graphs represent linguistic units (*e.g.*, words, phrases) and their syntactic or semantic relations, respectively.

Robustness of handling several types of questions is one of the key aspects of a question answering system. Recently, researchers started focusing on solving complex questions involving arithmetics or biological processes [66, 9]. A complex question can be described as a question requiring the collection and synthesis of information from multiple sentences [21]. The more complex the questions become, the harder it is to build a structural model that is general enough to capture information for all different types of questions. This section presents an architectural approach of representing entity relations as well as its application to complex question answering.

4.1.1 Semantics-based Knowledge Approach

The motivation arises from both the complexity and the variety of questions and their relevant contexts. The complexity concerns with exploiting syntactic dependencies, semantic role labels, named entities, and coreference links

all together for finding the best answers. For arithmetic questions, such complexity comes from the flow of entity relations across sentences and semantic polarities of verb predicates, which are required to transform the contexts in natural language into mathematical equations.

The variety concerns with robustly handling various types of questions. It is relatively easier to develop an architecture designated to handle just one type of questions (*e.g.*, a system to extract answers for factoid questions) than many different types of questions (*e.g.*, opinions, recommendations, commentaries). In this section, a semantic-based knowledge approach (constructed graph) is presented that not only conveys relations from different layers or linguistic theories, but also is effective for finding answers for various types of questions.

Components

Given a *document*, the system first parses each sentences into a dependency tree, then finds predicate-argument structures on top of the dependency tree. Once sentences are parsed, coreference links are found for nodes across all trees. Finally, each dependency node gets turned into an *instance*, which can be linked to other related instances. Multiple instances can be grouped together as an *entity* if they are coreferent. The graph is semantically driven because semantic predicate-argument relations take precedence over syntactic dependencies when both exist.

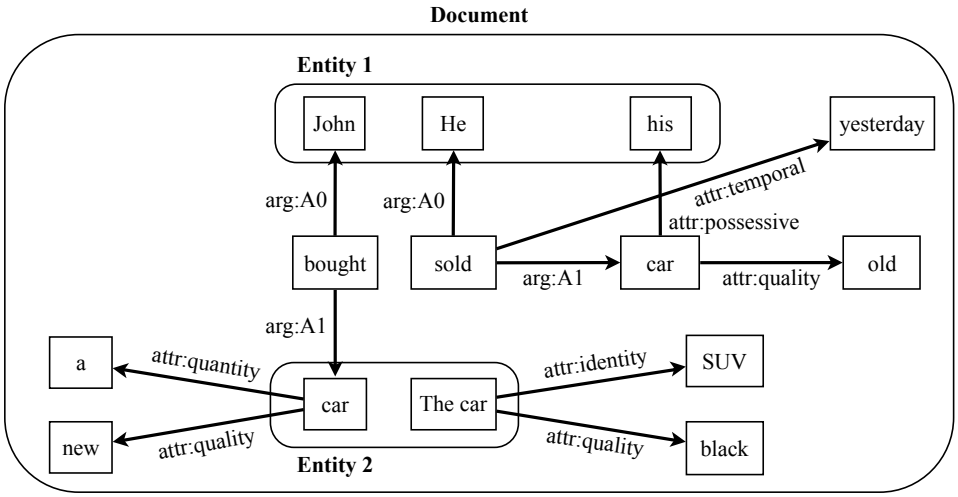

Figure 4.1: Example of the semantic-based graph given three sentences: *John bough a new car*, *The car was black SUV*, and *He sold his old car yesterday.*

Document

A *document* contains a graph consisting of a set of entities, instances, and relations between the instances (Figure 4.1). A document can be small as a microblog or big as the entire Wikipedia articles.

Entity

An *entity* can be described as a set of instances referring to the same object mostly found through coreference resolution. In Figure 4.1, although *John*, *He*, and *his* are recognized as individual instances, they are grouped into one entity because they all refer to *John*. Maintaining these relations is crucial for answering complex questions.

Instance

An *instance* is the atomic-level object in the graph that usually represents a word-token, but can also represent compound words (*e.g.*, *New York*), multi-word expressions, etc. The instance is linked to other instances as a predicate, an argument, or an attribute.

Predicate & Argument

An instance is a *predicate* of another instance if it forms any argument structure [102]. Currently, the designed graph takes non-auxiliary verbs and a few eventive nouns as predicates provided by a semantic role labeler. An instance is an *argument* of another if it is required to complete the meaning of the other instance. In Figure 4.1, *John* and *car* are arguments of *bought* because they are necessary to give an understanding of *bought*. It is a task for the future work how to improve these relations through semantic parsing.

The *predicate* and *argument* relations represent both semantic and syntactic relations between instances in the document. Semantic role labels [102] and dependency labels in [34] are used to represent semantic and syntactic relations in this graph. Experiments show that these relations play a crucial role in answering arithmetic questions (Section 4.1.4).

Attribute

An instance is an *attribute* of another if it is not an argument but gives extra information about the other instance. While an argument completes the meaning of its predicate, an attribute augments the meaning with specific information. In Figure 4.1, *new* is not an argument but an attribute of *car*

Type	Description
Locative	Geographical or relative location information (e.g., *New York*, *near my house*).
Temporal	Absolute or relative temporal information (e.g., *tomorrow noon*, *2 years ago*).
Possessive	Possessor of this instance (e.g., *his*, of *Mary*).
Quantity	Absolute or relative quantity information (e.g., *two* books, *few* books).
Quality	Every other kind of attributes.

Table 4.1: List of attributes used in the graph.

because this information is not required for understanding *car*, but provides finer-grained information about the car.

Attributes can be shared among instances within the same entity. In Figure 4.1, the attributes *new* and *black* are shared between instances *car* and *the car*. This is particularly useful for questions requiring information scattered across sentences. Table 4.1 shows the types of attributes that have been specified so far. This list will be continuously updated as more question types are added to the system.

4.1.2 Graph Construction

Algorithm 2 shows a pseudo-code for constructing the graph given a dependency tree, consisting of syntactic and semantic relations, and coreference

Input: D: a dependency tree,

C: a set of coreference links.

Output: G: Graph.

foreach *node* N *in* D **do**

  if N.*skip()* **then**

    continue;

  else if N.*isArgument()* **then**

    $P \leftarrow N$.getPredicate();

    $L \leftarrow N$.getArgumentLabel();

    G.addArgument(P, N, L);

  else if N.*isAttribute()* **then**

    $A \leftarrow N$.getAttributeHead();

    $L \leftarrow N$.getAttributeType();

    G.addAttribute(A, N, L);

  else

    $H \leftarrow N$.getSyntacticHead();

    $L \leftarrow N$.getSyntacticLabel();

    G.addArgument(H, N, L);

  end

  if C.*hasEntityFor(*N*)* **then**

    $E \leftarrow C$.getEntityFor(N) G.addToEntity(E, N);

end

Algorithm 2: Graph constructing algorithm.

links.

Every node in the dependency tree has exactly one syntactic head and can be a semantic argument of zero to many predicates. For each node, it first checks if this node should be added to the graph (i.e., auxiliary verbs are not added). If it should, it checks it is a semantic argument of some predicate. If not, it checks if it is an attribute of some instance. By default, it becomes an argument of its syntactic head. Finally, it gets added to an entity if it is coreferent to some other instance. Moreover, the graph is also designed to support weights of vertices and edges. Now, the algorithm assigns a value of 1 as a weight for every element, but it is planned for the future work to extend this work by determining the importance of different weights for specific semantic relations. It is likely that a more intelligent weighting system would improve the overall accuracy of the system by enhancing the matching process.

4.1.3 Arithmetic Questions

This section demonstrates an approach to the application of complex question answering, targeted on arithmetic questions. The purpose of this section is to show a proof of concept that described above graph can be effectively applied to answer such questions. For experiments, a set of arithmetic questions is taken and used for elementary and middle school students. These questions consist of simple arithmetic operations such as addition and subtraction. Table 4.2 shows a sample of these questions.

The main challenge of this task is mostly related to the contiguous representation of state changes. The question at the end concerns about either the

Question	Equation
A restaurant served 9 pizzas during lunch and 6 during dinner today. How many pizzas were served today?	x = 9 + 6
Tim's cat had kittens. He gave 3 to Jessica and 6 to Sara. He now has 9 kittens. How many kittens did he have to start with?	x = 3 + 6 + 9

Table 4.2: Sample of arithmetic questions.

start state, the transitions, or the end state of a specific theme (e.g., *pizza*, *kitten*). Therefore, simplistic string matching approaches, which would have worked well on factoid questions, would not perform well on this type of questions. Another challenge is found by coreference mentions in these questions. Arithmetic questions generally consist of multiple sentences such that coreference resolution plays a crucial role in getting high accuracy.

Verb polarity sequence classification

The task of solving of arithmetic questions is turned into a sequence classification of verb polarities. It is likely that the verbs need to be classified in sequence because the same verb can convey different polarities in different contexts. Three types of verb polarities are used: `+`, `-`, and `0`. Given the list of sentences in each question and the equation associated with it (Table 4.2), each verb is mapped with its polarity by comparing their quantities. '`+`' and

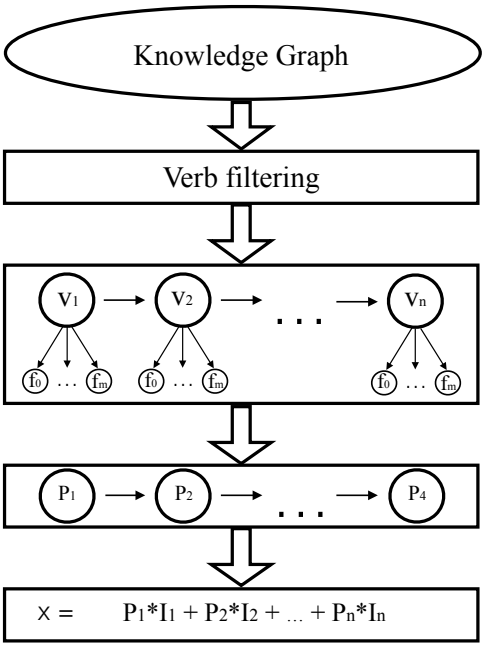

Figure 4.2: Flow of execution in the system for solving arithmetic questions. First, the verb filtering process is applied to select verbs in all sentences (V_i), which share the same semantic argument with the question. Given the selected verbs, their features (f_i) are extracted and the polarities (P_i) are predicted by a statistical model. Finally, the equation X is formed, where polarities are multiplied by the quantities of the arguments.

'−' are assigned to verbs whose arguments show a plus sign or a minus sign in the equation, respectively. '0' is assigned to verbs whose arguments do not appear in the equation. This information is used to build a statistical model, which is used for decoding.

Arithmetic questions often contain verbs whose arguments are not relevant to the final question. For instance, in "*Jason has 43 blue and 16 red marbles. Tom has 24 blue marbles. How many blue marbles do they have in all?*",

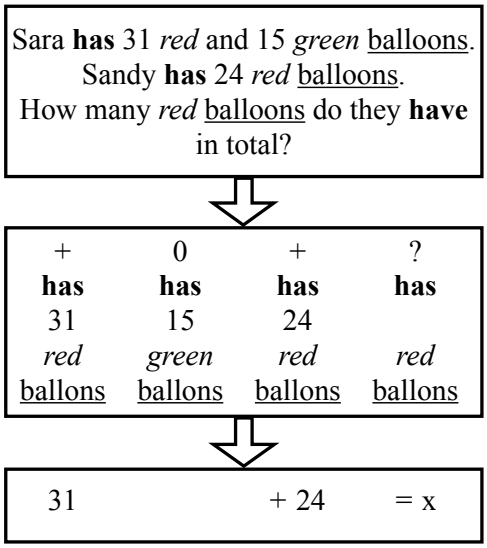

Figure 4.3: Flow of execution for the example document. First, verbs are filtered and selected for the polarity selection. Next, all necessary information (numericals, themes etc.) is collected and organized into states. Finally, based on the verbs polarity, equation is being formed.

"*16 red marbles*" is more like a noise to answer this question. Presented approach classifies such verbs as `0` so that they do not participate into the final equation. Once the equation is formed, it is trivial to solve the problem using simple algebra.

The designed approach is distinguished from some of the previous work where each verb is categorized into multiple classes [66] in a sense that the verb classes used in this work are automatically derived from the equations (no extra annotation is needed). Furthermore, this approach can be extended to more complicated operations such as multiplication and division as long as the correct equations are provided. The dataset used in [76] contains this

type of questions and it is planned for the future work to apply this approach to this dataset.

4.1.4 Experiments

Data

The arithmetic dataset provided by the Allen Institute.[1] is used for experiments. The corpus of 395 arithmetic questions together with their equations and answers. All data has been parsed using the dependency parser, the semantic role labeler, the named entity tagger, and the coreference resolution in ClearNLP[2] [31, 29]. Then, the dataset was split into 3-folds for cross-validation in a way that the polarity distributions are similar across different sets (Table 4.3).

Features

The following features are used for the experiments:

- Semantic role labels; especially numbered arguments as in PropBank [102].
- Sequence of verbs and arguments whose semantic roles are recognized as 'themes'.
- Frequency of verbs and theme arguments in the current context.
- Similarity between verbs and theme arguments across sentences.
- Attributes of themes related to specific verbs.
- Distance from the verb to the final question.

[1] `allenai.org/content/data/arithmeticquestions.pdf`
[2] `http://www.clearnlp.com`

It was trivial to extract all these features using the constructed graph.

Machine learning

To build statistical models, stochastic adaptive subgradient algorithm called AdaGrad is used; per-coordinate learning rates were used to exploit rarely seen features while remaining scalable [48]. This is suitable for NLP tasks where rarely seen features often play an important role and training data consists of a large number of instances with high dimensional features. The implementation of AdaGrad in ClearNLP using the hinge-loss was used with their default hyper-parameters: learning rate: $a = 0.01$, termination criterion: $r = 0.1$.

Evaluation

Table 4.3 shows the distributions of each fold and the accuracy of the designed system in answering arithmetic questions. Cross-validation score is 71.75%, which is promising given how complex these questions are. [66] were able to achieve 77.7% accuracy on the same dataset, which is higher than above result. However, the main goal of these experiments remains as to prove that the designed graph can be utilized to answer complex questions. Moreover, [66] performed an extra annotation for verb classes using crowdsourcing; their approach is based on 7 different verb classes that significantly increase the complexity of semantic analysis. The approach presented above does not require any extra annotation. It is likely that by improving the graph and features that could be extracted, the model's performance would increase.

The majority of prediction errors were caused by errors from dependency

	1st fold	**2nd fold**	**3rd fold**
# of questions	118	118	118
# of verbs	418	423	420
# of + verbs	326	330	328
# of - verbs	51	51	51
# of 0 verbs	41	42	41
Accuracy	67.80	76.27	71.19

Table 4.3: Distributions and accuracies of all folds.

parsing, semantic role labeling, or coreference resolution. For instance, verbs are not recognized correctly in some dependency trees, which becomes a major factor in decreasing accuracy. Also, semantic role labels sometimes were incorrectly assigned, which extremely influenced the structure of the graph, where features could not be properly extracted. Also, as mentioned earlier, coreference resolution remains as one of the main challenges in handling complex questions.

4.2 Multi-field Structural Decomposition for Event-based Question Answering

Towards machine reading, question answering has recently gained lots of interest among researchers from both natural language processing [96, 165, 63] and information retrieval [121, 74]. People from these two research fields, NLP and IR, have shown tremendous progress on question answering, yet

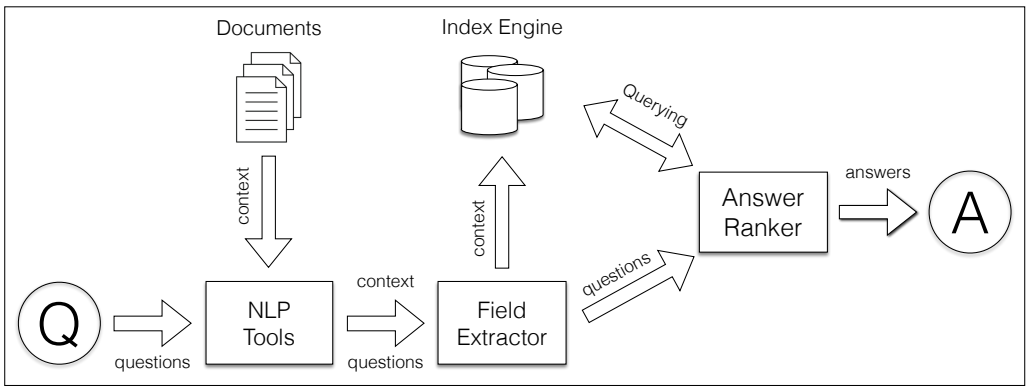

Figure 4.4: The overall framework of designed question answering system.

only a few efforts have been made to adapt technologies from both sides. The NLP side often tackles the task by analyzing linguistic aspects, whereas the IR side tackles it by searching likely patterns.

While these two approaches perform well individually, more sophisticated solutions are needed to handle a wide range of questions. By considering linguistic structures such as syntactic and semantic trees, QA systems can infer the deeper meaning of the context and handle more complex questions. However, extracting answers from these structures through either graph matching or predicate logic is not necessarily scalable when the size of the context is large. On the other hand, searching patterns is scalable for large data, especially when coupled with indexing, although it does not always concern with the actual meaning of the context.

4.2.1 Approach

Figure 4.4 shows the overall framework. The system is designed in a modular, architectural way, so any further extension of fields can be easily in-

tegrated. The designed system takes input documents, generates linguistic structures using natural language processing tools, decomposes them into multiple fields, and indexes those fields. Questions are processed in the same way. To answer a question, the system queries the index for each field extracted from the question and measures the relevance score. All documents are ranked with respect to the relevance scores and their weights associated with the fields, and the document with the highest score is selected as the answer.

Modules

The system consists of several modules closely connected together providing a fully working solution for the question answering selection task.

Documents and questions

Documents provide the context where the questions find their answers from. Each document can contain one or more sentences, in which answers for coming questions are annotated for training. Documents may simply be Wikipedia articles, news articles, fictional stories, etc. Questions are treated as regular documents containing only one sentence.

NLP tools

For the generation of syntactic and semantic structures the following parsers are used: part-of-speech tagger [33], dependency parser [31], semantic role labeler [32], and coreference resolution tool in ClearNLP[3]. Ensuring good and

[3] `http://www.clearnlp.com`

robust accuracy of these NLP tools is important because all the following modules depend on their output.

Field extractor

The field extractor takes the linguistic structures from the natural language processing tools and decomposes them into multiple fields (Section 4.2.1). All fields extracted from the documents are passed to the index engine, whereas fields extracted from the questions are sent directly to the answer ranker module.

Index engine

The index engine is a search server that receives a list of fields decomposed by the field extractor, indexes terms in the fields, and responses to the queries generated from questions with their relevance scores. Elastic Search[4] is used as it provides a distributed, multi-tenancy-capable search.

Answer ranker

The answer ranker takes the decomposed fields extracted from a question, converts them into queries, and builds a matrix of documents with their relevance scores across all fields through the index engine. It also uses different weights for individual fields trained by statistical modeling; the statistical learning details are described later in this section.

[4] `https://www.elastic.co`

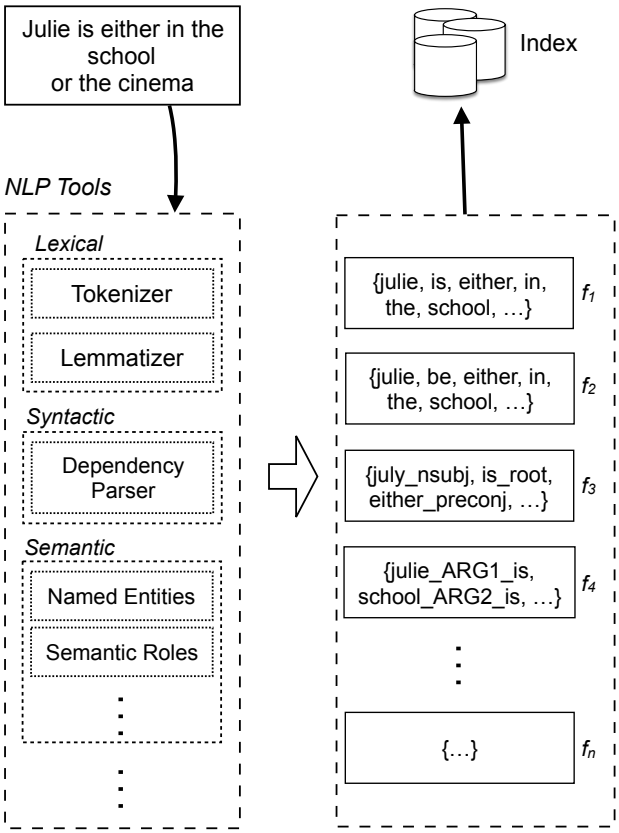

Figure 4.5: The flow of the sentence, *Julie is either in the school or the cinema*, through the system.

Structural decomposition

Each sentence is represented by the index engine as a document with multiple fields grouped into categories. Figure 4.5 shows an example of how the sentence is decomposed into multiple fields consisting of syntactic and semantic structures. Due to the extensible nature of the field extractor, additional groups and fields can be easily integrated. Currently, the system supports 24 fields that can be grouped in three categories: *lexical*, *syntactic* and *seman-*

Lexical fields
— word forms
— lemmas
— word stems
— verbs
— sentence id (used for distance)
Syntactic fields
— dependency labels
— dependency labels paired with words (*e.g.*: 'julie_nsubj')
Semantic features
— verb synonyms
— 'A0' roles
— 'A1' roles
— 'A2' roles
— 'A4' roles in a sentence
— 'DIR' modifiers
— 'LOC' modifiers
— 'NER' tags

Table 4.4: The list of all supported features divided into three categories.

tic (Table 4.4). Please note that some of the features are combined. (*e.g.*: 'julie_ARG1_school_ARG2', which combines A1 with A2)

Answer ranking

When a question q is asked, it is decomposed into the n-number of fields. Each field is transformed into a query where certain words are replaced with wildcards (*e.g.*, $\{where\_a1, is\_pred, she\_a2\} \rightarrow \{*\_a1\ is\_pred\ she\_a2\}$). Then, the relevance score r is measured between each field in the question and the

same field in each document $d^t \in D$ by the index engine.[5] The product of the relevance scores and individual weights for all fields are summed, and the document $\hat{d}$ with the highest score f is taken as the answer. Note that in the tested scenario, each document contains only one sentence so that retrieving a document is equivalent to retrieving a sentence. The following equations describe how the document $\hat{d}$ is selected by measuring the overall score $f(q, d^t)$ using the relevance scores $r(q_i, d_i^t)$ and the weights λ_i.

$$\hat{d} = \arg\max_{d^t \in D} f(q, d^t)$$
$$f(q, d^t) = \sum_{i=1}^{n} \lambda_i \cdot r(q_i, d_i^t)$$
$$r(q_i, d_i^t) = \sum_{v \in q_i \cap d_i^t} \mathrm{tf}_i^t(v) \cdot \mathrm{idf}_i(v) \cdot \mathrm{norm}_i^t(v)$$

Training weights for individual fields

Algorithm 3 shows how the weights for all fields are learned during training. The averaged perceptron algorithm is adapted to the training process, which has been widely used for many NLP tasks. All the weights $\vec{\lambda}$ are initialized to 1. For each question $q \in Q$, it predicts the document $\hat{d}$ that most likely contains the answer. If $\hat{d}$ is incorrect, then it compares the relevance score r between $(q, \hat{d})$ and (q, d) for each field, and updates the weight accordingly, where d is the true document from the oracle. This procedure is repeated multiple times through iterations. Finally, the algorithm returns the averaged

[5]The search results limit in Elasticsearch is set to 20.

weights, where each dimension represents the weight for each field. All hyper-parameters were optimized on the development sets and evaluated on the test sets. For the experiments, the following hyper-parameters were used: $M = 40, \alpha = 0.002$.

Input: D: document set, Q: question set.
M: max-number of iterations, α: learning rate.
Output: The averaged weight vector.

$\vec{\lambda} \leftarrow 1;\ \vec{\lambda'} \leftarrow 0$
for $iter \in [1, M]$ **do**
  foreach $q \in Q$ **do**
    $\hat{d} = \arg\max_{d^t \in D} f(q, d^t)$
    if $\hat{d} \neq d$ **then** # d is the oracle
      foreach $i \in [1, n]$ **do** ; ; ; # for each field
        $\delta \leftarrow \alpha \cdot \text{sign}[r(q_i, d_i) - r(q_i, \hat{d}_i)]$
        $\lambda_i \leftarrow \lambda_i + \delta$
    $\vec{\lambda'} \leftarrow \vec{\lambda'} + \vec{\lambda}$
return $\vec{\lambda'} \cdot \frac{1}{M*|Q|}$

Algorithm 3: Averaged perceptron training.

4.2.2 Experiments

Data and evaluation metrics

The approach is evaluated on a subset of the bAbI tasks [154]. The original data contains 20 tasks, where each task represents a different kind of question answering challenge. Eight tasks have been selected in which answer for

a single question is located within a single sentence. For consistency and replicability, the same training, development, and evaluation set splits as provided were followed; every set contains 1,000 questions.

For the evaluation metrics, mean average precision (MAP) and mean reciprocal rank (MRR) of the top-3 predictions are used. The mean average precision is measured by counting the number of questions, for which sentences containing the answers are correctly selected as the best predictions. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. Mean reciprocal rank is the average of the reciprocal ranks of all question queries.

Type	Lexical				Lexical + Syntax				Lexical + Syntax + Semantics			
	$\lambda = 1$		λ is learned		$\lambda = 1$		λ is learned		$\lambda = 1$		λ is learned	
	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR	MAP	MRR
1 (qa1)	39.62	61.73	39.62	61.73	29.90	48.05	40.50	61.47	72.60	85.07	**100.0**	**100.0**
2 (qa4)	62.90	81.45	62.90	81.45	64.00	82.00	64.00	82.00	55.70	77.85	**64.10**	**82.05**
3 (qa5)	37.10	54.00	38.20	54.70	48.00	62.15	48.40	62.25	72.60	82.65	**94.20**	**96.33**
4 (qa6)	64.00	75.07	64.00	75.07	65.80	78.47	66.10	78.53	78.20	88.33	**89.30**	**94.27**
5 (qa9)	47.90	63.50	48.10	63.62	47.90	63.67	50.50	65.47	53.90	67.88	**94.40**	**96.72**
6 (qa10)	47.80	63.78	47.90	63.92	49.20	65.52	50.20	66.33	57.60	70.68	**96.90**	**98.23**
7 (qa12)	19.20	38.68	19.20	38.68	25.10	40.83	31.90	49.82	55.00	70.60	**99.60**	**99.80**
8 (qa20)	37.10	51.82	37.10	51.82	31.40	42.00	35.70	44.22	31.20	46.50	**42.80**	**56.32**
Avg.	44.45	61.25	44.63	61.37	45.16	60.34	48.41	63.76	59.60	73.70	**85.16**	**90.47**

Table 4.5: Results from the question answering system on 8 types of questions in the bAbI tasks.

Evaluation

Table 4.5 shows the results on different types of questions. The MAP and MRR show clear correlation with respect to the number of active fields. For the majority of tasks, using only the lexical fields does not perform well. The fictional stories included in this data often contain multiple occurrences of the same lexicons, and the lexical fields alone are not able to select the correct answer. Significantly lower accuracy for the last task is due to a fact that besides an answer is located within a single sentence, multiple passages for the single question are required to correctly locate the sentence with the answers. Lexical fields coupled with only syntactic fields do not perform much better. It may be due to a fact that the syntactic fields containing ordinary dependency labels do not provide sufficient context-wise information so that they do not generate enough features for statistical learning to capture the specific characteristic of the context. The significant improvement, however, is reached when the semantics fields are added as they provide the deeper understanding of the context.

This data set has also been used for evaluating the Memory Networks approach to question answering [154]. The authors achieved high accuracy, reaching 100% in several tasks; however, this work still finds its own value because the approach is completely data-driven such that it can be easily adapted or extended to other types of questions. As a matter of fact, the same system is used for all tasks with different trained models, yet still able to achieve high accuracy for most tasks it was evaluated on.

4.3 Summary

This chapter addressed two examples of non-factoid question types: arithmetic and event-based questions. Math and logical problems can be classified as complex questions that require dedicated solutions. Due to their complex nature, the approaches that are often applied to factoid questions, would not work on them. Semantics-based graph approach was presented and applied to solve arithmetic questions. The designed graph combines several natural language processing annotation outputs to construct the semantically-driven structure. The results on publicly available dataset prove that the structure can be successfully applied to complex questions. Next, a multi-field structural decomposition with average perceptron learning for question answering was presented. The algorithm first decomposes source texts to the structure that can be grouped into three layers: lexical, syntactic and semantic. When the system receives a question, it is decomposed in the same way as the source texts and the supporting sentence is returned. As a proof of concept, the system was tested on the publicly available non-factoid questions dataset and showed promising results.

Chapter 5

Applications to Cross-genre Tasks

Chapters 3 and 4 described two most common branches in question answering: factoid and non-factoid questions. In these branches, questions and the corpora from which answers are extracted from are usually of the same type. More recently, the human-computer interaction has attracted increasing attention from both research and commercial worlds. The ultimate goal is to build a conversation agent that will be able to handle human conversation in natural way. However, to build the natural language applications with the focus on human conversation, these applications must first perform comprehension and understanding of the human dialog.

Document retrieval has been a central task in natural language processing and information retrieval. The main goal is to retrieve the most relevant documents from a set of documents given any query. Most of the previous work in this field for question answering assumed homogeneity of source

(document) and target (query) texts. When entering the era of conversation agents and chatbots, this task will likely be extended to dialogues as well. Consider a collection of conversations and a person who would like to find the dialog that corresponds to: *"Brandon talked about Mary doing most of the tasks before the second version of the app is released"*. Script data which consists of human conversation poses several challenges in terms of natural language processing such as dialog disfluency, the speaker-utterance structure or short and long term context, to name a few.

The recent developments of neural architectures have allowed researchers to focus on more complicated tasks in natural language processing such as reading comprehension. The popularity of this task has resulted in the growing number of corpora that were released: MCTest [112], MS MARCO [100], and CNN/Daily News [60]. All these corpora are based on formal writings such as articles, stories, *etc.* In the era of conversation agents and chatbots, it will be crucial to research machine comprehension from the perspective of dialog data.

This chapter explores how text applications related to question answering perform when the cross-genre aspect is the main concern. First, Section 5.1 describes an approach based on structure matching for the document retrieval task. Relation extraction is performed using deterministic rules for conversation and formal writings. Then, the reranking process is performed using statistical learning based on the feed-forward neural architecture that leads to a significant 4% improvement. Next, the passage completion task regarding text understanding is explored in the domain of conversation data. Due to the lack of existing corpora in this aspect, the annotation is performed.

It results in a new cross-genre corpus for reading comprehension. The experimentation process using already existing methods and a multi-gram convolutional neural network with attention mechanism show promising results and lay the groundwork for passage completion in the cross-genre aspect.

5.1 Cross-genre Document Retrieval

This section analyzes the performance of state-of-the-art retrieval techniques targeting on TV show transcripts and their descriptions. First, a dataset is created by collecting transcripts from a popular TV show and their summaries and plots (Section 5.1.1). Then, a solid baseline by adapting an advanced search engine and propose structure reranking to improve the initial ranking from the search engine is established (Section 5.1.2).

5.1.1 Data

The Character Mining project provides transcripts of the TV show, *Friends*; transcripts from 10 seasons of the show are publicly available in the JSON format,[1] where the first 2 seasons are annotated for the character identification task [26]. Each season consists of episodes, each episode contains scenes, each scene includes utterances, where each utterance comes with the speaker information.

For each episode, the episode summary and plot are first collected from fan sites,[2] then sentence segmented by NLP4J,[3] the same tool used for the

[1] `nlp.mathcs.emory.edu/character-mining`

[2] `friends-tv.org`, `friends.wikia.com`

[3] `github.com/emorynlp/nlp4j`

Dialogue		**Summary + Plot**	
# of episodes	194	# of queries	5,075
# of tokens	897,446	# of tokens	119,624

Table 5.1: Dialogue, summary, and plot data.

provided transcripts. Generally, summaries give broad descriptions of the episodes, whereas plots describe facts within individual scenes. Finally, a dataset is created by treating each sentence as a query and its relevant episode as the target document. Table 5.1 shows the distributions of this dataset.

Alternatively, other data sources can be used as source texts in the cross-genre document retrieval. For instance, the twitter conversations can be crawled and used as multi-party conversations. However, in this case, the task of collecting the target texts (episode summaries and plots) would be more problematic.

5.1.2 Structure Reranking

For each query (summary or plot) in the dataset, the task is to retrieve the document (episode) most relevant to the query. The challenge comes from the cross-domain aspect: how to retrieve documents in dialogues given the queries in formal writing. This section describes the structure reranking approach that significantly outperforms an advance search engine, Elasticsearch, for this cross-domain task.

Dialogue		Summary + Plot
Joey	One woman? That's like saying there's only one flavor of ice cream for you. Lemme tell you something, Ross. There's lots of flavors out there.	Joey compares women to ice cream.
Ross Rachel	You know you probably didn't know this, but back in high school, I had a, um, major crush on you. I knew.	Ross reveals his high school crush on Rachel.
Chandler Joey	Alright, one of you give me your underpants. Can't help you, I'm not wearing any.	Chandler asks Joey for his underwear, but Joey can't help him out as he's not wearing any.

Table 5.2: Three examples of dialogues and their descriptions.

Relation Extraction

Since the queries and documents appear very different on the surface level (Table 5.2), relations are first extracted from them and matching is performed on the relation level, which abstracts certain pragmatic differences between these two types of writings. All data are lemmatized, tagged with parts-of-speech and named entities, parsed into dependency trees, and labeled with semantic roles using NLP4J.

A sentence may consist of multiple predicates, and each predicate comes with a set of arguments. A predicate together with its arguments is considered a relation. For each argument, heuristics are applied to extract meaningful contextual words by traversing the subtree of the argument. The heuristics are designed for the type of dependency trees generated by NLP4J, but similar rules can be generalized to other types of dependency trees. Relations from dialogues are attached with the speaker names to compensate the lack of entity information.

By extracting relations that comprise only meaningful words, it prunes out much noise (e.g., disfluency), which allows to retrieve relevant documents

with higher precision. While the relation extraction is based on the sentence level, it can be extended to the document level by adding coreference relations, which will be explored in the future.

Predicate-argument structures have already been successfully applied to several retrieval and classification tasks [125, 80, 56].

Predicate-argument groups. The process starts by using the output of dependency trees and semantic role labels to extract all predicate nodes with their semantic arguments. At this point, the algorithm extracts a list of relation groups, where each relation group is enclosed by the predicate's arguments. The number of these groups depends on how many predicates are located within a sentence.

Argument context extraction. Extracted relations from the previous step are then extended with their contextual backgrounds. While predicate-argument groups often represent most of the semantics, entire context is needed to improve the relevance ranking. For each argument, using a few simple deterministic rules the following elements are extracted: *nouns*, *compounds*, *prepositions*, *conjunctions* and *adjectives*. The subtree traversal is limited to look up only as deep as to grandchildren to prevent node overlapping with other semantic groups.

Named entity completion. The final step is to extract all recognized named entities and add them to the corresponding relations groups. The dependency tree is traversed upwards from the named entity node to the closest predicate node. All intermediate nodes that are not present in this group are added.

Input: D: a list of documents, q: a query.

f_r: a function returning all relations.

f_c: a comparator function.

Output: S: a list of matching scores for D.

$S \leftarrow [0 \text{ for } i \in [1, |S|]$

foreach $d_i \in D$ **do**

  foreach $r^q \in f_r(q)$ **do**

    $R^d \leftarrow [r \text{ for } r \in f_r(d_i) \text{ if } |r \cap r^q| \geq 1]$

    $s_m \leftarrow 0$

    foreach r^d *in* R^d **do**

      $s \leftarrow f_c(r^d, r^q)$

      $s_m \leftarrow \max(s_m, s)$

    end

    $S_i \leftarrow S_i + s_m$

  end

end

Algorithm 4: The structure matching algorithm.

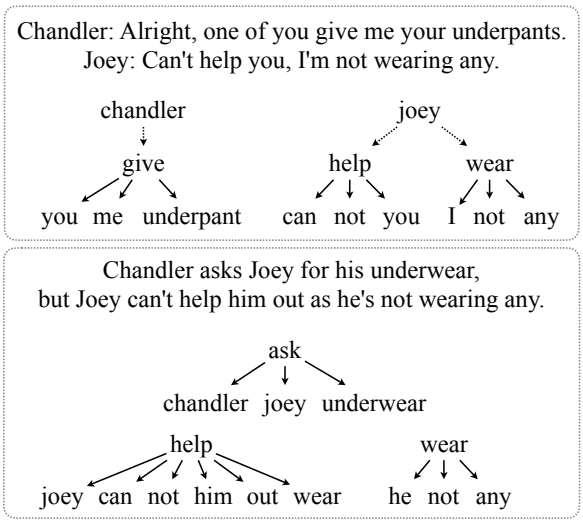

Figure 5.1: Two sets of relations, from dialogue and plot, extracted from the examples in Table 5.2.

Structure Matching

All relations extracted from dialogues are stored in an inverted index manner, where words in each relation are associated with the relation and the episode that the relation is from. Algorithm 4 shows how the structure matching works. Given a list of documents and a query q, it first initializes scores for all documents to 0. For each document d_i, it compares each relation r^q from q to relations extracted from d_i. The relation r from d_i is kept to R^d if it has at least one word overlaps with r^q. For each relation $r^d \in R^d$, the comparator function returns the matching score between r^d and r^q. The maximum matching score is added to the overall score of this document. This procedure is repeated; finally, the algorithm returns the overall matching scores for all documents.

The comparator function f_c takes two relation sets, r^d and r^q, and returns

the matching score between those two sets. For *word* and *lemma*, the count of overlapping words between them is used to produce two scores, r_s^d, and r_s^q, normalized by the length of the utterance and the query, respectively. The harmonic mean of the two scores is then returned as the final score. For *embedding*, f_c uses embeddings to generate sum vectors from both sets and returns the cosine similarity of these two vectors.

Document Reranking

The Elasticsearch scores and the 3 sets of matching scores for the top-k documents (ranked by Elasticsearch) are fed into a binary classifier to determine whether or not to accept the highest ranked document. A Feed Forward Neural Network with one hidden layer of size 15 is used for this classification. If the binary classifier disqualifies the top-ranked document, the top-k documents are reranked by the weighted sums of these scores. A grid search is performed on the development set to find the optimized set of the weights. At last, the system returns the document with the highest reranked scores: $d_i = \arg\max_i(\lambda_e \cdot e_i + \lambda_w \cdot w_i + \lambda_l \cdot l_i + \lambda_m \cdot m_i)$.

5.1.3 Experiments

The data from Section 5.1.1 is split into training, development and evaluation sets, where queries from each episode are randomly assigned. Two standard metrics are used for evaluation, recall at k (R@k) and mean reciprocal rank (MRR).

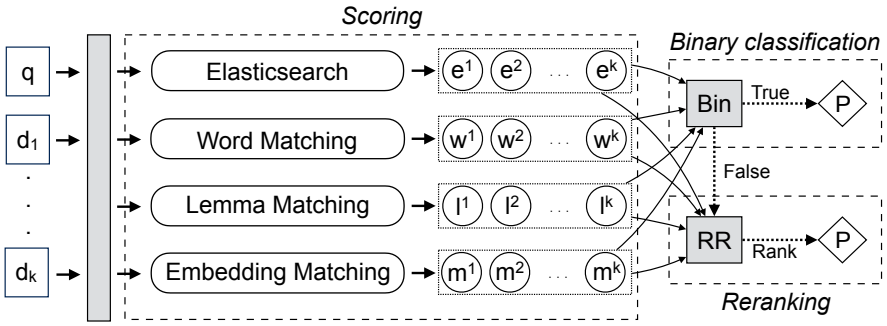

Figure 5.2: The overview of the prediction process. Given documents $d_1, \ldots, d_k$ and a query q, 4 sets of scores are generated: the Elasticsearch scores and the matching scores using 3 comparators: *word*, *lemma*, and *embedding*. The binary classifier `Bin` predicts whether the highest ranked document from Elasticsearch is the correct answer. If not, the system `RR` reranks the documents using all scores and returns a new top-ranked prediction.

Elasticsearch

Elasticsearch is used to establish a strong baseline.[4] Each episode is indexed as a document using the default setting, Okapi BM25 [115] and the TF-IDF based similarity with improved normalization, and the top-k most relevant documents are retrieved for each query. While R@1 is below 50% (Table 5.4), R@10 shows over a 70% coverage such that it is possible to achieve a higher R@1 by reranking results from $k \geq 10$.

Convolutional Neural Network

Neural network architectures have already been successfully applied to multiple tasks in text applications. Inspired by the fact of nonlinearity in the

[4] `www.elastic.co/products/elasticsearch`

Dataset	Summary	Plot	Total
Training	970	3,013	3,983 (78.48%)
Development	97	403	500 (9.85%)
Evaluation	150	442	592 (11.67%)

Table 5.3: Data split (# of queries).

k	Development		Evaluation	
	R@k	MRR	R@k	MRR
1	**46.00**	46.00	**47.64**	47.64
5	65.80	53.80	69.26	69.26
10	72.60	**54.71**	74.66	**56.53**
20	78.80	55.13	79.73	56.91
40	83.80	55.31	84.80	57.08

Table 5.4: Elasticsearch results on (summary + plot).

problem, a convolutional neural network model is built. Elasticsearch *top*10 retrieval ranking is used to generate three splits for the model. The problem is reformulated as a binary classification: given a sample s of a query q and episode text e, predict whether the query is supported by this episode. The sample is represented as a sequence of 25 scenes where each scene is padded to 200 words, and a sequence of words from the query. The idf scores of all the scene words are used to select the ones that most likely hold any semantic value. The model consists of three convolution layers with *tanh* activation function and max pooling. First two are applied on the sequence of scenes, and then on the scene vectors' matrix, respectively, producing an episode vector. The last convolution outputs the query vector given the sequence of words in the query. Finally, a dot product of these two vectors is taken and the model is trained using binary crossentropy. Experimentation setup showed *recall@1* of 20.76% and 18.75% for the development and evaluation sets, respectively. It is likely that such disproportion in the scores between deep learning and lexical approaches is caused by the lack of training data and an enormous amount of noise that the neural architecture is unable to address.

Structure Matching

The Struct$_*$ rows in Table 5.6 show the results based on structure matching (Section 5.1.2). The highest R@1 of 39.53% is achieved on the evaluation set using lemmas. Although it is about 8% lower than the one achieved by Elasticsearch, the hypothesis can be made that this approach can correctly retrieve documents for certain queries that Elasticsearch cannot.

Model	Development		Evaluation	
	R@1	**MRR**	**R@1**	**MRR**
Elastic_{10}	0	16.07	0	16.99
Struct_{w}	14.44	23.57	19.68	28.11
Struct_{l}	14.81	**25.59**	**20.97**	**30.14**
Struct_{e}	**15.56**	24.47	20.32	29.22

Table 5.5: Results on queries failed by Elasticsearch.

To validate the hypothesis, structure matching on the subset of queries failed by Elasticsearch is tested. First, the top-10 results from Elasticsearch are taken and then reranked using the scores from structure matching for queries that Elasticsearch gives R@1 of 0%. As shown in Table 5.5, structure matching is capable of reranking a significant portion (around 20%) of these queries correctly, establishing that the hypothesis is true.

Document Reranking

The scores from Elastic_{10} and Struct_{*} for each document are fed into the binary classifier that decides whether or not to accept the top-1 result from Elasticsearch. If not, the documents are reranked by the weighted sum of these scores (Section 5.1.2). The Rerank_{1} row in Table 5.6 shows the results when all the weights = 1, which gives an over 4% improvement of R@1 on the evaluation set. The Rerank_{λ} row shows the results when the optimized weights are used, which gives an additional 3% boost on the development set but not on the evaluation set.

Model	Development						Evaluation					
	Summary		Plot		All		Summary		Plot		All	
	R@1	MRR	R@1	MRR	R@1	MRR	R@1	MRR	R@1	MRR	R@1	MRR
Elastic_{10}	44.33	53.64	46.40	54.97	46.00	54.71	50.67	60.87	46.61	55.06	47.64	56.53
Struct_{w}	38.14	48.42	34.00	45.11	34.80	45.75	35.33	48.34	35.52	47.08	35.47	47.40
Struct_{l}	39.18	49.24	34.74	46.29	**35.60**	**46.86**	44.00	55.55	38.01	49.24	**39.53**	**50.84**
Struct_{m}	35.05	46.71	33.50	44.72	33.80	45.10	36.00	50.14	35.97	46.95	35.98	47.76
Rerank_{1}	47.42	55.66	48.39	56.10	48.20	56.02	56.67	63.77	50.23	57.99	51.86	59.46
Rerank_{λ}	50.52	57.66	51.36	57.76	**51.20**	**57.74**	55.33	63.88	50.90	58.47	**52.03**	**59.84**

Table 5.6: Evaluation on the development and evaluation sets for summary, plot, and all (summary + plot). Elastic_{10}: Elasticsearch with $k = 10$, $\text{Struct}_{w,l,m}$: structure matching using words, lemmas, embeddings, $\text{Rerank}_{1,\lambda}$: unweighted and weighted reranking.

5.2 Cross-genre Passage Completion

In this section, the passage completion task is explored in the cross-genre data aspect. The motivation for this task comes from the recent growth of interest in conversation agents. These systems will have to understand the human conversation and extract knowledge from it. The cross-genre passage completion task presented in this work is a first and major step towards question answering based on conversational data. An annotation is performed to clean the data and prepare the passage completion configuration. Next, an experimentation is performed using already existing methods for reading comprehension. Also, a multi-gram convolution-based network with attention is presented showing a good compromise between training speed and accuracy.

5.2.1 Data

The *"Friends"* scripts corpus presented in Section 5.1 is used to create a new dataset that will be dedicated for passage completion. All the queries that have been used to generate the document retrieval corpus are candidates for the new dataset. Each query is associated with a single episode. The entire episode is likely too large and contains an enormous amount of noise to be used as an input in the passage completion task (Section 5.1.3). Unfortunately, the current version of dataset lacks the scene-association.

Not all queries are valid and should be included in the final dataset. Consider the query *"Now everybody knows, except Ross."*. In this case, it would be almost impossible for the model to comprehend the entity *Ross* from the dialogue. On the other side, consider the query *"When she picks up still another, Chandler isn't sure he can take it."*. It contains one proper noun that is correctly recognized by the named entity recognizer (*Chandler*) and two pronouns ((*she*, *he*). When given this description with the masked *Chandler* position, the query does not hold enough information to be inferred from the text. The pronoun *she* should be resolved to give necessary context to the query. Moreover, additional pronoun annotation would allow generating more passage completion samples. These challenges must be addressed before constructing the corpus to ensure the highest possible quality of the dataset.

Annotation

A crowdsourcing annotation is performed to address the described challenges. Two separate tasks are handled on Amazon Mechanical Turk in order to

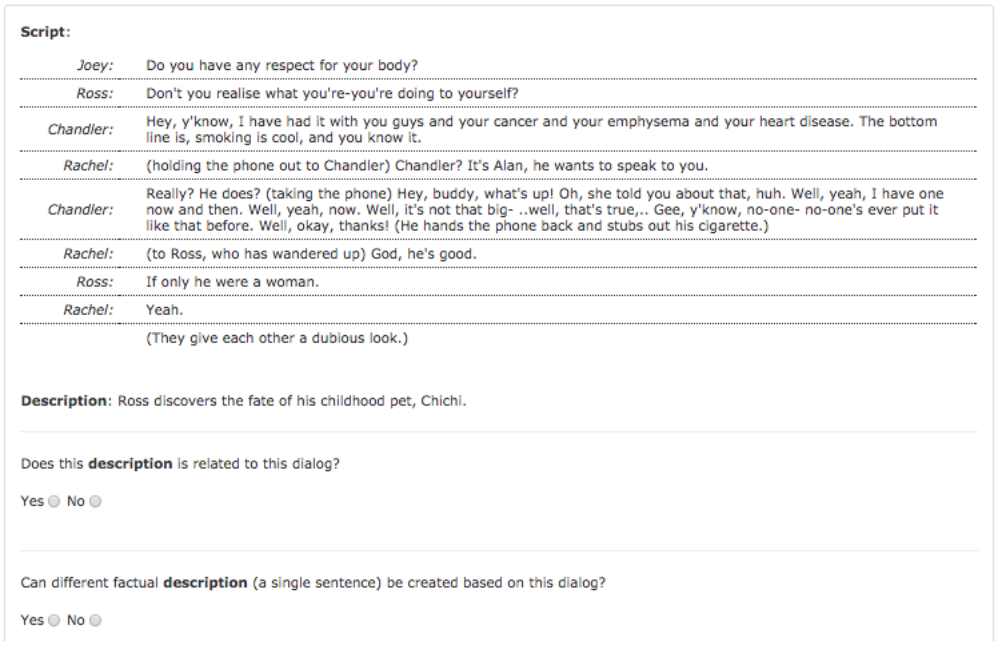

Figure 5.3: Example of *Task 1* of the annotation.

prepare the set of queries for the passage completion configuration.

Task 1 is designed to associate each query with its scene and validate the query appropriateness. Given a query and all the scenes from its episode (the episode-association is known), Elasticsearch is used to select the scene that has the closest lexical distance to the query. The empirical analysis shows that this method can correctly locate the scene in the majority of cases. Next, annotators are presented the query and its selected scene from Elasticsearch. The first asked question to them is: *"Does this description (question) is related to this dialog?"*. If the annotator chooses *yes*, she is asked to rephrase the pronouns with their proper nouns, if possible. Otherwise, the annotator is asked to rephrase or recreate the query to the form when it will relate to the given dialog. The second question to the annotator

is: *"Can different factual description (a single sentence) be created based on this dialog?"*. If the annotator chooses *yes*, she is asked to enter a new query. Figure 5.3 presents one of the annotation tasks. Please note that the instructions are not covered in this figure but are visible to the annotator.

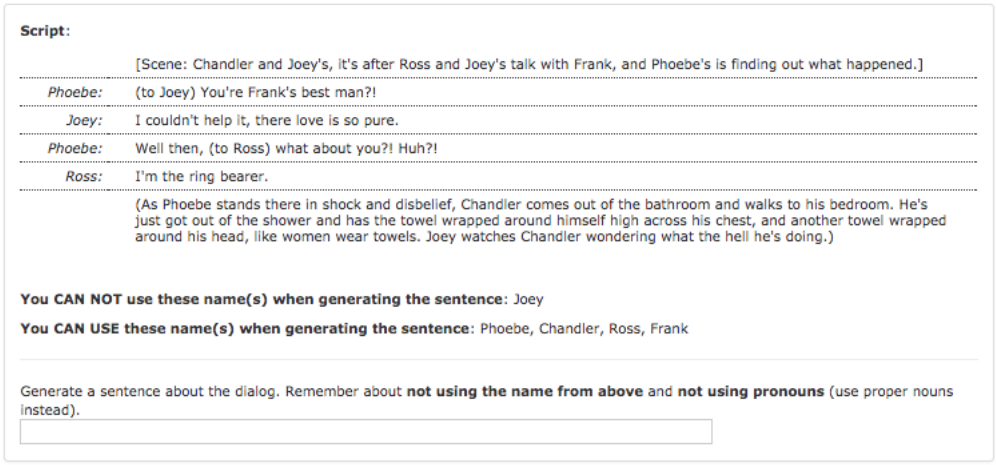

Figure 5.4: Example of *Task 2* of the annotation.

Task 2 is designed to provide more contextual diversity to the dataset and to make the queries more complex. An annotator is given a scene along with the list of speaker(s) that the annotator cannot use when forming a query. Then, the annotator is asked to create a new description (sentence) about the events in the given scene. Excluded name(s) consists of the most frequent speakers in the scene or the names that have already been used in the previous task. Figure 5.4 presents one of the annotation tasks.

Table 5.7 shows the lexical and semantic analysis of created corpus.

number of queries	5,055
number of samples	13,380
# of unique entities in a scene	19
min/avg/max entities per query	1.00/2.95/15.00
min/avg/max entities per scene	5.00/25.39/116.00
min/avg/max query length	1.00/20.05/126.00
min/avg/max scene length	52.00/312.55/1413.00

Table 5.7: Statistics of the generated corpus

Dataset

Both tasks have been executed for over 6,000 queries[5] (Section 5.1.1). The post-annotation analysis has shown that cleaning process is necessary. For instance, some of the characters are often mentioned using various pseudonyms within a single scene. These exceptions have been handled by manually generating the mappings of pseudonyms to the proper character names. The anonymization approach originally applied in the CNN/Daily News dataset is followed in the new corpus. All the entities within a scene and query are anonymized. The goal is to limit the bias of the main characters (*Chandler, Joey, Monica, Phoebe, Rachel, Ross*, who dominate the data. Figure 5.5 shows the example after the anonymization process is applied. The token `@placeholder` is used to mark the entity that is to be predicted.

[5]Additional queries for seasons 9 and 10 have been collected.

Query: **@placeholder** and @ent15 try to return a cat to Mr. @ent00.

@ent12:	(stops at a door) Oh no, the @ent07, they hate all living things, right?
@ent15:	Oh. (they knock at the next door, Mr. @ent00 answers) Hi. We just found this cat and we're looking for the owner.
@ent06:	Er, yeah, it's mine.
@ent12:	(trying to hold back the struggling cat) He seems to hate you. Are you sure?
@ent06:	Yeah, it's my cat. Give me my cat.
@ent12:	Wait a minute. What's his name?
@ent06:	@ent14 ... B - Buttons.
@ent15:	@ent04?
@ent06:	Mmm. @ent10 Buttons. Here, @ent04.
@ent12:	(the cat runs away from her) Oooh! You are a very bad man!
@ent06:	(as @ent12 and @ent15 leave) You owe me a cat.

Answer: @ent12

Figure 5.5: One of samples in the created dataset.

5.2.2 Approach

Convolutional neural networks have already been successfully applied to several text-based classification tasks. These approaches can learn to recognize the common patterns across space rather than trying to find sequential information as in recurrent neural networks. These patterns will be generic in a sense that the will be identical across the entire input space. It is likely that in the cross-genre passage completion task, the patterns that provide the semantic information necessary to solve this task are generic in their, and therefore this approach is promising. Moreover, a convolutional approach will allow the work to be extended in the future to the form when an input

will be a sequence of utterances rather that a sequence of tokens in a scene. Additionally, convolutional neural networks are significantly faster to train, which for the amount of data currently available, will be crucial.

All of the above is the motivation to build a multi-gram convolutional neural network approach with attention that is dedicated to the cross-genre passage completion. The attention mechanism is inspired by the work presented by Santos et al. [46]. Figure 5.6 shows the developed architecture. The task is the following: given a query with the encoded placeholder and a scene containing a set of entities, complete the query with the entity from the scene. The input to the network consists of two text images: a scene and a query: S and Q. Multi-gram filters are then convoluted through the input resulting in the feature maps. The feature maps that participate in the attention are multiplied with an attention weights $attn_m$ which results in the attention matrix.

$$attention = tanh(S^T \cdot attn_m \cdot Q)$$

Vertical and horizontal max pooling is applied to the attention matrix forming two attention vectors: S_{attn} and Q_{attn}. These vectors are then multiplied with the original feature maps which result in two attended vectors. The vectors are merged with pooled feature maps of other n-gram filter maps, forming one large $1D$ vector. This vector is then projected into the vectors size of a maximum number of entities in a single scene from the entire dataset.

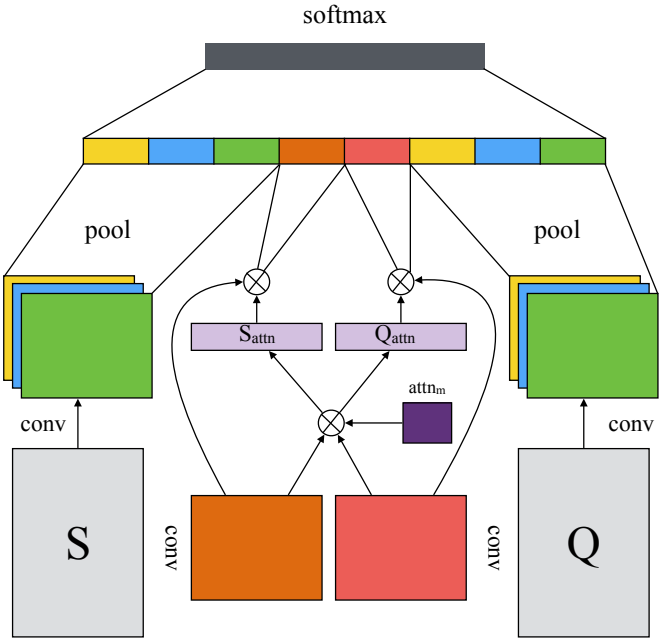

Figure 5.6: The architecture of designed convolutional neural network system.

5.2.3 Experiments

The corpus is divided into training, development and test sets with the the ratio of 80/10/10. The split is performed on the episode-level, where each episode's queries are randomly divided into training, development and test sets. Table 5.8 presents the experimentation results.

At the beginning, two deterministic baselines are evaluated. $Baseline_1$ is based on the entity majority where the most frequent entity from the scene is chosen as a prediction. $Baseline_2$ comprises of the word distance based model. The entity that has the closest lexical distance to the query words is predicted. Both baselines do not perform well showing that this task is challenging and requires more advanced approaches. The linguistic approach

Baseline_1	27.30%
Baseline_2	27.26%
Linguistic approach (L2R) [23]	51.16%
Bi-LSTM + attn [23]	**69.26**% (test: 65.52%)
Multi-gram CNN	57.43%
Multi-gram CNN + attn_{word} [128]	62.45%
Multi-gram CNN + attn_{dot}	**63.58**%

Table 5.8: The development set accuracy for the passage completion task using different approaches.

is a pairwise learning to rank system based on manually crafted lexical and semantic features [23]. Features such as frequency and positions of entities, the n-gram matches, and the dependency parse matches are used. Despite the fact that this approach performs significantly better than the baselines, there is still large room for improvement.

Now, the neural networks based approaches are evaluated. First, a bi-directional long-short term memory [23] is reimplemented and tested. It achieves the highest score of 69.26% proving that the deep learning architectures can provide powerful and robust models for reading comprehension tasks. The neural network model shows its power by extracting the semantic information that the previous approaches could not. Next, the multi-gram convolutional neural network without the attention is tested and reaches 57.43% which is significantly lower than the previous approach. Its poor performance could be caused by the lack of attention mechanism. Therefore, a multi-gram model with the attention presented by Shin et al. [128] is tested,

reaching the score of 62.45%. Finally, the convolutional neural network with the attention described in Section 5.2.2 is evaluated. It shows the score of 63.58%, which is 6% higher than the original convolutional neural network, and over 5% lower than the bi-directional LSTM. It is important to point out that the CNN model is on average seven times faster to train. If this model is further improved, it has the potential reaching the state of the art presented by the recurrent network and even surpassing it. The advancements will concern more advanced attention mechanisms. Also, the network currently takes an entire scene as one document. It is possible that further improvements could be achieved by addressing the internal structure of the dialog: a mix of speakers and utterances.

5.3 Summary

This chapter explored two tasks in the cross-genre text configuration: document retrieval and passage completion. The main goal of the document retrieval task is to select the episode related to the descriptive query 5.1. The challenge comes from the cross-genre aspect which has not been a focus in the previous work. Structure matching technique has been introduced that addresses the misalignment between the scripts and queries. A significant improvement has been shown when the structure matching features were used to rerank the candidates. In the second part, the passage completion task has been explored. This task is designed to complete the descriptive query with the context of the scene. Several reading comprehension approaches were used to benchmark this task. Also, a multi-gram CNN with the attention

mechanism was presented and evaluated on this dataset showing a promising result. The future work consists of more advanced attention mechanism and restructuring the input to reflect the internal structure of human dialogue.

Chapter 6

Conclusions and future work

6.1 Summary

In the last decade, researchers have revolutionized the way of answering any questions we might have. Recent advancements in the question answering field have allowed us to retrieve information significantly faster and easier. While this field has already seen a tremendous growth, several challenges still remain. This thesis proposed numerous approaches and techniques to improve the overall performance of various question answering systems.

Chapter 3 focused on major branch of question answering - factoid questions. A multi-scheme annotation scheme was presented in which the primary goal is to let researchers build diverse and challenging corpora for open-domain question answering. Neural architectures that are currently a common choice for researchers often require an enormous amount of data to be trained. This data can be built in an unsupervised way by collecting large corpora from the web. However, to allow these networks learn the con-

textual similarity measurements, they must also be exposed to direct signals found in supervised data collections. The presented scheme provides an easy and low-cost crowdsourcing-based framework that can be used by researchers to build such corpora. The analysis shows that by performing five sequential annotation tasks, it is possible to build a corpus that is both diverse and artificially-difficult.

Neural architectures have been shown to be extremely useful in finding contextually similar contexts in the open-domain configurations. On the other side, the advancements in the natural language processing field, especially in parsing, allow researchers to use these extracted structures for various problems. A subtree matching mechanism that is based on any syntactic structures has been introduced. It is based on the tree slice-by-slice comparisons rather than on the tree-edit distance paradigm. Because of this fact, it is more computationally efficient and faster. When paired with the extended state-of-the-art convolutional neural network, it showed consistent boosts in the performance, proving it is capable of extracting contextual similarity.

The growth of popularity in question answering has caused that multiple corpora for various tasks have been released. Although their main purpose is to help training statistical models and benchmarking their performances separately, they can be combined which potentially would lead to other improvements. A thorough intrinsic and extrinsic analysis on recently published open-domain question answering corpora has been presented. First, the passage retrieval task was applied that allowed the existing datasets to be combined. The experiments using the-state-of-the-art convolutional neural network showed promising results grounding the future work for transfer

learning in this field.

Chapter 4 addressed existing challenges in non-factoid question answering. Math and logical problems such as arithmetic questions require dedicated approaches that are capable of extracting abstract representations of text. A semantic-based graph approach was introduced that combines linguistic structures from currently existing natural language processing tasks. Arithmetic questions were used as a case study to show the potential of designed structure. The evaluation showed that the graph can be used to extract the abstract representations of text.

Researchers in natural language processing and information retrieval often tackle the problems in question answering form either sides. Section 4.2 described a multi-field structural decomposition method that combines good aspects of NLP and IR. The system decomposes linguistic structures into multiple fields and organizes them in an inverted index manner. When the system receives a question, it decomposes it according to the same rules and performs the retrieval step, ranking the retrieved sentences with previously learned weights. The experimentation on eight tasks of the event-based question answering tasks showed significant improvements over the baseline that uses simple search.

Chapter 5 explored two tasks in the cross-genre aspect of text: document retrieval and passage completion. The main challenge in these tasks concerns the misalignment and differences between the conversational and formal writings. For the document retrieval task, a reranking mechanism based on structure matching was presented. The algorithm deterministically extracts relations that comprise of only most important words from the text.

The algorithm then improves the initial ranking by incorporating the semantic matching features. The evaluation showed an improvement of over 4% in precision on the dataset that has been created during this work.

In addition to the document retrieval task, this dissertation explores a subtask in machine comprehension called passage completion. It will be an integral part of the future conversation dialog systems to be able to understand and extract contexts from the human dialog. At the beginning, a crowdsourcing-based annotation was performed. It resulted in the first passage completion dataset that is based on the conversational data. The evaluation using current architectures showed the challenging nature of this task. A multi-gram convolutional neural network with attention mechanism showed a promising result. More importantly, it is faster to train by an order of magnitude than the state-of-the-art recurrent neural network approach.

6.2 Limitations

Chapter 3 was focused on factoid question answering. The generation method presented in this thesis is based on a low-cost crowdsourcing scheme. It is expected then that annotations are biased towards high overlapping lexical tokens between answer sentences and questions. While extra precautions have been taken, in some cases these questions can still be artificially easy for a machine. Also, currently the scheme is based on only a single paraphrase of questions. Presented subtree matching algorithm is currently based on tree slices that have been extracted directly from the overlapping words. Thus, the aspect of synonyms and semantically-related words is not addressed,

which might often be a case when a question and candidates do not share any words. On the other hand, by considering only parent, siblings, and children, the algorithm may omit the context that could be located in a different part of the sentence. The expected input to the designed convolution neural network considers lists of tokens which almost always are sentences. The text structures such as listings, bullet points, *etc.*, are special cases and should be handled because the analysis has shown that they were often missed by the system. The convolution neural network expects a constant size of the input which can be problematic when used in various open-domain text collections. More precisely, a larger input will allow longer sentences to be fully exposed to the network, but it will likely slow down the training time and create more noise. On the other side, the shorter input might cause that crucial information will not be included in the input, but it will speed up training. The cross-evaluation of currently existing question answering corpora is based on the fact that the network is fully trained on one corpus and evaluated on another. This assumption means that any potential transfer of knowledge from one dataset to another is limited.

Chapter 4 presented two approaches for different problems in non-factoid question answering. A semantics-based graph that was introduced is a combination of the representations coming from several natural language processing tasks. Since the current process of building the graph is deterministic, it is likely that it cannot be easily applied to other branches. Also, the framework is based on the output from specific parsers making it less flexible and customizable. The system presented for event-based questions is based on a sentence level and thus would not work if the answer is sparse. Also,

the system currently selects the entire sentence, which might contain a large amount of irrelevant information to the question.

Two approaches for the document retrieval and passage completion tasks in the cross-genre text domains have been presented. For the document retrieval task, the relation extraction process was introduced that is currently based on extracting only pure word-forms from the linguistic structures. Therefore, it misses all the additional annotation that is given by the parsers. Also, the approach currently extracts relations only from a single utterance, missing the cross-utterance aspect. The passage completion task is based only on the entities that can be extracted from both speaker information and the named entity tags. The latter periodically miss tags, which is common in case of pseudonyms. The current state of the art in this task is based on a recurrent neural network that does not incorporate the internal structure of the dialogue.

6.3 Future work

The thesis presented several advancements to already existing systems and introduced new approaches that push the current research in question answering forward. Despite the work and impact described in this thesis, there is still room for future work.

Future work in factoid question answering could be focused on improving the annotation scheme. First, more paraphrases could be performed in sequence instead of a single paraphrase. Also, the dynamics of the annotation process could be improved. When an annotator is generating a question, the

backend checks whether the overlapping ratio is too high after every typed word. Additionally, during the question generation, the backend could compare the generated question with already created questions and point out when it is too similar. Next, the subtree matching algorithm could be expanded to a wider contextual window. For instance, the algorithm could be based on the n backward and forward words, which would address the aspect of semantic similarity better. Moreover, the slice extraction process could be extended to support not only overlapping words. The convolution network used in the experiments could be extended with various attentions mechanisms. Also, the matching algorithm could play a role in the attention mechanism. For instance, contextually similar fragments (tree slices) from sentences could be exposed to the network more explicitly. Finally, more advanced transfer learning methods could be applied in the cross-evaluation setup. Researchers have already started looking into such combinations and the preliminary results are promising [90].

Non-factoid question answering often requires dedicated and semantic-driven approaches. A semantics-based graph presented in this work could be potentially merged with the Abstract Meaning Representation. At the moment, the semantics-based graph can only be applied to relatively short text documents. Extending it to larger documents such as an article or even a collection of documents would improve its flexibility. The multi-field approach described in this thesis was evaluated on the event-based type of questions. While the system has performed well during the experimentation, it is currently based on the single sentence basis. It is worth to extend the system to handle larger documents as well.

In the era of growing popularity of conversation agents, the fields of natural language processing, information retrieval, and machine learning will have to address several new challenges. In the structural matching approach presented in this thesis, the future work will consist of extending the graph structure to a more advanced form. The edges will have different weights that will be based explicitly on the syntactic and semantic relations. Also, the structure could be extended to cover larger contexts, in this case, the scenes and episodes. In the passage completion task, it is worth to incorporate the internal structure of human dialog into the model architecture. Finally, the attention mechanism should be directly impacted by the actual speakers of each utterance. This would likely help the network to comprehend the human dialog.

Bibliography

[1] Steven Abney, Michael Collins, and Amit Singhal. Answer extraction. In *Proceedings of the sixth conference on Applied natural language processing*, pages 296–301. Association for Computational Linguistics, 2000.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[3] PM Athira, M Sreeja, and PC Reghuraj. Architecture of an ontology-based domain-specific natural language question answering system. *International Journal of Web & Semantic Technology*, 4(4):31, 2013.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.

[5] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of*

the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, pages 86–90. Association for Computational Linguistics, 1998.

[6] Jaroslaw Baliński and Czeslaw Daniłowicz. Re-ranking method based on inter-document distances. *Information processing & management*, 41(4):759–775, 2005.

[7] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, 2013.

[8] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6, 2013.

[9] Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. Modeling Biological Processes for Reading Comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 1499–1510, Doha, Qatar, October 2014. Association for Computational Linguistics.

[10] Philip Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1):217–239, 2005.

[11] Matthew W Bilotti, Jonathan Elsas, Jaime Carbonell, and Eric Nyberg. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 459–468. ACM, 2010.

[12] David C Blair and Melvin E Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3):289–299, 1985.

[13] Phil Blunsom, Edward Grefenstette, Nal Kalchbrenner, et al. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[14] Daniel G. Bobrow. A question-answering system for high school algebra word problems. In *Proceedings of the October 27-29, 1964, Fall Joint Computer Conference, Part I*, AFIPS '64 (Fall, part I), pages 591–614, New York, NY, USA, 1964. ACM.

[15] J Kathryn Bock. The effect of a pragmatic presupposition on syntactic structure in question answering. *Journal of Verbal Learning and Verbal Behavior*, 16(6):723–734, 1977.

[16] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD in-*

ternational conference on Management of data, pages 1247–1250. AcM, 2008.

[17] James P Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310. Springer-Verlag New York, Inc., 1994.

[18] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM, 2006.

[19] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[20] Raquel Cerdán, Eduardo Vidal-Abarca, Tomás Martínez, Ramiro Gilabert, and Laura Gil. Impact of question-answering tasks on search processes and reading comprehension. *Learning and Instruction*, 19(1):13–27, 2009.

[21] Yllias Chali and Shafiq R Joty. Selecting sentences for answering complex questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 304–313. Association for Computational Linguistics, 2008.

[22] Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*, pages 1–24, 2011.

[23] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany, August 2016. Association for Computational Linguistics.

[24] Hsinchun Chen. Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms. *Journal of the Association for Information Science and Technology*, 46(3):194–216, 1995.

[25] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015.

[26] Yu-Hsin Chen and Jinho D. Choi. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles, September 2016. Association for Computational Linguistics.

[27] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[28] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[29] Jinho D. Choi. *Optimization of Natural Language Processing Components for Robustness and Scalability*. PhD thesis, University of Colorado Boulder, 2012.

[30] Jinho D. Choi and Andrew McCallum. Transition-based Dependency Parsing with Selectional Branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL'13, pages 1052–1062, 2013.

[31] Jinho D. Choi and Andrew McCallum. Transition-based Dependency Parsing with Selectional Branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL'13, pages 1052–1062, Sofia, Bulgaria, August 2013.

[32] Jinho D. Choi and Martha Palmer. Transition-based Semantic Role Labeling Using Predicate Argument Clustering. In *Proceedings of ACL workshop on Relational Models of Semantics*, RELMS'11, pages 37–45, 2011.

[33] Jinho D. Choi and Martha Palmer. Fast and Robust Part-of-Speech Tagging Using Dynamic Model Selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL'12, pages 363–367, Jeju Island, Korea, July 2012.

[34] Jinho D. Choi and Martha Palmer. Guidelines for the Clear Style Constituent to Dependency Conversion. Technical report, Technical Report 01-12, University of Colorado at Boulder, 2012.

[35] Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016.

[36] William S. Cooper. Fact retrieval and deductive question-answering information retrieval systems. *J. ACM*, 11(2):117–137, April 1964.

[37] W Bruce Croft and David J Harper. Using probabilistic models of document retrieval without relevance information. *Journal of documentation*, 35(4):285–295, 1979.

[38] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 400–407. ACM, 2005.

[39] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.

[40] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[41] Dina Demner-Fushman and Jimmy Lin. Answer extraction, semantic clustering, and extractive summarization for clinical question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 841–848. Association for Computational Linguistics, 2006.

[42] Shibhansh Dohare and Harish Karnick. Text summarization using abstract meaning representation. *CoRR*, abs/1706.01678, 2017.

[43] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over freebase with multi-column convolutional neural 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, pages 260–269, 2015.

[44] Cícero Nogueira Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.

[45] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *CoRR, abs/1602.03609*, 2016.

[46] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *CoRR, abs/1602.03609*, 2016.

[47] Cícero Nogueira dos Santos, Kahini Wadhawan, and Bowen Zhou. Learning loss functions for semi-supervised learning via discriminative adversarial networks. *CoRR*, abs/1707.02198, 2017.

[48] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research*, 12(39):2121–2159, 2011.

[49] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-Driven Learning for Open Question Answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL'13, pages 1608–1618, 2013.

[50] Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. Applying Deep Learning to Answer Selection: A Study and An Open Task. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 813–820, 2015.

[51] Raquel Fernández, Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and David Schlangen. Reciprocal learning via dialogue interaction: Challenges and prospects. In *Proceedings of the IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT 2011)*, 2011.

[52] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79, 2010.

[53] Annika Flycht-Eriksson and Arne Jönsson. Some empirical findings on dialogue management and domain ontologies in dialogue systems - implications from an evaluation of birdquest. In Akira Kurematsu, Alexander Rudnicky, and Syun Tutiya, editors, *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, pages 158–167, 2003.

[54] Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[55] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.

[56] Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*, pages 643–653, 2015.

[57] Zhiyang He, Xien Liu, Ping Lv, and Ji Wu. Hidden softmax sequence model for dialogue structure analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2063–2072, Berlin, Germany, August 2016. Association for Computational Linguistics.

[58] Michael Heilman and Noah A. Smith. Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT'10, pages 1011–1019, 2010.

[59] Michael Heilman and Noah A Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics, 2010.

[60] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

[61] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[62] Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. Learning knowledge graphs for question answering through conversational dialog. In *HLT-NAACL*, pages 851–861, 2015.

[63] Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. Learning Knowledge Graphs for Question Answering through Conversational Dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL'15, pages 851–861, 2015.

[64] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivize High Quality Crowdwork. In *Proceedings of the 24th World Wide Web Conference*, WWW'15, 2015.

[65] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

[66] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to Solve Arithmetic Word Problems with Verb Categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 523–533, Doha, Qatar, October 2014.

[67] Eduard Hovy, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Using knowledge to facilitate factoid answer pinpointing. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[68] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 633–644, 2014.

[69] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics, 2011.

[70] Tomasz Jurczyk, Michael Zhai, and Jinho D. Choi. SelQA: A New Benchmark for Selection-based Question Answering. In *Proceedings of the 28th International Conference on Tools with Artificial Intelligence*, ICTAI'16, 2016.

[71] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*, 2016.

[72] Seokhwan Kim, Rafael Banchs, and Haizhou Li. Exploring convolutional and recurrent neural networks in sequential labelling for dialogue topic tracking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Berlin, Germany, August 2016. Association for Computational Linguistics.

[73] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[74] Oleksandr Kolomiyets and Marie-Francine Moens. A Survey on Question Answering Technology from an Information Retrieval Perspective. *Information Sciences*, 181(24):5412–5434, 2011.

[75] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.

[76] Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. Learning to Automatically Solve Algebra Word Problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL'14, pages 271–281, Baltimore, Maryland, June 2014.

[77] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Advances in neural information processing systems*, pages 1243–1251, 2010.

[78] Willem JM Levelt and Stephanie Kelter. Surface form and memory in question answering. *Cognitive psychology*, 14(1):78–106, 1982.

[79] Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, COLING '02, pages 1–7, 2002.

[80] Zuyao Li, Peter Exner, and Pierre Nugues. Using semantic role labeling to predict answer types. In *Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 29–31. ACM, 2014.

[81] Qiaoling Liu, Tomasz Jurczyk, Jinho Choi, and Eugene Agichtein. Real-time community question answering: Exploring content recommendation and user notification strategies. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 50–61. ACM, 2015.

[82] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[83] Samuel Louvan, Chetan Naik, Sadhana Kumaravel, Heeyoung Kwon, Niranjan Balasubramanian, and Peter Clark. Cross sentence inference for process knowledge. In *EMNLP*, pages 1442–1451, 2016.

[84] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*, 2015.

[85] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[86] Yu-Fei Ma and Hong-Jiang Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 374–381. ACM, 2003.

[87] James R McSkimin and Jack Minker. *The use of a semantic network in a deductive question-answering system*. University of Maryland. Computer Science, 1977.

[88] Yishu Miao, Lei Yu, and Phil Blunsom. Neural Variational Inference for Text Processing. *arXiv*, arXiv:1511.06038, 2015.

[89] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[90] Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[91] Amita Misra, Pranav Anand, Jean E. Fox Tree, and Marilyn Walker. Using summarization to discover argument facets in online idealogical dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 430–440, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[92] Raymond J Mooney and Razvan Bunescu. Mining knowledge from text using information extraction. *ACM SIGKDD explorations newsletter*, 7(1):3–10, 2005.

[93] Alvaro Morales, Varot Premtoon, Cordelia Avery, Sue Felshin, and Boris Katz. Learning to answer questions from wikipedia infoboxes. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1930–1935, Austin, Texas, November 2016. Association for Computational Linguistics.

[94] Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*, volume 4212, pages 318–329. Springer, 2006.

[95] Alessandro Moschitti and Silvia Quarteroni. Linguistic kernels for answer re-ranking in question answering systems. *Information Processing & Management*, 47(6):825–842, 2011.

[96] Alessandro Moschitti and Silvia Quarteroni. Linguistic kernels for answer re-ranking in question answering systems. *Information and Processing Management*, 47(6):825–842, 2011.

[97] Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Annual meeting-association for computational linguistics*, volume 45, page 776, 2007.

[98] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. Semeval-

2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, 2017.

[99] Srini Narayanan and Sanda Harabagiu. Question answering based on semantic structures. In *Proceedings of the 20th international conference on Computational Linguistics*, page 693. Association for Computational Linguistics, 2004.

[100] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[101] Tatsuro Oya and Giuseppe Carenini. Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 133–140, Philadelphia, PA, U.S.A., June 2014. Association for Computational Linguistics.

[102] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational linguistics*, 31(1):71–106, 2005.

[103] Luiz Augusto Pizzato and Diego Mollá. Indexing on Semantic Roles for Question Answering. In *Proceedings of the 2nd workshop on In-*

formation Retrieval for Question Answering, IR4QA'08, pages 74–81, 2008.

[104] Vasin Punyakanok, Dan Roth, and Wen-tau Yih. Mapping dependencies trees: An application to question answering. In *Proceedings of AI&Math 2004*, pages 1–10, 2004.

[105] Rani Qumsiyeh, Maria S Pera, and Yiu-Kai Ng. Generating exact- and ranked partially-matched answers to questions in advertisements. *Proceedings of the VLDB Endowment*, 5(3):217–228, 2011.

[106] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[107] Jinfeng Rao, Hua He, and Jimmy Lin. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1913–1916. ACM, 2016.

[108] Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. Biomedical event extraction using abstract meaning representation. *BioNLP 2017*, pages 126–135, 2017.

[109] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

[110] Deepak Ravichandran, Abraham Ittycheriah, and Salim Roukos. Automatic derivation of surface text patterns for a maximum entropy based question answering system. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 85–87. Association for Computational Linguistics, 2003.

[111] Philip Resnik et al. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. 1999.

[112] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, volume 3, page 4, 2013.

[113] Ellen Riloff and Michael Thelen. A rule-based question answering system for reading comprehension tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems-Volume 6*, pages 13–19. Association for Computational Linguistics, 2000.

[114] Ian Roberts and Robert Gaizauskas. Evaluating passage retrieval approaches for question answering. *Advances in Information Retrieval*, pages 72–84, 2004.

[115] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[116] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49. ACM, 2004.

[117] Sophie Rosset, Olivier Galibert, Gabriel Illouz, and Aurélien Max. Integrating spoken dialog and question answering: the ritel project. In *Ninth International Conference on Spoken Language Processing*, 2006.

[118] Subhro Roy and Dan Roth. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*, 2016.

[119] Subhro Roy and Dan Roth. Unit dependency graph and its application to arithmetic word problem solving. *arXiv preprint arXiv:1612.00969*, 2016.

[120] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[121] Barry Schiffman, Kathleen McKeown, Ralph Grishman, and James Allan. Question Answering Using Integrated Information Retrieval and Information Extraction. In *The Conference of the North American Chapter of the Association for Computational Linguistics*, ACL'07, pages 532–539, 2007.

[122] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[123] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM, 2015.

[124] Dan Shen, Geert-Jan Kruijff, and Dietrich Klakow. Exploring syntactic relation patterns for question answering. *Natural Language Processing–IJCNLP 2005*, pages 507–518, 2005.

[125] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Emnlp-conll*, pages 12–21, 2007.

[126] Dinghan Shen, Martin Renqiang Min, Yitong Li, and Lawrence Carin. Adaptive convolutional filter generation for natural language understanding. *arXiv preprint arXiv:1709.08294*, 2017.

[127] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM, 2017.

[128] Bonggun Shin, Timothy Lee, and Jinho D. Choi. Lexicon integrated cnn models with attention for sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sen-*

timent and Social Media Analysis, pages 149–158, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[129] Robert F. Simmons. Natural language question-answering systems: 1969. *Commun. ACM*, 13(1):15–30, January 1970.

[130] James R. Slagle. Experiments with a deductive question-answering program. *Commun. ACM*, 8(12):792–798, December 1965.

[131] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809, 2011.

[132] Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*, 2016.

[133] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[134] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):422–433, 1979.

[135] Wai Lok Tam, Namgi Han, Juan Ignacio Navarro-Horniacek, and Yusuke Miyao. Finding prototypes of answers for improving answer sentence selection. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4103–4108, 2017.

[136] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Hyperqa: Hyperbolic embeddings for fast and efficient ranking of question answer pairs. *arXiv preprint arXiv:1707.07847*, 2017.

[137] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM, 2003.

[138] Ferhan Ture and Oliver Jojic. No need to pay attention: Simple recurrent neural networks work! In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2856–2862, 2017.

[139] Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *HLT-NAACL*, pages 1268–1278, 2016.

[140] Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736. ACM, 2007.

[141] Ellen M Voorhees. The trec question answering track. *Natural Language Engineering*, 7(04):361–378, 2001.

[142] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

[143] David L Waltz. An english language question answering system for a large relational database. *Communications of the ACM*, 21(7):526–539, 1978.

[144] Bingning Wang, Kang Liu, and Jun Zhao. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1288–1297, Berlin, Germany, August 2016. Association for Computational Linguistics.

[145] Di Wang and Eric Nyberg. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 707–712, Beijing, China, July 2015. Association for Computational Linguistics.

[146] Lu Wang and Claire Cardie. Focused meeting summarization via unsupervised relation extraction. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 304–313, Seoul, South Korea, July 2012. Association for Computational Linguistics.

[147] Mengqiu Wang. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1), 2006.

[148] Mengqiu Wang and Christopher Manning. Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING'10, pages 1164–1172, 2010.

[149] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL'07, pages 22–32, 2007.

[150] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.

[151] Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*, 2016.

[152] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. Sentence Similarity Learning by Lexical Decomposition and Composition. *arXiv*, arXiv:1602.07019, 2016.

[153] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web*, pages 515–526. ACM, 2014.

[154] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[155] Jason D Williams, Nobal B Niraula, Pradeep Dasigi, Aparna Lakshmiratan, Carlos Garcia Jurado Suarez, Mouni Reddy, and Geoff Zweig. Rapidly scaling dialog systems with interactive learning. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 1–13. Springer, 2015.

[156] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

[157] Yuk Wah Wong and Raymond J Mooney. Learning for semantic parsing with statistical machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 439–446. Association for Computational Linguistics, 2006.

[158] Yang Xu and David Reitter. Entropy converges between dialogue participants: Explanations from an information-theoretic perspective. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 537–546, Berlin, Germany, August 2016. Association for Computational Linguistics.

[159] Lingpeng Yang, Donghong Ji, Guodong Zhou, Yu Nie, and Guozheng Xiao. Document re-ranking using cluster validation and label propa-

gation. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 690–697. ACM, 2006.

[160] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 287–296. ACM, 2016.

[161] Yi Yang, Wen-tau Yih, and Christopher Meek. WIKIQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 2013–2018, 2015.

[162] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.

[163] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[164] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, pages 858–867, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[165] Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. Question Answering Using Enhanced Lexical Semantic Models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL'13, pages 1744–1753, 2013.

[166] Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *Proceedings of ACL*, 2014.

[167] Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. Attention-based convolutional neural network for machine comprehension. *arXiv preprint arXiv:1602.04341*, 2016.

[168] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*, 2015.

[169] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *arXiv*, arXiv:1512.05193, 2015.

[170] Koichiro Yoshino, Shinsuke Mori, and Tatsuya Kawahara. Spoken dialogue system based on information extraction using similarity of predicate argument structures. In *Proceedings of the SIGDIAL 2011 Conference*, pages 59–66, Portland, Oregon, June 2011. Association for Computational Linguistics.

[171] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*, 2014.

[172] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 815–824. ACM, 2006.

[173] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32. ACM, 2003.

[174] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491. Association for Computational Linguistics, 2010.

[175] Dong Zhou and Vincent Wade. Latent document re-ranking. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1571–1580. Association for Computational Linguistics, 2009.