SAARLAND UNIVERSITY

MASTER THESIS

Neural Networks for reading comprehension applied to non-factoid question answering

Author:
Yauhen Klimovich

Supervisors:
Prof. Dr. Alessandro Moschitti
Prof. Dr. Günter Neumann

*A thesis submitted in fulfilment of the requirements
for the degree of*

Master of Science in Language Science and Technology

Department of Language Science & Technology

12 March, 2018

Declaration of Authorship

Eidesstattliche Erklärung
Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Declaration
I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged.

Saarbrücken, March 14, 2018, Yauhen Klimovich

Acknowledgements

Without the following people this work couldn't happen:

- My thesis supervisor at University of Trento Prof. Dr. Alessandro Moschitti and PhD students at iKernels group
- My study life supervisor and LCT coordinator at University of Trento Prof. Dr. Raffaella Bernardi and students of The Language, Interaction and Computation Laboratory (CLIC lab)
- Prof. Dr. Gunter Neumann, whose seminars on Question Answering at University of Saarland will be remembered as nice and exciting time
- Prof. Dr. Dietrich Klakow, whose lectures on pattern recognition are of a high-quality, and resourceful conversations at any time on any topic are very helpful
- Prof. Dr. Manfred Pinkal, who gave important suggestions on improvements while working on a research paper
- Dr. Jon Dehdari, who possess the endless inspiration for the field of Computational Linguistics
- Prof. Dr. Viktor V. Krasnoproshin, Prof. Dr. Igor V. Sovpel, Prof. Dr. Vladimir A. Obraztsov, who inspired me to study human language from computational perspective at my years at Belarusian State University
- Ursula Kröner, Dr. Stefan Thater, Dr. ing. Ivana Kruijff-Korbayova, Dr. Jürgen Trouvain, and Bobbye Pernice, whose efforts to make LCT/LST program one of the best in the field are vital.

Additionally, I'd like to show my high appreciation to the LCT-students for their genius ideas often shared with me, and also to LCT committee, which had made my participation in the program possible.

Thank you.

Abstract

Yauhen Klimovich

Neural networks
for reading comprehension applied to
non-factoid question answering

Recent advances on Machine Reading Comprehension of Text have shown a potential of artificial neural network based models for solving the task, but at the same time comparatively high performance is bound with availability of high-quality large-scale datasets. Major part of solutions proposed by the community are complex neural architectures.

An example of state-of-the-art approach on traditional datasets, such as TREC-QA, WikiQA, is Support Vector Machine (SVM) model employing Partial Tree Kernels (PTKs). Though neural and traditional (SVM+PTK) approaches can be compared in the lower-scale setting, it is important to understand the performance of Tree Kernel (TK) methods in a large-scale setting, as TK is a reliable representation of syntactic information, in contrast to neural architectures, which couldn't employ complete syntactic parsing yet.

Though there were some attempts to set baselines with a simpler and more transparent approaches, it seems that there is a gap in the field of having no attempt to apply SVM+TK model in the new large-scale setting.

The thesis gives an overview of novel neural network architectures developed on large-scale datasets and reports the results of our attempts to apply SVM+TK model to large-scale Stanford Question Answering Dataset (SQuAD). We applied SVM+TK, but faced some difficulties mainly due to efficiency problems. We also used different relaxation approaches in order to make TK models applicable and to set the upper bound.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine reading comprehension task (MRC) asking questions about a given text document is a central problem in natural language understanding. It is an important task in computational linguistics as it requires many other Natural Language Processing (NLP) tasks to be solved and at the same time can be a part of building a better search engine or a conversational agent.

Reading comprehension system's goal is to answer any question that could be posed against the facts in a reference text. The task is challenging for machines, as it requires understanding of natural language, knowledge about the world and sometimes complex reasoning. Mainly challenges are due to the requirement to combine facts from different sentences or background knowledge, difficulties in extracting individual facts due to highly compositional semantics, lexical and syntactic variation (Joshi et al., 2017).

The task is not novel, but recently significant progress has been made by introduction of MRC datasets with a volume significantly larger than previously, e.g. Rajpurkar et al. (2016) released SQuAD dataset containing more than 100,000 question-answer pairs while TREC-9 (Voorhees and Tice, 2000) has less than one thousand. A large dataset became an important factor as a high number of data-points allows to train a better machine learning models, especially based on artificial neural networks.

A high level of interest has been shown recently to the task of MRC especially on large-scale datasets. The exact definition of MRC task varies depending on a dataset and evaluation metrics proposed. The details of the definition for MRC and the connection to Question Answering task will be discussed in Chapter 2.

The interest led to a number of neural network based methods applied to datasets like SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017), NewsQA (Trischler et al., 2016b). Though simple sliding window, logistic regression (Rajpurkar et al., 2016) and neural network baselines (Ture and Jojic, 2016; Weissenborn, Wiese, and Seiffe, 2017a; Weissenborn, Wiese, and Seiffe, 2017b) were proposed, it seems that an approach employing Kernel Machines, like Support Vector Machine (SVM), which are able to encode syntactic information, is still missing. The thesis reports the attempt to fill this gap, i.e. to set up a new baseline based on SVM+PTK for SQuAD, in contrast to artificial neural network baseline (Sec. 2.5.2). Our hypothesis of using syntactic information is also supported by observations reported by Xie and Xing (2017), that about 70% answers in SQuAD are exactly constituents (N = 0) and about 97% answers differ from the closest constituents by less or equal to 4 words.

Another motivation for our work is a better understanding of the contribution of syntactic information, as studies on neural approaches report different levels of success while using syntactic features (Sec. 2.5.2). Having a better idea on the role of syntax in a large-scale setting can help to underline both the complexity of the

MRC task tested on SQuAD, and compare strong and weak points of traditional and neural approaches.

The rest of the work is organized as follows: Chapter 2 gives an overview of previous work done in the field, goes into details on the recent advances and an overview of the model we attempt to apply, Chapter 3 illustrates our experiments, Chapter 4 derives the conclusion and names a list of ideas for future research.

Chapter 2

Background

Question Answering (QA) is not only an interesting and challenging application, but also the techniques and methods developed from question answering inspire new ideas in many closely related areas such as document retrieval, time and named-entity expression recognition, etc.

2.1 Question Answering and Reading comprehension

2.1.1 Question Answering

According to Yu et al. (2014), QA can be divided into two categories: an approach which focuses on semantic parsing used to generate a query to a knowledge base, and the other category is open domain question answering, which is more closely related to the field of information retrieval. The task can be seen from more different angles and exact formal definition can vary depending on whether a question requires a fact as an answer or not (factoid/non-factoid QA), whether the topic domain is limited or not (open-domain QA), what kind of and how many sources are used to retrieve the answer (knowledge base, Web; single text document or collection of documents; single or multiple text passages; whether expected answer is a sentence or span of words), by the way of getting the answer (generative or extractive).

A standard task for open-domain Information Retrieval QA is annual Text REtrieval Conference (TREC) competitions (Voorhees and Tice, 2000). Though TREC QA shared task covers different topics, it's usually limited in size. Motivated by a will to test QA systems by wider variety of aspects of language and allow to develop a class of attention based deep neural networks applied to MRC, Hermann et al. (2015) proposed a methodology to build large scale supervised datasets for machine reading. Thus the number of datasets for MRC has shown recently a significant increase, but having differences in task formulation, creating strategies, and evaluation metrics.

Knowledge base style task

There are datasets defined via prediction of textual values from the structured knowledge bases (KB), e.g. WikiReading by Hewlett et al. (2016), an example is given in Table 2.1. KB QA involves translating natural language queries into logical forms which can be executed over a KB.

	Categorization		Extraction	
Doc.	Folkart Towers are twin skyscrapers in the Bayrakli district of the Turkish city of Izmir. Reaching a structural height of 200 m (656 ft) above ground level, they are the tallest . . .	Angeles blancos is a Mexican telenovela produced by Carlos Sotomayor for Televisa in 1990. Jacqueline Andere, Rogelio Guerra and Alfonso Iturralde star as the main . . .	Canada is a country in the northern part of North America. Its ten provinces and three territories extend from the **Atlantic** to the **Pacific** and northward into the **Arctic Ocean**, . . .	Breaking Bad is an American crime drama television series created and produced by Vince Gilligan. The show originally aired on the AMC network for five seasons, from **January 20, 2008**, to . . .
Property	country	original language of work	located next to body of water	start time
Answer	Turkey	Spanish	Atlantic Ocean, Arctic Ocean, Pacific Ocean	20 January 2008

TABLE 2.1: Examples instances from WIKIREADING. The task is to predict the answer given the document and property. Answer tokens that can be extracted are shown in bold, the remaining instances require classification or another form of inference.

Cloze-style task

Formalization of a particular MRC task can vary because of the way a dataset is built. E.g. MRC cloze-style datasets define the task as a word prediction/completion problem (CNN/Daily Mail, Hermann et al. (2015); Children Book Test (CBT) dataset Hill et al. (2015); LAMBADA dataset by Paperno et al. (2016), an example follows).

- **Context**: “Why?" “I would have thought you’d find him rather dry," she said. “I don’t know about that," said Gabriel. “He was a great craftsman," said Heather. “That he was," said Flannery.
- **Target sentence**: “And Polish, to boot," said _____.
- **Target word**: Gabriel

Due to significant interest to the task of MRC boosted by new large-scale datasets evaluated by means of extractive question answering, in this work our main focus is on open-domain non-factoid extractive question answering. The formal definition used in the thesis is given in Chapter 2.3.

2.1.2 Machine Reading Comprehension

The Machine Comprehension of Text (MCT), or Machine Reading comprehension (MRC), has been a central goal of Artificial Intelligence for over fifty years. MRC task goal is testing the ability of a system to understand a document using questions based on upon the content of the document (Joshi et al., 2017). "Machine reading" itself is a loosely-defined notion, ranging from extracting selective facts to constructing complex, inference-supporting representations of text.

Trying to define machine comprehension in terms of Question Answering in a general way, Burges (2013) proposed the following definition: "A machine comprehends a passage of text if, for any question regarding that text that can be answered correctly by a majority of native speakers, that machine can provide a string which those speakers would agree both answers that question, and does not contain information irrelevant to that question."

2.2 Factoid and Non-factoid Question Answering

In the real-world QA scenario, people may ask questions about both entities (factoid) and non-entities such as explanations and reasons (non-factoid), examples of both types are given in the Table 2.3, Yu et al. (2016).

Factoid QA

The first type of questions that research focused on was factoid questions. For example, "When was X born?", "In what year did Y take place?", or definitional questions (biographical questions such as "Who is Hilary Clinton?", and entity definition questions such as "What is DNA?"), list questions (e.g. "List the countries that have won the World Cup"), scenario-based QA (given a short description of a scenario, answer questions about relations between entities mentioned in the scenario) and 'why'-type questions. Starting in 1999, an annual evaluation track of question answering systems has been held at the Text REtrieval Conference (TREC)(Voorhees and Tice, 2000), TREC summary over the number of question is in the Table 2.2. Following the success of TREC, in 2002 both CLEF and NTCIR workshops started to organize multilingual and cross-lingual QA tracks (Wang, 2006).

TABLE 2.2: Summary of the number of question in TREC over years

Evaluation	# of Qs
TREC 8 (1999)	198
TREC 9 (2000)	692
TREC 10 (2001)	491
TREC 11 (2002)	499
TREC 12 (2003)	413
TREC 13 (2004)	231

According to the survey by Wang (2006), typical factoid QA system employs a pipeline architecture that consist of three main modules:

- *Question analysis* module processes the question, including the extraction of a question type and runs syntactic and semantic analysis, dependency parsing. Output of the module is a set of keywords for further retrieval.

- *Document and passage retrieval* takes the keywords from *Question analysis* and uses a search engine to retrieve relevant document or passage.

- *Answer extraction* module analyzes the retrieved documents/passage, produces a list of answer candidates and rank them according to a scoring function.

Mostly models for QA are different in features used for matching question and answer sentence in answer extraction module. Early TREC systems focused on hand-crafting or automatically acquiring surface text patterns, which became a reason of low recall at the evaluation. Pattern-based and rule-based approaches also led to a problem of inability to capture long-distance dependencies (Ravichandran and Hovy, 2002; Riloff and Thelen, 2000). Motivated by the assumption, that factoid

TABLE 2.3: Example of questions (with answers) which can be potentially answered with RC on a Wikipedia passage. The first question is factoid, asking for an entity. The second and third are non-factoid.

The United Kingdom (UK) intends to withdraw from the European Union (EU), a process commonly known as Brexit, as a result of a June 2016 referendum in which 51.9% voted to leave the EU. The separation process is complex, causing political and economic changes for the UK and other countries. As of September 2016, neither the timetable nor the terms for withdrawal have been established: in the meantime, the UK remains a full member of the European Union. The term "Brexit" is a portmanteau of the words "British" and "exit".
Q1. Which country withdrew from EU in 2016? A1. United Kingdom
Q2. How did UK decide to leave the European Union? A2. as a result of a June 2016 referendum in which 51.9% voted to leave the EU
Q3. What has not been finalized for Brexit as of September 2016? A3. neither the timetable nor the terms for withdrawal

questions can be easily classified into distinct classes (date, person, location, number, etc.), named entities became another feature for answer extraction (Shih et al., 2005) and had shown some little improvement.

Later, Shen, Kruijff, and Klakow (2005) had shown that incorporation of a tree kernel function (Collins and Duffy, 2002) to compute the similarity between two dependency trees (question and answer sentence) improves Mean Reciprocal Rank (MRR) by 6.91 %. An example of the dependency tree is given in the Figure A.2. In the succeeding work Shen and Klakow (2006) also used dependency tree partial matching feature as input into maximum entropy classifier to produce the final scoring; and experimental results showed that the method significantly outperformed state-of-the-art syntactic relation-based methods by up to 20% in MRR.

Wang, Smith, and Mitamura (2007) proposed a method, which models relations between question and answer candidate, based on the assumption, that questions and their (correct) answers relate to each other via loose but predictable syntactic transformations. The method is based on probabilistic quasi-syntactic grammar. Other models like proposed by Heilman and Smith (2010), Wang and Manning (2010), and Yao et al. (2013) had focused on methods to improve the accuracy of Tree Edit Distance (TED).

Severyn and Moschitti (2013) showed that tree kernels can be applied to learn structural patterns for both answer sentence selection and answer extraction employing automatic way of learning the features. The approach yielded the improvement of up to 22% over previous state-of-the-art in F1 measure on TREC-QA dataset. An example of question answer trees pair is given in Figure 2.6.

Non-factoid QA

Non-factoid QA can be intuitively defined as those cases, when a question asks for explanations and descriptions as opposed to named entities and facts. Compared to the relatively easier QA task of predicting single tokens/entities, predicting answers of arbitrary lengths and positions significantly increases the search space: the number of possible candidates to consider is in the order of O(n^2), where n is the length of a sentence in words. However, for previous works, in which answers are single tokens/entities or from candidate lists, the complexity is O(n) or the size of candidates list l (usually $1 \leq 5$), respectively (Yu et al., 2016). An example of the non-factoid question answering task can be community Question Answering challenge (cQA) (Nakov et al., 2016), where for a given unseen question a system should predict the best answer from multiple answers given to other (potentially similar) questions from an Internet forum. For cQA Tymoshenko, Bonadiman, and Moschitti (2016) proposed a successful kernel based classifier and used a kernel method to rank related answers.

2.3 Extractive question answering

According to Lee et al. (2016), *extractive question answering* systems take as input a question $\mathbf{q} = \{q_0, \dots, q_n\}$ and a passage of text $\mathbf{p} = \{p_0, \dots, p_m\}$ from which they predict a single answer span $\mathbf{a} = \langle a_{start}, a_{end} \rangle$, represented as a pair of indices into $\mathbf{p}$. Machine learned extractive question answering systems learns a predictor function $f(\mathbf{q}, \mathbf{p}) \rightarrow \mathbf{a}$ from a training dataset of $\langle \mathbf{q}, \mathbf{p}, \mathbf{a} \rangle$ triples.

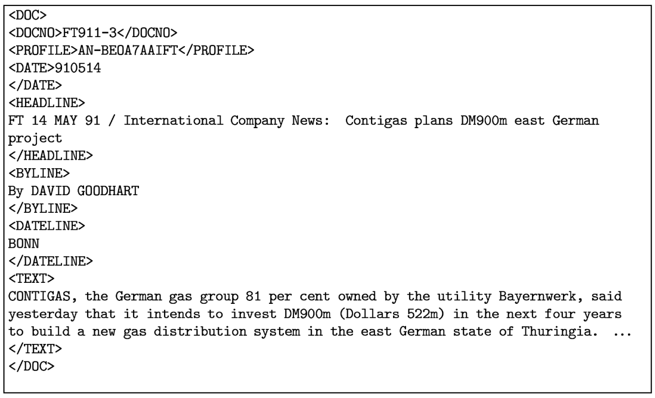
Recent large-scale MRC dataset examples which formalize the task as extractive question answering are SQuAD (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2016b). Overview in more details of the aforementioned and other modern datasets is given in Sec. 2.4.

2.4 Datasets

2.4.1 TREC-QA

Series of TREC-QA challenges are one the first in question answering. The document collection used in the TREC-9 QA task was the set of newspaper/newswire documents. The source for the answer is a list of documents. Example of question-answer pair from TREC-9 is (*"How many hexagons are on a soccer ball?", "20"*). A QA system should detect answer from a set of 1000 documents, an illustrative example of a document is given in the Figure 2.1.

FIGURE 2.1: A document extract from Financial Times (Voorhees and Tice, 2000)



```
<DOC>
<DOCNO>FT911-3</DOCNO>
<PROFILE>AN-BEOA7AAIFT</PROFILE>
<DATE>910514
</DATE>
<HEADLINE>
FT 14 MAY 91 / International Company News:  Contigas plans DM900m east German
project
</HEADLINE>
<BYLINE>
By DAVID GOODHART
</BYLINE>
<DATELINE>
BONN
</DATELINE>
<TEXT>
CONTIGAS, the German gas group 81 per cent owned by the utility Bayernwerk, said
yesterday that it intends to invest DM900m (Dollars 522m) in the next four years
to build a new gas distribution system in the east German state of Thuringia.  ...
</TEXT>
</DOC>
```

2.4.2 Large-scale MRC datasets tested by QA

MRC also can be defined as extractive open-domain question answering task. Related large-scale datasets are MCTest with multi-choice answers (Richardson, 2013), SQuAD (Rajpurkar et al., 2016), RACE (Lai et al., 2017), NewsQA (Trischler et al., 2016b), NarrativeQA (Kočiský et al., 2017), MS MARCO proposed by Nguyen et

	Train	Dev	Test	Total
WikiHop	43,738	5,129	2,451	51,318
MedHop	1,620	342	546	2,508

TABLE 2.4: WikiHop and MedHop datasets volumes.

al. (2016), TriviaQA (Joshi et al., 2017), SearchQA - by Dunn et al. (2017), WikiHop (Welbl, Stenetorp, and Riedel, 2017), Tables A.1, 2.4.

Below we give a short overview of other both low- and large-scale datasets for MRC and underline their differences as well as some critics. All datasets questions, answers, passages are in English.

2.4.3 SQuAD

SQuAD (Rajpurkar et al., 2016) is in focus of this work as the dataset was first in the list of large-scale datasets developed for the task of extractive question answering (span selection), defined in Sec. 2.3. An example of Paragraph-Question-Answer triple is given Figure 2.2.

The development of SQuAD was motivated by a will to solve the issue which was inevitable for earlier developed datasets either being small and in high quality, like TREC-QA, or large but semi-synthetic, like bAbI (Weston et al., 2015).

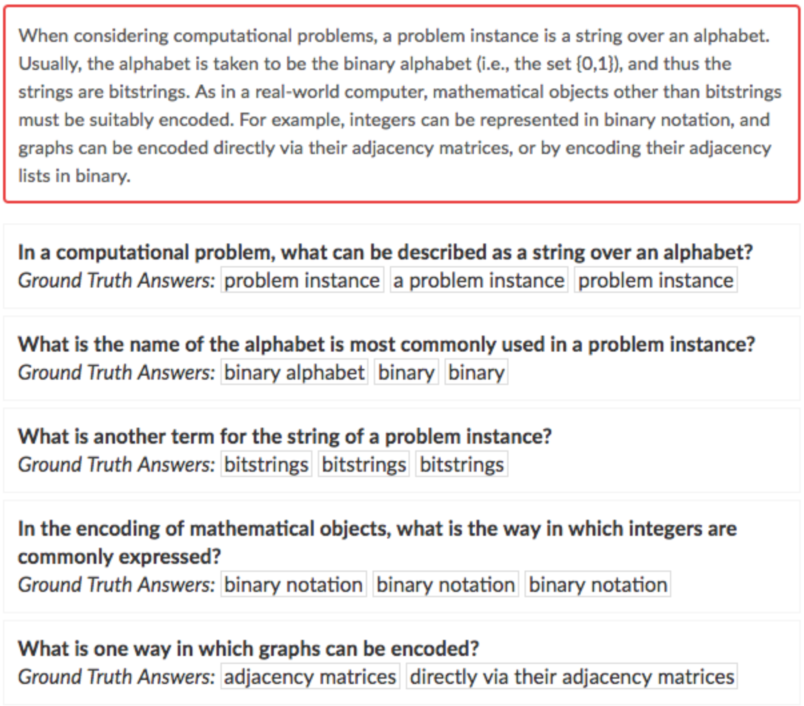
To create the dataset at first 536 articles were randomly selected out of top 10000 articles in English Wikipedia. After cleaning up selected articles, individual paragraphs were extracted, so that they are not shorter than 500 characters. In total there were 23,215 paragraphs covering wide range of topics. The dataset was split into train, dev, test samples by proportion 8/1/1. Then each paragraph was assigned up to 5 question-answer pairs by crowd-workers. To control post-collection quality crowd-worker was asked to select the shortest span in a paragraph given a question.

Evaluation employs two measures - exact match (EM) and F1 score ($F_1 = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$). Both metrics ignore punctuation and English articles (*the,a,an*). EM measures the percentage of predictions that match any one of the ground truth answers exactly (ignoring punctuations and articles), F1 is set in the macro-average way: considering a prediction and ground-truth as bag-of-words, F1 is measured as following: F1 score is measured over all pairs of prediction and ground-truth per answer, then the maximum value is chosen among those, then this max value contributes to get a final score - average over F1 over all of the questions. These standard accuracy metrics got criticized because of their not clear ability to measure real Natural Language Understanding (NLU) of methods developed for SQuAD. Jia and Liang (2017) proposed two additional metrics: AddAny and AddSent, an example is shown in Figure 2.3. The idea is to add distracting sentences to a passage to fool a system, thus testing the ability to understand the passage more accurately. Two popular models for SQuAD (BiDAF, Match-LSTM) tested with these additional metrics have shown dramatic decrease in accuracy (around 2 times lower scores), though Human performance decreased only by around 3%.

2.4.4 MCTest

MCTest (Richardson, 2013) consists of 660 elementary-level children's stories with associated questions and answers, collected with the help of crowd-sourcing. Each question is linked with a set of 4 candidate answers, ranging from a single word

FIGURE 2.2: Example of paragraph - question - answer from SQuAD train dataset



When considering computational problems, a problem instance is a string over an alphabet. Usually, the alphabet is taken to be the binary alphabet (i.e., the set {0,1}), and thus the strings are bitstrings. As in a real-world computer, mathematical objects other than bitstrings must be suitably encoded. For example, integers can be represented in binary notation, and graphs can be encoded directly via their adjacency matrices, or by encoding their adjacency lists in binary.

In a computational problem, what can be described as a string over an alphabet?
Ground Truth Answers: problem instance | a problem instance | problem instance

What is the name of the alphabet is most commonly used in a problem instance?
Ground Truth Answers: binary alphabet | binary | binary

What is another term for the string of a problem instance?
Ground Truth Answers: bitstrings | bitstrings | bitstrings

In the encoding of mathematical objects, what is the way in which integers are commonly expressed?
Ground Truth Answers: binary notation | binary notation | binary notation

What is one way in which graphs can be encoded?
Ground Truth Answers: adjacency matrices | directly via their adjacency matrices

to a full sentence. The questions are designed to require rudimentary reasoning and synthesis of information across sentences, making the dataset quite challenging. Though the dataset size can limit the training of expressive statistical models, recent comprehension models have performed well on *MCTest*, including a highly structured neural model (Trischler et al., 2016a). Also the model by Wang et al. (2015) employs syntax, frame and semantic features, which emphasize usefulness of structured linguistic data on relatively small datasets. The evaluation metric is an accuracy (number of correctly predicted answers).

2.4.5 CNN/Daily Mail

The *CNN/Daily Mail* corpus (Hermann et al., 2015) is a set of news articles scraped from those CNN/Daily Mail newspapers with corresponding cloze-style questions, when a detection of the correct answer relies mostly on recognizing textual entailment between the article and the question. The named entities within an article are identified and anonymized in a preprocessing step and constitute the set of candidate answers, Table 2.5. It's relatively easy to collect the dataset as the process can be semi-automatized. Chen, Bolton, and Manning (2016) showed that the task requires only limited reasoning.

Children's Book Test

The *Children's Book Test* (*CBT*), Hill et al. (2015), is similar to that of *CNN/Daily Mail* by the collection method, but differs by the source, it consists of passages from children's books available through Project Gutenberg; last sentence of the passage is a cloze-style question.

FIGURE 2.3: An example from the SQuAD dataset. The BiDAF Ensemble model originally gets the answer correct, but is fooled by the addition of an adversarial distracting sentence (in blue)

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

2.4.6 BookTest

Bajgar, Kadlec, and Kleindienst (2016) presented *BookTest*. This is an extension to the named-entity and common-noun strata of *CBT* that increases their size by over 60 times. Bajgar, Kadlec, and Kleindienst (2016) showed that training on the augmented dataset yields a model (Bajgar, Kadlec, and Kleindienst, 2016) that matches human performance on *CBT*.

2.4.7 NewsQA

NewsQA (Trischler et al., 2016b) is similar to SQuAD by size (over 100,000 human-generated question-answer pairs) and methodology (naturally collected by crowd-sourcing), but differs by the source (news articles from CNN/Daily Mail) and the building process designed to encourage exploratory, curiosity-based questions that 'reflect' human information seeking. This modification in collection strategy made NewsQA more challenging as more questions (comparing to SQuAD) require more difficult forms of reasoning: synthesis and inference are almost doubled the number of similar types in SQuAD.

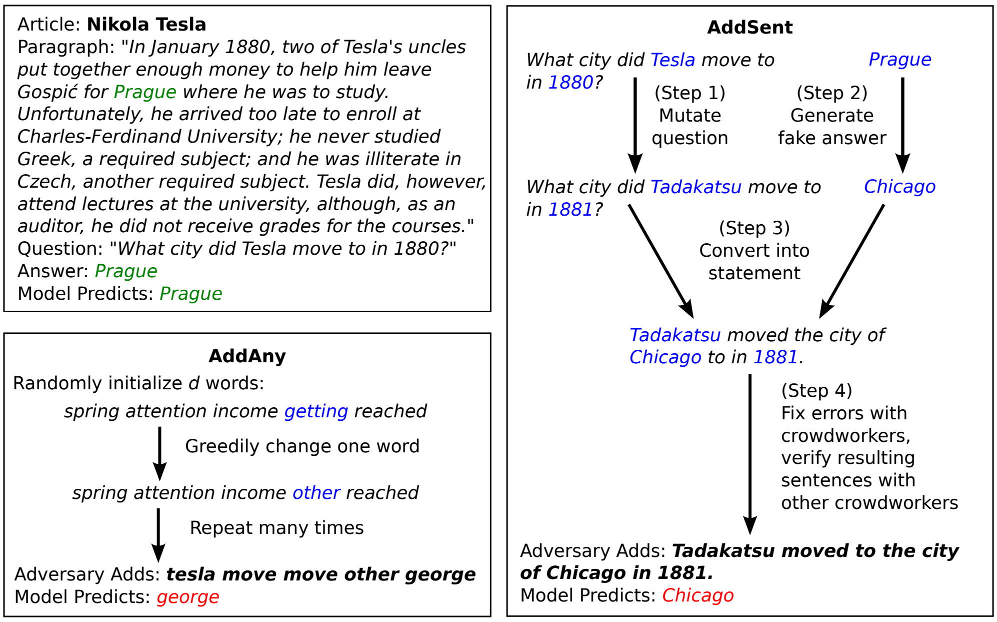
2.4.8 MS MARCO

MicroSoft MAchine Reading COmprehension (MS MARCO) (Nguyen et al., 2016) is a recent large-scale (100,000 queries with corresponding answers in first version) MRC dataset which sets generative question answering task and collected from real-user queries of a web-search engine.

2.4.9 TriviaQA

Joshi et al. (2017) collected TriviaQA, the dataset consisting of over 650,000 question-answer evidence triples and questions are authored organically by enthusiasts (independently from the task), another advantage is that evidence documents are gathered from different sources(Web and Wikipedia). The motivation was to provide a dataset with more challenging questions, with substantial syntactic and lexical variability, often requiring multi-sentence reasoning (volume is three times larger than in SQuAD). Examples of analysis of reasoning are presented in the Table A.1.

FIGURE 2.4: An illustration of the ADDSENT and ADDANY adversaries.

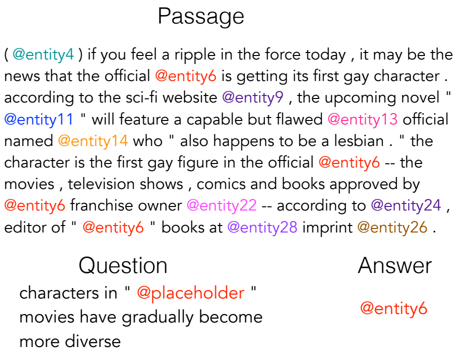


Article: **Nikola Tesla**
Paragraph: *"In January 1880, two of Tesla's uncles put together enough money to help him leave Gospić for Prague where he was to study. Unfortunately, he arrived too late to enroll at Charles-Ferdinand University; he never studied Greek, a required subject; and he was illiterate in Czech, another required subject. Tesla did, however, attend lectures at the university, although, as an auditor, he did not receive grades for the courses."*
Question: *"What city did Tesla move to in 1880?"*
Answer: *Prague*
Model Predicts: *Prague*

AddAny
Randomly initialize *d* words:
spring attention income getting reached
Greedily change one word
spring attention income other reached
Repeat many times
Adversary Adds: ***tesla move move other george***
Model Predicts: *george*

AddSent
What city did Tesla move to in 1880?
(Step 1) Mutate question
Prague
(Step 2) Generate fake answer
What city did Tadakatsu move to in 1881?
Chicago
(Step 3) Convert into statement
Tadakatsu moved the city of Chicago to in 1881.
(Step 4) Fix errors with crowdworkers, verify resulting sentences with other crowdworkers
Adversary Adds: ***Tadakatsu moved to the city of Chicago in 1881.***
Model Predicts: *Chicago*

FIGURE 2.5: An example item from CNN dataset



Passage

(@entity4) if you feel a ripple in the force today , it may be the news that the official @entity6 is getting its first gay character . according to the sci-fi website @entity9 , the upcoming novel " @entity11 " will feature a capable but flawed @entity13 official named @entity14 who " also happens to be a lesbian . " the character is the first gay figure in the official @entity6 -- the movies , television shows , comics and books approved by @entity6 franchise owner @entity22 -- according to @entity24 , editor of " @entity6 " books at @entity28 imprint @entity26 .

Question

characters in " @placeholder " movies have gradually become more diverse

Answer

@entity6

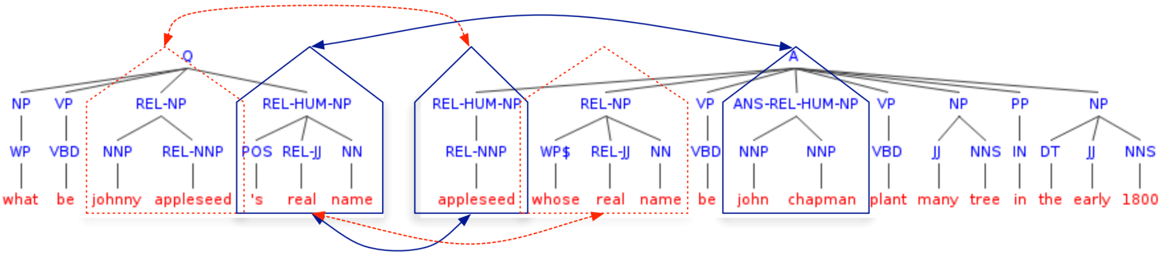
2.5 Models

2.5.1 Support Vector Machines(SVMs)

SVMs (Cortes and Vapnik, 1995) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Support vector machines attempt to pass a linearly separable hyperplane through a dataset in order to classify the data into two groups. This hyperplane is a linear separator for any dimension, what makes SVM effective, as to separate data-points which are not linearly separable in a given number of dimension, adding 'artificial' dimension makes it possible.

Main advantage of SVM is effectiveness and easy to encode structural data as a feature thanks to a Kernel Trick, which requires only a definition of a similarity function between points of data, but the same time computational time grows exponentially.

FIGURE 2.6: Shallow tree representation of the example q/a pair. Dashed arrows (red) indicate the tree fragments (red dashed boxes) in the question and its answer sentence linked by the relational REL tag, which is established via syntactic match on the word lemmas. Solid arrows (blue) connect a question focus word name with the related named entities of type Person corresponding to the question category (HUM) via a relational tag REL-HUM. Additional ANS tag is used to mark chunks containing candidate answer (here the correct answer John Chapman), (Severyn and Moschitti, 2013)



Encoding syntactic trees for QA in SVM

Previous work discussed in Sec. 2.2 used a rich number of distributional semantic, knowledge-based, translation and paraphrase resources to build explicit feature vector representations. One evident potential downside of using feature vectors is that a great deal of structural information encoded in a given text pair is lost (Severyn and Moschitti, 2013). A more versatile approach in terms of the input representation relies on kernels and measures the similarity between question and answer pairs. Question/answer pair is defined as a triple consisting of a question tree T_q and answer sentence tree T_s and a similarity feature vector v. Kernel is defined then as in Figure 2.7, where K_{tk} computes a structural kernel, e.g., tree kernel, and K_v is a kernel over feature vectors, e.g., linear, polynomial, gaussian, etc. Structural kernels can capture the structural representation of a question/answer pair whereas traditional feature vectors can encode some sort of similarity (lexical, syntactic, semantic, between a question and its candidate answer) (Severyn and Moschitti, 2013).

$$\begin{aligned} K(\boldsymbol{x}^i, \boldsymbol{x}^j) &= K_{\text{TK}}(\boldsymbol{T}_q^i, \boldsymbol{T}_q^j) \\ &+ K_{\text{TK}}(\boldsymbol{T}_s^i, \boldsymbol{T}_s^j) \\ &+ K_{\text{v}}(\boldsymbol{v}^i, \boldsymbol{v}^j), \end{aligned}$$

FIGURE 2.7: Kernel used in (Severyn and Moschitti, 2013)

Partial Tree Kernels

Partial Tree Kernel (PTK) (Moschitti, 2006) are used to compute K_{tk}. PTK can be effectively applied to both constituency and dependency parse trees, (Manning and Schütze, 1999). It generalizes the syntactic tree kernel (STK) (Collins and Duffy, 2002), which maps a tree into the space of all possible tree fragments constrained by the rule that sibling nodes cannot be separated. In contrast, the PTK fragment can contain any subset of siblings, i.e., PTK allows for breaking the production rules in syntactic trees. Consequently, PTK generates an extremely rich feature space, which results in higher generalization ability (Severyn and Moschitti, 2013).

2.5.2 Artificial Neural Networks

Many artificial intelligence tasks can be solved by designing the right set of features to extract for that task, these features are then used by a simple machine learning algorithm. Though, usually it is difficult to know what features should be extracted. This can be solved by using machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. A major source of difficulty in many real-world artificial intelligence applications is that many of the factors of variation influence every single piece of data we are able to observe, which leads to difficulties of extracting such high-level, abstract features from raw data. *Deep learning* solves this central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts (Goodfellow, Bengio, and Courville, 2016). There is no formal definition of the term *Deep learning* as through the history of the field it was called differently, e.g. cybernetics, parallel distributed processing. Nowadays, usually the term is used to call machine learning methods based on multi-layer neural networks. The place of deep learning methods in Artificial Intelligence field is shown in Figure 2.8.

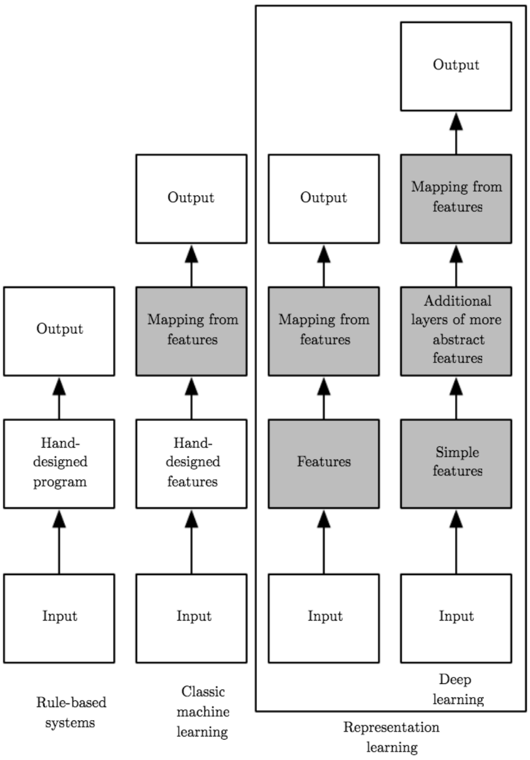


FIGURE 2.8: How different parts of an AI system relate to each other within AI disciplines, (Goodfellow, Bengio, and Courville, 2016)

Taking the advantage of rapid development of deep learning, and large-scale MRC benchmark datasets discussed in Sec. 2.4, end-to-end neural networks have achieved promising results (Hu, Peng, and Qiu, 2017). In the following section we will study recent neural architectures applied for SQuAD.

Neural network architectures for SQuAD

This section will give an overview for state-of-the-art neural architectures for SQuAD dataset. The leader-board of models for the dataset is available on the Web [1]. At the moment of publication of the thesis top model Hybrid AoA Reader (ensemble), by Joint Laboratory of HIT and iFLYTEK Research, surpassed Human upper baseline in Exact Match metric (82.482 against 82.304), being still lower in terms of F1 score (89.281 against 91.221). Another top model, Reinforced Mnemonic Reader (Hu, Peng, and Qiu, 2017) + A2D (ensemble), by Microsoft Research Asia & NUDT, showed higher result on EM than Human Performance, but not F1 [2].

A common approach to build a neural network architecture of recent works is based on 'encoder-interaction-pointer' framework (Hu, Peng, and Qiu, 2017).

Encoder part is responsible for projection of both query(question) and context (answer, paragraph) into distributional space and encoding by a neural network, usually Recurrent Neural Network(RNN).Complex interaction between a query (question) and context (answer) is done by attention mechanism (Bahdanau, Cho, and Bengio, 2014), in extractive QA Pointer Network (Vinyals, Fortunato, and Jaitly, 2015) is used to select the boundaries of an answer.

FIGURE 2.9: A conceptual architecture illustrating recent advances in MRC, Huang et al. (2017). Lowest level represents input vectors, Rectangular boxes represent RNNs and the numbered arrows is an attention mechanism

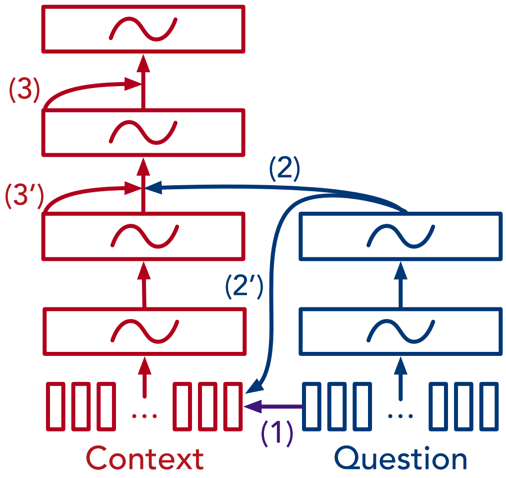


Figure 2.9 shows a schema of a common architecture of majority of the neural network models for MRC.

In the following subsections we will give an overview of selective set of neural architectures for SQuAD. Evaluation results for the reviewed models are in the Table 2.5.

End-to-end answer extraction and Ranking for Reading Comprehension

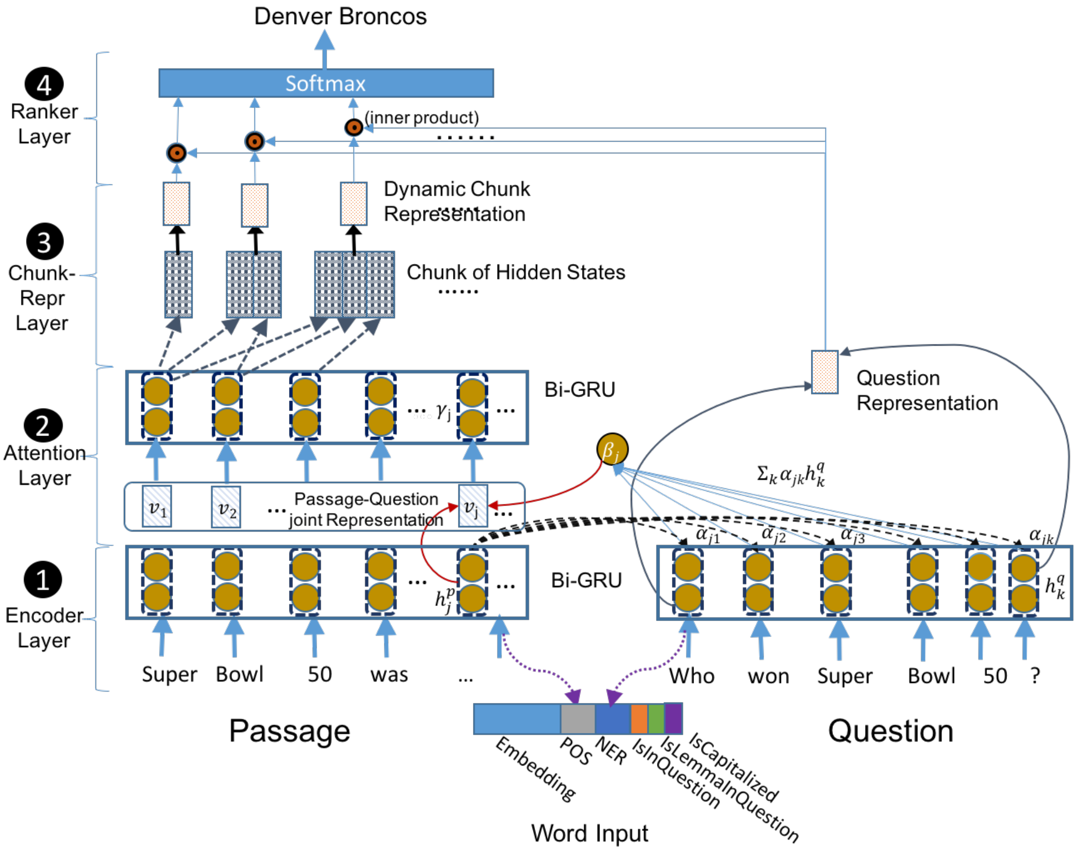
Yu et al. (2016) proposed dynamic chunk reader (DCR), which was the first model that generated and ranks the answer spans, which made the model different from original logistic regression baseline (Rajpurkar et al., 2016), and represents answer

[1] http://stanford-qa.com

[2] The models usually reported in two versions (single and ensemble). Ensemble score is an average results of top-N prediction of multiple runs of training of the same model architecture but initialized with different hyper-parameters.

candidates as chunks instead or previous word-level representations. Another important contribution is a new baseline to understand the gap between cloze-style models applied to span-extraction type models. POS pattern trie tree of answer subsequences was build on the training set and the experiments had shown that for > 90 % of the questions on the development set, the ground truth answer is included in the candidate list generated by the POS pattern trie tree. The overview of the neural architecture is given in the Figure 2.10

FIGURE 2.10: Dynamic chunk reader neural architecture



Dynamic coattention networks for question answering

Xiong, Zhong, and Socher (2016) proposed Dynamic coattention networks (DCN), which is different from previous models by introduction of Coattention encoder. The motivation was as to overcome the issue of not being able to recover from local maxima corresponding to answer span, due the single-pass nature of previously proposed neural network models. DCN tries to solve this problem and makes it able to estimate the start and the end positions of the span multiple times. Figure 2.11, 2.12 gives a schematic overview of the model architecture.

Bi-Directional attention flow for machine comprehension (BiDAF)

Seo et al. (2016) proposed multi-stage hierarchical process that represents the context at different levels of granularity (character, word, context) and bi-directional attention flow. The novelty of the model is also in the way of reduction of the information loss usually faced due to early summarization: BiDAF's attention layer vector is

FIGURE 2.11: Overview of dynamic coattention network

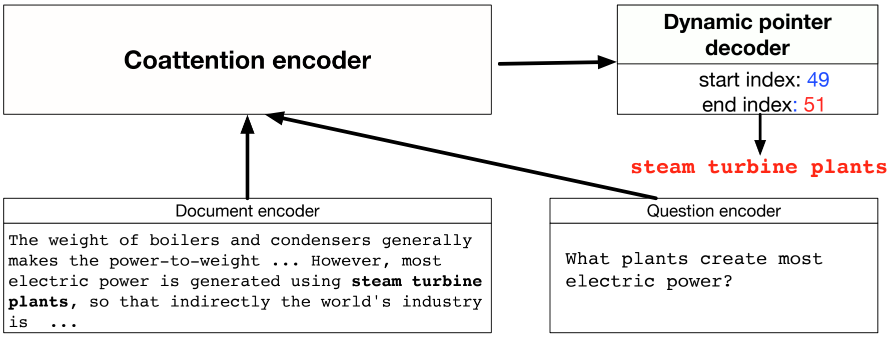
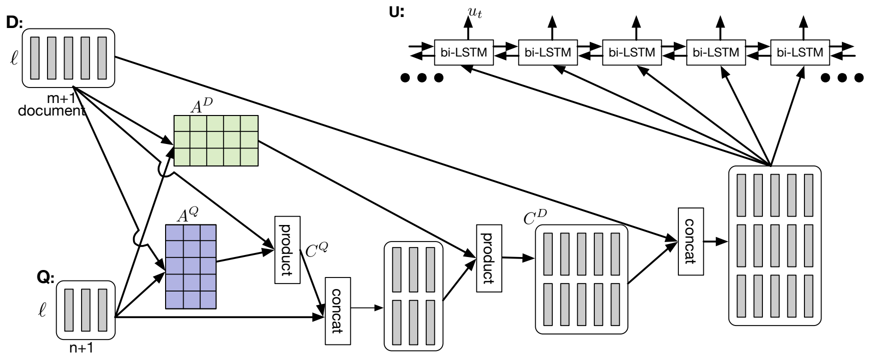

Coattention encoder
Dynamic pointer decoder
start index: 49
end index: 51
steam turbine plants
Document encoder
The weight of boilers and condensers generally makes the power-to-weight ... However, most electric power is generated using **steam turbine plants**, so that indirectly the world's industry is ...
Question encoder
What plants create most electric power?

FIGURE 2.12: Coattention architecture



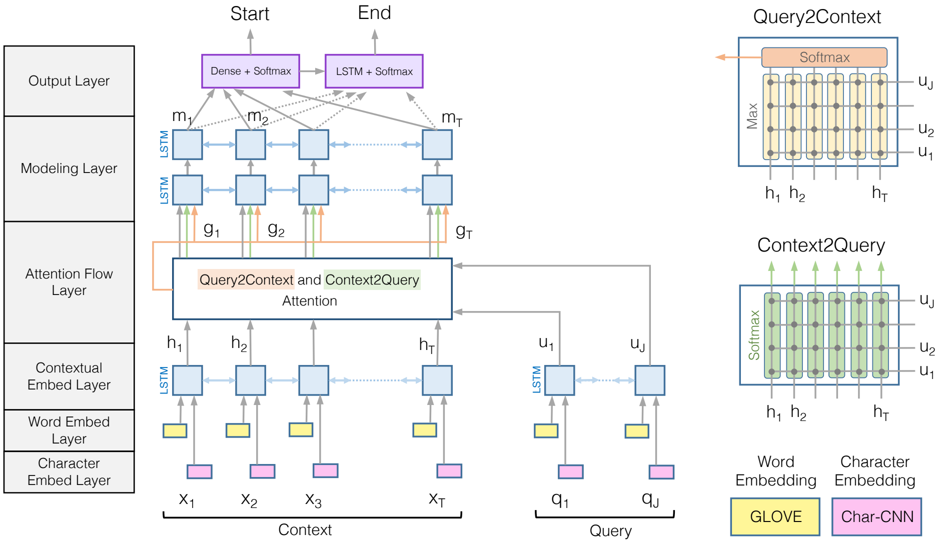
Model	Dev EM	Dev F1	Test EM	Test F1
BiDAF (Seo et al., 2016)	67.7	77.3	73.3	81.1
DCN (Xiong, Zhong, and Socher, 2016)	70.3	79.4	71.2	80.4
Dynamic Chunk Reader (Yu et al., 2016)	62.5	71.2	62.5	71.0
FastQA (Weissenborn, Wiese, and Seiffe, 2017a)	-	-	78.9	70.8
SECT-LSTM (Liu et al., 2017)	-	-	68.12	77.21
SEDT-LSTM (Liu et al., 2017)	-	-	68.48	77.97
Reinforced M-Reader (Hu, Peng, and Qiu, 2017)	-	-	77.7	84.9
MEMEN (Pan et al., 2017)	-	-	78.23	85.34
R-NET (Wang et al., 2017)	-	-	82.65	88.49
Baseline (Rajpurkar et al., 2016)	40.0	51.0	40.4	51.0
Human (Rajpurkar et al., 2016)	81.4	91.0	82.3	91.2

TABLE 2.5: Comparative results on performance at the time of publishing of DCN to original baseline and Yu et al. (2016)

computed at every step along with the representations from previous layers and is allowed to *flow* through subsequent modeling layer. The neural architecture is depicted in the Figure 2.14

Table 2.6 gives an overview of ablation study results on BiDAF model by the authors, and emphasizes the importance of word embeddings and C2Q (context to query) attention which signifies which query works are more relevant to each of context word.

FIGURE 2.13: BiDAF neural architecture



	EM	F1
No char embedding	65.0	75.4
No word embedding	55.5	66.8
No C2Q attention	57.2	67.7
No Q2C attention	63.6	73.7
Dynamic attention	63.5	73.6
BiDAF (single)	67.7	77.3
BiDAF (ensemble)	72.6	80.7

TABLE 2.6: BiDAF ablation study on SQuAD dev set

Figure 2.14 also underlines the lowest performance on 'why'-type questions (non-factoid), which are in the focus of the thesis.

FastQA: A simple and efficient Neural architecture for Question Answering

Weissenborn, Wiese, and Seiffe (2017a) pointed out on a missed neural network baseline. As significant number of previous models were built in *top-down* way, this led to complex architectures and the analysis was mostly done by ablation studies. Motivated by the assumption, that ablation study is not a fully sufficient approach to understand the impact of each of many parts of a complex neural network architecture, the authors proposed FastQA, a neural network based model built as a result of an incremental extension process (*bottom-up*), starting from very basic model that is enhanced by extensions only by necessity The initial core heuristic feature is based on the following axioms:

- the type of answer span should correspond to the answer type given by question
- the correct answer should further be surrounded by a context that fits the question

Having only three layers in the NN architecture (embedding, encoding, and answer layer), FastQA is different comparing to other systems which employ complex attention layer, Figure 2.15 The quantitative analysis had shown, that a simple binary

FIGURE 2.14: Correctly answered questions by BiDAF broken down by the 10 most frequent first words in the question, from Seo et al. (2016)

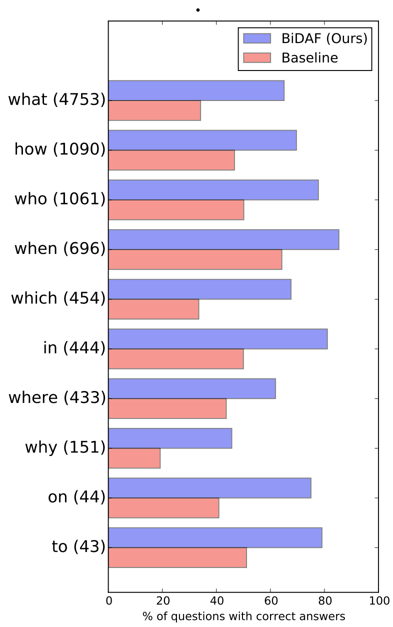
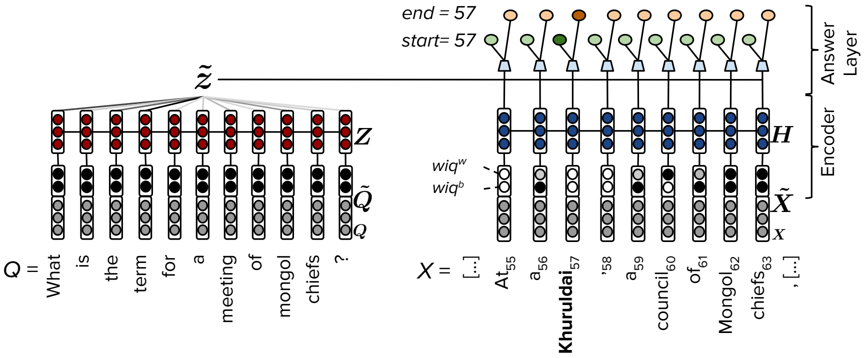


FIGURE 2.15: FastQA neural architecture



word-in-question feature plays an important role in evaluation (according to ablation studies, the drop is 13.6 in F1 and EM), that character based embeddings have a notable effect. The qualitative analysis results underline the importance of syntactic understanding, semantic distinction between lexemes, co-reference resolution and context sensitive binding. Taking into account the error analysis, authors claim that an extractive QA system does not have to solve the reasoning types that were used to classify SQuAD instances to be successful.

The model's errors analysis in capturing the syntactic structures in order to extract the correct answer span are important for the motivation of the thesis.

Structural embeddings of Syntactic trees for Machine Comprehension

Liu et al. (2017) proposed the model different to other by extending the encoder with a structural embeddings of syntactic trees: constituency(SECT) and dependency (SEDT). Attention layer is based on BiDAF neural architecture (Seo et al., 2016). Each word in SECT is encoded by a path from the root of the tree, and in SEDT each word

Method	**EM**	**F1**
SECT-Random	5.64	12.85
SECT-Random-Order	30.04	39.98
SECT-Only	34.21	44.53
SEDT-Random	0.92	8.82
SEDT-Random-Order	31.82	43.65
SEDT-Only	32.96	44.37

TABLE 2.7: Performance comparisons of models with only syntactic information against their counterparts with randomly shuffled node sequences and randomly generated tree nodes using the SQuAD Dev set

is represented by all dependent children, according to a dependency link. Examples are given in the Figure 2.18

FIGURE 2.16: Partial dependency parse tree of an example context "The Annual Conference, roughly the equivalent of a diocese in the Anglican Communion and the Roman Catholic Church or a synod in some Lutheran denominations such as the Evangelical Lutheran Church in America, is the basic unit of organization within the UMC."

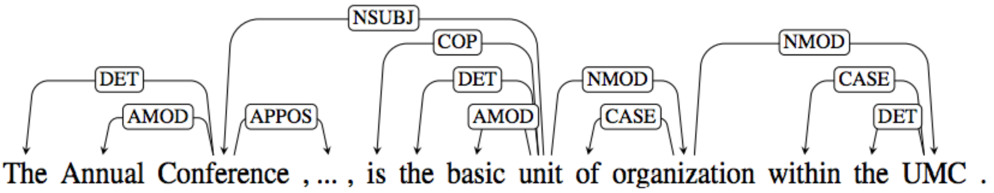
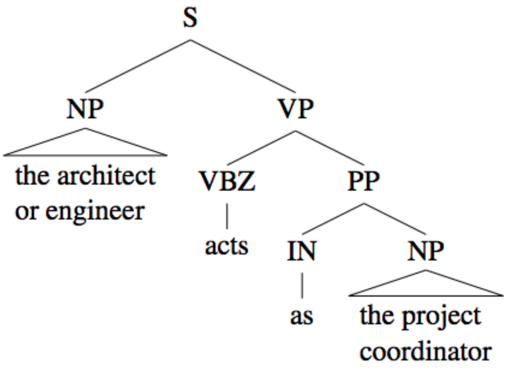
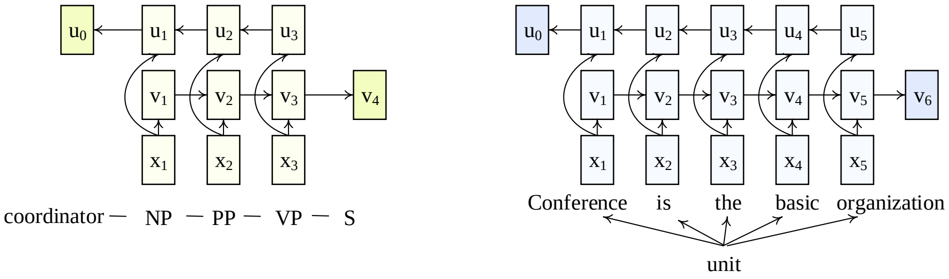

NSUBJ
COP
DET
NMOD
DET
AMOD
APPOS
DET
AMOD
NMOD
CASE
CASE
DET
The Annual Conference , ... , is the basic unit of organization within the UMC .

FIGURE 2.17: Constituency tree for the example "the architect or engineer acts as the project coordinator"



The role of structural embeddings was evaluated by the ablation studies, result is given in the table 2.7

The results can imply that both ordering and the content of the syntactic tree are important as SECT- and SEDT-Only (no word or character embeddings) give more than 30 to EM score. This finding also motivated us to investigate the role of syntactic trees for SQuAD dataset, but not in neural network setup.



(a) A SECT example

(b) A SEDT example

FIGURE 2.18: Two examples are used to illustrate how the syntactic information is encoded for SECT and SEDT respectively. Take Bi-directional LSTM (Hochreiter and Schmidhuber, 1997) as examples, where **x** is a vector such as word embedding, **v** and **u** are the outputs of the forward and backward LSTMs respectively. For SECT, the encoding is the syntactic sequence (NP, PP, VP) for the word **coordinator** in Figure 2.17. The fixed vectors for syntactic tags (e.g., NP, PP and VP) is in use, initialized with multivariate normal distribution. The final representation for the target word "coordinator" can be represented as the concatenation (**Ew**,**u**,**v**), where **Ew** is the word embedding for "coordinator" that is 100 dimensions in the experiments and each of the encoded vector **u** and **v** can be 30 dimensional. For SEDT, encoded word "unit" in Figure 2.16 with its dependent nodes including "Conference", "is" ,"the", "basic", "organization", ordered by their positions in the original sentence. Each word is represented with its word embedding. Similar to SECT, the final representation is the concatenation (**Ew**, **u**, **v**), which will be sent to the input layer of a neural network.

Reinforced mnemonic reader for Machine Comprehension

Reinforced mnemonic reader (ensemble) + A2D model is currently the only one which surpassed Human performance on EM. At the moment of writing there is no publication available for the full model including A2D, but just Reinforced Mnemonic Reader (M-Reader) is proposed by Hu, Peng, and Qiu (2017), model architecture is presented in the Figure 2.20. This work was motivated by the limitations observed on previously published models: there is a lack of usage of syntactic and linguistic information (POS-tags, named entities, query category) in the encoder layer, an LSTM/GRU-based interaction layer fails to fully capture the long-distance contextual interaction between parts of the context, in the pointer layer boundary detection strategies are not strong enough for detection of answer boundaries which are fuzzy or too long.

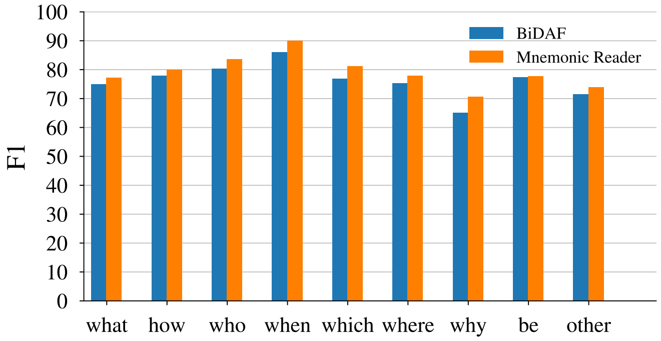
Similarly to previous models, M-Reader's encoding layer relies on character and word level embeddings, but differs in employing binary exact match feature (a word is both in context and question) and additionally POS and named entity tags embeddings which are concatenated with the word embedding. Additionally, each query gets an explicit query category embedding (made out of top-9 query-categories, e.g. *who*, *where*, *when*). The *semantic fusion unit* has the main difference at interaction layer and memory-based answer pointing module is a novel part of the pointer layer of the end-to-end neural network architecture. Performance of the ensemble version of M-Reader is given in Table 2.5. Ablation studies on SQuAD development set have shown that removal of feature-rich encoder decreases the overall performance less than other extensions of M-Reader (Table 2.8).

Also, 'why'-questions are still the hardest to answer, according to the results reported in Figure 2.19

Model	EM	F1
M-Reader+RL	**72.1**	**81.6**
M-Reader	71.8	81.2
- feature-rich encoder	70.5	80.1
- interactive aligning	65.2	74.3
- self aligning	69.7	78.9
- memory-based answer pointer	70.1	79.8

TABLE 2.8: Ablation results on SQuAD dev set.

FIGURE 2.19: M-reader F1 results by question type comparing to BiDAF model



Gated Self-Matching Networks for Reading Comprehension and Question Answering

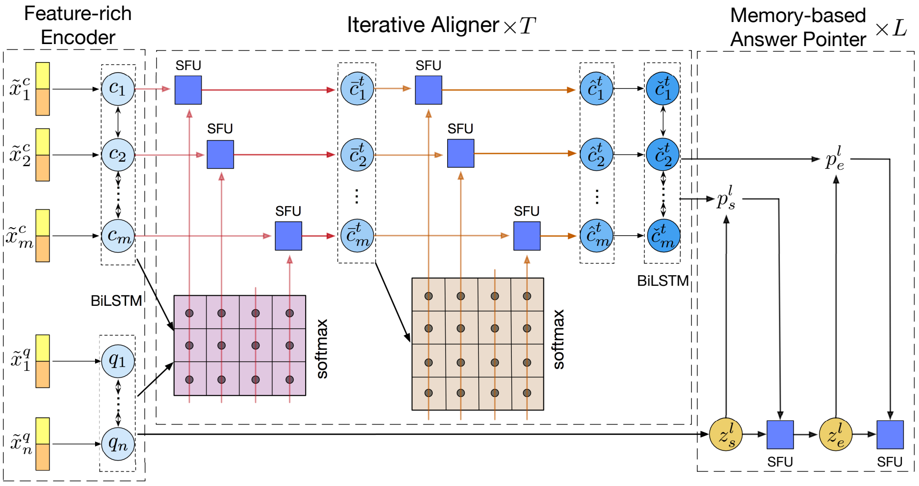
R-Net (Wang et al., 2017) proposed by Microsoft Research Asia, is an end-to-end neural network model fore MRC and QA. It's in the top-3 models on the SQuAD leaderboard at the moment of writing. The novelty of the work is related to the modifications at attention layer - new gated attention-based recurrent neural network to aim the goal of assign different levels of importance to passage parts depending of their relevance to the question, and also new self-matching layer to dynamically refine passage representation with information from the while passage (due to limits of RNN capturing long-distance dependencies in a sentence).

Other important findings are failed attempts to improve the architecture (1) by adding syntax information like POS tag, NER results, linearized PCFG tree tags, dependency labels, and (2) representing the extractive QA task as a sequence of sentence ranking and span selection subtasks (either as a separate module independently of answer selection from a sentence or in a multi-task setting), (3) dependency parsing trees. These unsuccessful attempts can imply the limits of extensibility of neural networks approaches with structured data, like syntactic trees.

MEMEN: Multi-layer Embedding with Memory Networks for Machine Comprehension

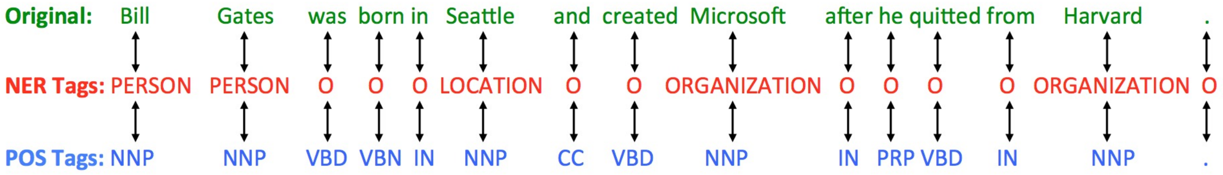
New MRC neural network model architecture, MEMEN, was introduced by Pan et al. (2017). It was motivated by inability of previous models to handle key words (important for an answer) differently from the rest of words in a sentence. There are two main contributions of the work. First is a new multi-layer embedding, which employs POS and NER tags besides word and character level embeddings (Figure 2.21). Second is memory networks of full-orientation matching. Authors reported

FIGURE 2.20: M-Reader model architecture



ablation studies (Figure 2.22), which tells that both syntactic (POS tags) and semantic (NER tags) embeddings contribute towards the model performance.

FIGURE 2.21: The passage and its according transformed “passages“. The first row (green) is the original sentence from the passage, the second row(red) is the name-entity recognition (NER) tags, and the last row (blue) is the part-of-speech (POS) tags.



Original:	Bill	Gates	was	born	in	Seattle	and	created	Microsoft	after	he	quitted	from	Harvard	.
NER Tags:	PERSON	PERSON	O	O	O	LOCATION	O	O	ORGANIZATION	O	O	O	O	ORGANIZATION	O
POS Tags:	NNP	NNP	VBD	VBN	IN	NNP	CC	VBD	NNP	IN	PRP	VBD	IN	NNP	.

FIGURE 2.22: MEMEN: Ablation results on the SQuAD dev set

Ablation Part	EM	F1
Syntactic Embedding	69.92	79.77
Semantic Embedding	70.57	80.13
Integral Query Matching	68.43	78.72
Query-Based Similarity Matching	61.26	72.25
Context-Based Similarity Matching	67.40	76.34
MEMEN	70.98	80.36

There are more related works, which in particular try to improve previously proposed neural architectures, e.g. Shen et al. (2017) discovered that multiple-turn reasoning outperforms single-turn reasoning for all question and answer types; and also that enabling a flexible number of turns generally improves upon a fixed multiple-turn strategy. Chen et al. (2017) proposed lexical gating mechanism to dynamically combine the words and characters and Interactive Attention and Memory Network.

Chapter 3

Experiments

This work was motivated by the evidence of importance of syntactic structures in open-domain QA, and for neural networks empowered MRC (models overview in Sec. 2.5) on large-scale datasets. Our goal was to fill the gap (to our knowledge no one tried it before) of a missing baseline, which doesn't employ neural network models, but was successfully applied to QA datasets like TREC before. State-of-art model for TREC in our focus is based on the open source framework RelTextRank[1] proposed by Tymoshenko et al. (2017), and the exact model explained in Severyn and Moschitti (2013). The framework described as a flexible Java pipeline for converting pairs of raw texts into structured representations and enriching them with semantic information about the relations between the two pieces of text. An important feature of the framework is a flexible way of combining different types of features (syntactic, semantic, question type, question focus, etc.).

Model and framework

Before applying the model, we studied the dataset in more details.

Detailed analysis of SQuAD

The dataset has been in the focus of many recent MRC studies, thus we mention here some characteristic reported before and also our findings. Rajpurkar et al., 2016 set logistic regression baseline on the dataset and reported ablation studies which emphasized lexicalized and dependency tree path features importance, results are presented in the Table 3.1.

[1] https://github.com/iKernels/RelTextRank

	Train	Dev
Logistic Regression	91.7%	51.0%
– Lex., – Dep. Paths	**33.9**%	**35.8**%
– Lexicalized	53.5%	45.4%
– Dep. Paths	91.4%	46.4%
– Match. Word Freq.	91.7%	48.1%
– Span POS Tags	91.7%	49.7%
– Match. Bigram Freq.	91.7%	50.3%
– Constituent Label	91.7%	50.4%
– Lengths	91.8%	50.5%
– Span Word Freq.	91.7%	50.5%
– Root Match	91.7%	50.6%

TABLE 3.1: F1 performance with feature ablations.

Xie and Xing (2017) reported that about 70% answers are exactly constituents (N = 0) and about 97% answers differ from the closest constituents by less or equal to 4 words, Figure 3.1.

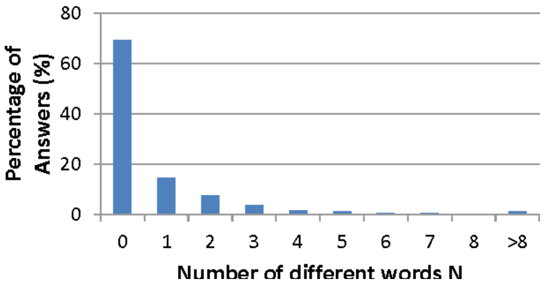


FIGURE 3.1: Percentage of answers that differ from their closest constituents by *N* words

To get the data, we used NLTK Python library (Loper and Bird, 2002). First, we found some unnatural (question, answer, passage) triples, which, in our opinion, can be a result of a crowd-sourcing way of collecting the dataset:

- ’What means’, ’Advaita’, ’Advaita literally means "not two, sole, unity".’
- ’k’, ’ks’, ’Nanjing (listen; Chinese: .., "Southern Capital") is the city situated in the heartland of lower Yangtze River region in China, which has long been a major centre of culture, education, research, politics, economy, transport networks and tourism.’
- ’j’, ’Ch’, <same as previous>
- ’n’, ’n’, <same as previous>
- ’b’, ’b’, <same as previous>
- ’v’, ’v’, <same as previous>
- ’dd’, set of answers with the same question (’Buddhism’, ’yptian Se’, ’Buddh’, ’m and E’), ’The religious sphere expanded to include new gods such as the Greco-Egyptian Serapis, eastern deities such as Attis and Cybele and the Greek adoption of Buddhism.’
- And cloze-style example: ’Himachal is?’,’multireligional, multicultural as well as multilingual state like other Indian states’, ’It is a multireligional, multicultural as well as multilingual state like other Indian states.’

After, we gathered data on minimum, maximum and average of the length of answer sentences, Table 3.2, and Figure 3.2 gives an overview of answer span length by question type (first question word). The reported results confirm that ’why’-type of questions are the most challenging. Though, surprisingly, on average an answer span to a ’how’-type question is not longer that ’what’ or ’where’. Similar results on train and dev sets can be an another reason for a good neural network model performance.

Additionally, we studied distribution of question by the first 1,2,3-gram to get a shallow understanding of what type of questions are major part in the set.

We studied the distributions of top-20 types of question by first word (unigram): Figures 3.3, 3.5, first two: Figures 3.7, 3.8 and first three words: Figures 3.9, 3.10. This

Metric	Train	Dev
Avg. sent. length	31.785	33.13
Min sent. length	2	2
Max sent. length	382	263
Avg. ans. span length	3.37	3.08
Avg. ans. span proportion to whole sentence	0.129	0.115

TABLE 3.2: By number of words in an answer span/sentence

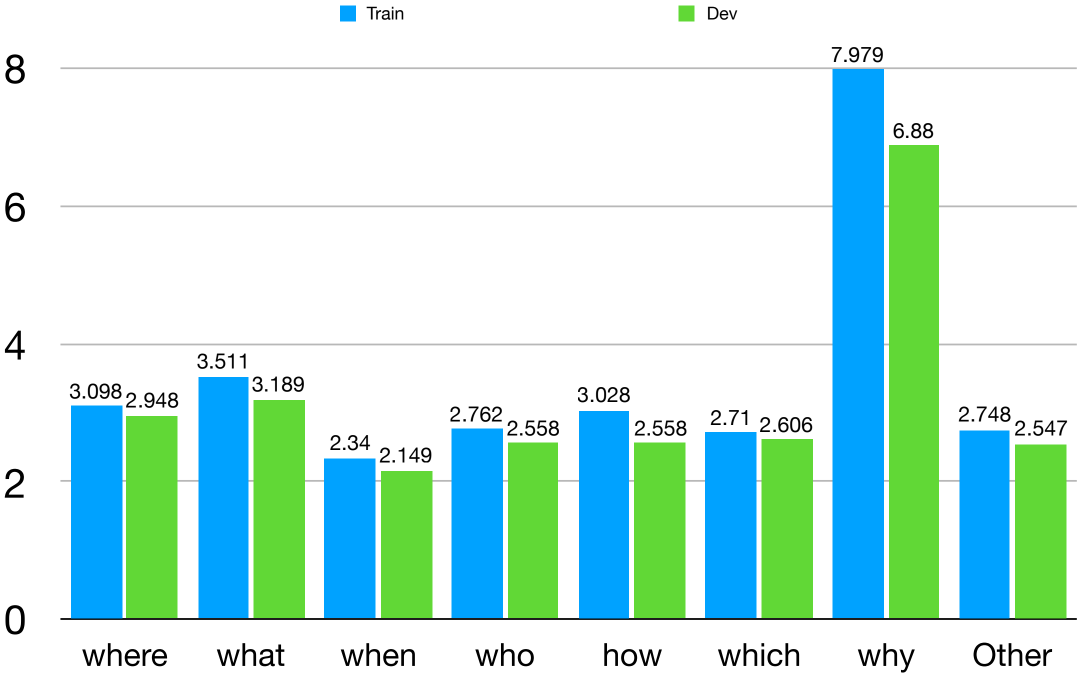


FIGURE 3.2: Average answer span length by words on Train and Dev sets

study can be important as we apply the model based which has answer type (HUM, LOCATION, ENTITY, DATE, QUANTITY, CURRENCY) as feature. We conclude that the prevailing part of the dataset is of ’what’-type.

Training and development sets are distributed similarly according to this analysis, which can make it easier for some statistical models, like neural networks to reach comparatively high performance.

According to the analysis based on the first word of a question, Figures 3.3 , 3.4, despite the majority of question are to an entity (what, who, which), there are also ’how’- and ’why’-types, which usually implies non-factoid answers.

To have a better understanding of the distribution on answer sentences for the task of answer sentence selection, we collected the data on a sequential number of an answer sentence in a passage in Figures 3.11, 3.12. We conclude, that majority of answers are in top-5 sentences in a paragraph. This can help to optimize training performance of the model.

Task setting

Usually, question answering can be divided into three steps: document retrieval, sentence answer selection and answer extraction. According to Brill et al. (2002)

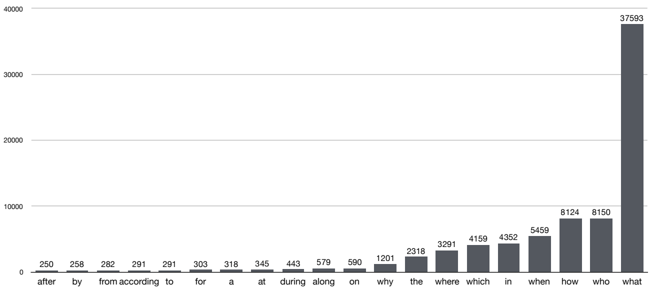


FIGURE 3.3: Top-20 first unigrams in questions in SQuAD **Train** set

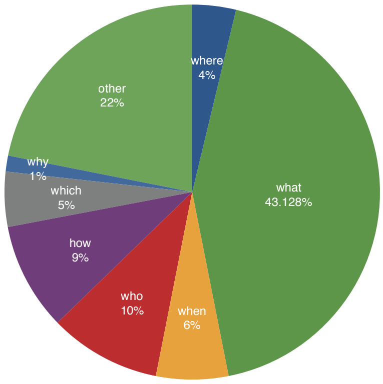


FIGURE 3.4: Unigram-based question types on **Train** set

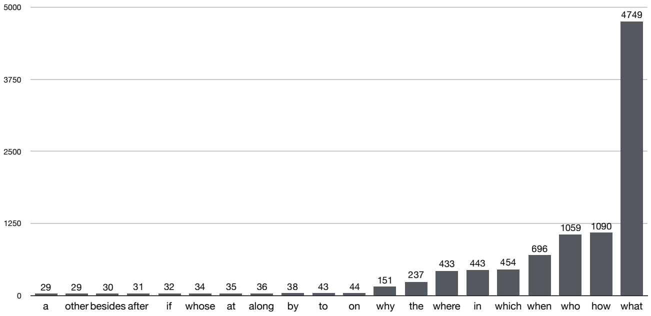


FIGURE 3.5: Top-20 first unigrams in questions in SQuAD **Dev** set

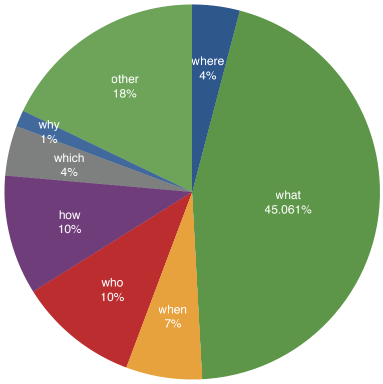


FIGURE 3.6: Unigram-based question types on **Train** set

and Rajpurkar et al. (2016), one key difference between SQuAD MRC setting and answer extraction is that answer extraction typically exploits the fact that the answer occurs in multiple documents, while in SQuAD the system has access to a single reading passage. The formal setting of the task of extractive question answering was discussed in Sec. 2.3. We remind here that the inputs are a question (as a single sentence) and a text paragraph, which usually consists of multiple sentences, the output is a text span which is not longer than a sentence. Thus, the task can be also considered as two sequential tasks:

- Answer *sentence selection*: finding the best sentence candidate from a passage;
- Answer *span extraction*: extracting exact answer span from the best sentence candidate.

Experiments

We tried both settings: applying the model end-to-end, and also trying to solve only second task separately, considering the first as comparatively easier (Ghigi et al. (2017) reports 83.8% of Sentence Level Accuracy reached by proposed Deep Word Match on SQuAD, the model doesn't employ any neural network based approaches).

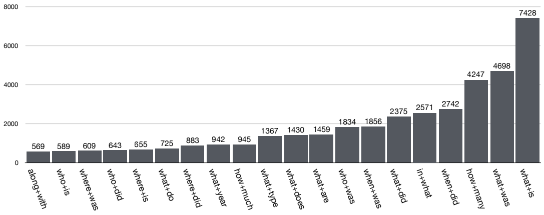


FIGURE 3.7: Top-20 first bigrams in questions in **Train** set

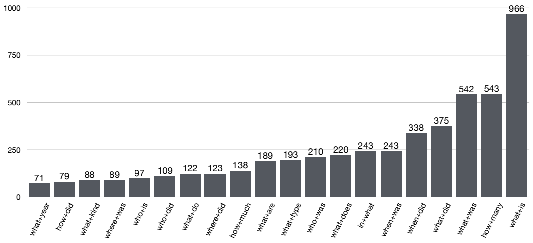


FIGURE 3.8: Top-20 first bigrams in questions in **Dev** set

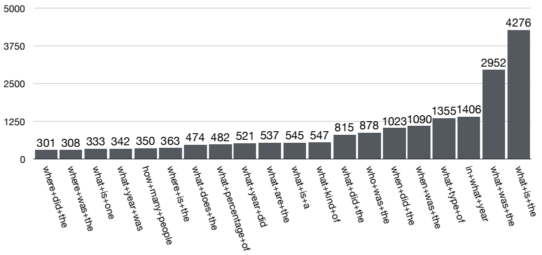


FIGURE 3.9: Top-20 first trigrams in questions in SQuAD **Train** set

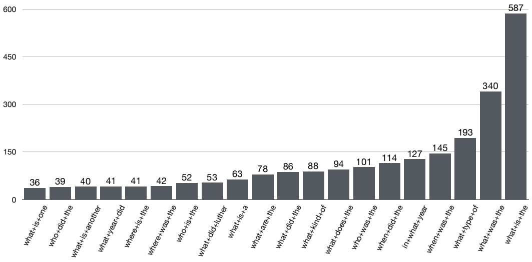


FIGURE 3.10: Top-20 first trigrams in questions in SQuAD **Dev** set

First, in end-to-end setting, consisting of two tasks, we trained a model only on 100 data-points and testing on the whole development set, and got **EM** = 3.7748, **F1** = 6.6125.

Then we decided to relax the conditions to set a better upper-bound for our method. At train time we reduced the input passage to only answer sentence, keeping development set untouched (whole passage is used during evaluation), on 100 examples for training we got **EM**: 6.000, **F1**: 12.000, and while training on 500 samples, we got the decrease on F1: **EM**: 6.8496, **F1**: 9.5828.

Thus we continued the exploration by further task relaxation and reduced both training and development passages to a single answer sentence, which is equal to solving only *span extraction* task. Results were predictably higher (train - 100 samples, development - whole set) **EM**: 20.1135, **F1**: 29.0432. The rest of the results are depicted in Figures 3.13, 3.14.

Example of the processing pipeline

In this section we give an overview on system data-flow with a step-by-step example. The framework overview is given in Figure 3.15.

First, we converted original SQuAD data into the format, which can be read by Input module: three files are for answers, questions, answer passages (as from the dataset we have only positive examples, all samples are labeled positively). The format of one sample in the answer file is the following: [*datapoint-id datapoint-id-seq-number positive-label positive-label-float positive answer-text*]

An answer and a question formats are [*datapoint-id answer-text*] and [*datapoint-id question-text*] accordingly.

The real example is:

- **Passage** (here answer sentence is marked by a color): 5733be284776f41900661182 5733be284776f41900661182-0 1 1.0 true *Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary.*

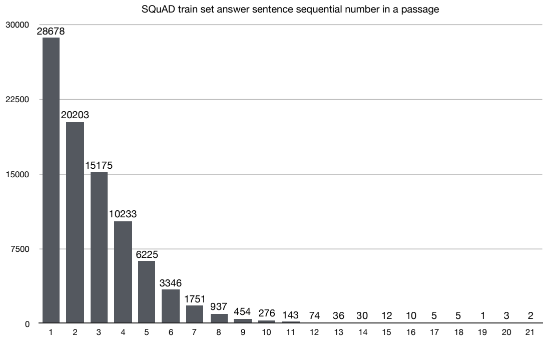


FIGURE 3.11: Sequential number of an answer sentence, **Train** set

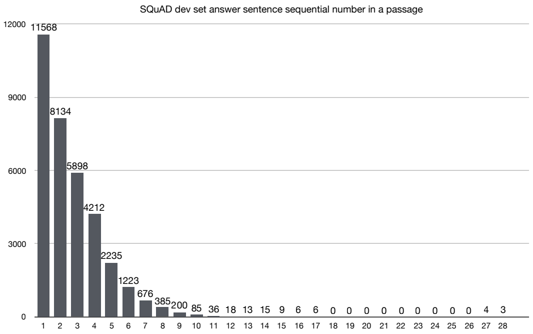


FIGURE 3.12: Sequential number of an answer sentence, **Dev** set

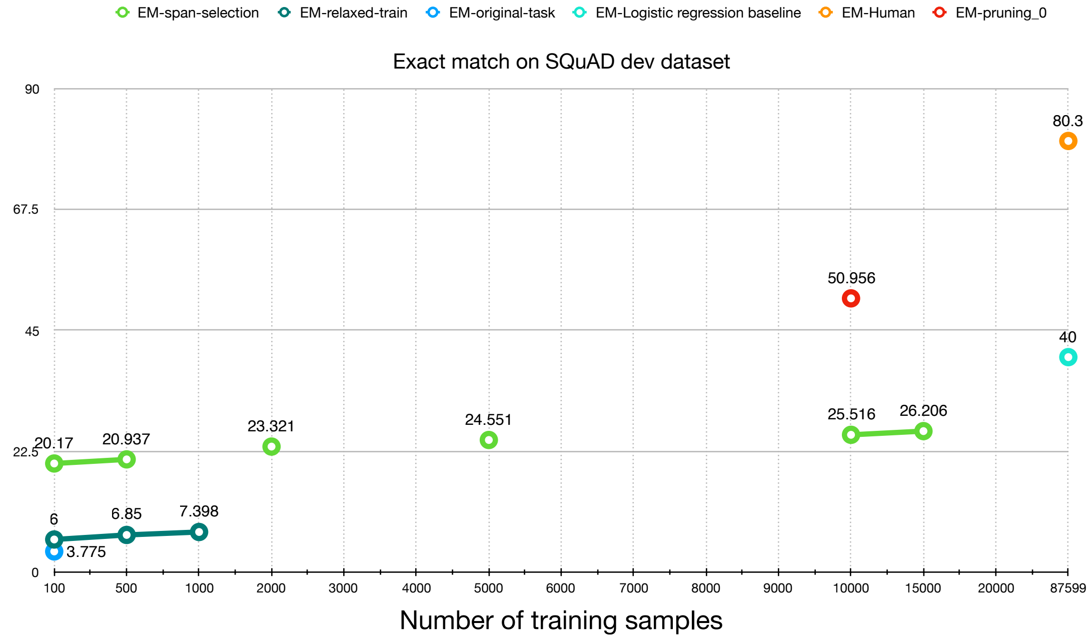


FIGURE 3.13: SVM+TK EM values of the experiments on SQuAD

> *Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.*

- **Question**: 5733be284776f41900661182 *To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?*
- **Answer**: 5733be284776f41900661182 *Saint Bernadette Soubirous*

Following the system architecture scheme in Figure 3.15, the processing pipeline employs the following steps:

- *Linguistic annotation*
  A pipeline of UIMA Analysis Engines (AEs), which wrap linguistic annotators, e.g., Sentence Splitters, Tokenizers, Syntactic parsers, to convert the input text

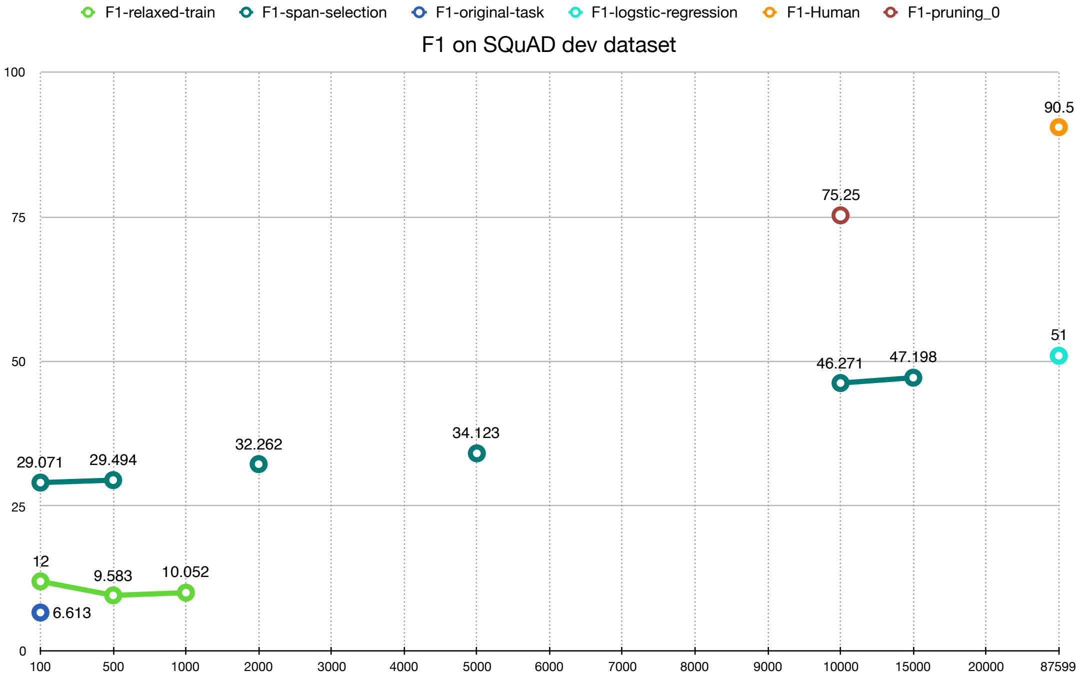


FIGURE 3.14: SVM+TK F1 values of the experiments on SQuAD

pairs into the UIMA Common Analysis Structures (CAS). CASes contain the original texts and all the linguistic annotations produced by Answer Extractor. These produce linguistic annotations defined by a UIMA Type System.

- *Generation of structural representations and feature vectors*
  The Experiment module uses CASes as input and generates the relational structures on trees (T) along with their feature vector representation (FV).

- *Generation of the output files*
  For training the model in *train* mode or for ranking at *test* mode. A real example follows.

An fragment of the output example, which is then used by SVM-light-TK 1.5 library (Moschitti, 2006) (written in C programming language) is given below:

- +1/-1 depicts a class of the example (correct/incorrect)

- Question tree: *(ROOT (S (PP (TO (to))) (NP (WP (whom))) (VP (VBD (do))) (REL-NP (DT (the)) (REL-NNP (virgin)) (REL-NNP (mary))) (ADVP (RB (allegedly))) (VP (VBP (appear))) (PP (IN (in))) (REL-NP (REL-CD (1858))) (PP (IN (in))) (REL-NP (REL-NNP (lourdes)) (REL-NNP (france)))))* Figure 3.16

- Answer tree *(ROOT (S (NP (PRP (it))) (VP (VBZ (be))) (NP (DT (a)) (NN (replica))) (PP (IN (of))) (NP (DT (the)) (NN (grotto))) (PP (IN (at))) (REL-NP (REL-NNP (lourdes))) (REL-NP (REL-NNP (france))) (ADVP (WRB (where))) (REL-FOCUS-HUM-NP (DT (the)) (REL-NNP (virgin)) (REL-NNP (mary))) (ADVP (RB (reputedly))) (VP (VBD (appear))) (PP (TO (to))) (ANS-NP (NNP (saint)) (NNP (bernadette)) (NNP (soubirous))) (PP (IN (in))) (REL-NP (REL-CD (1858)))))* Figure 3.17

- Similarity feature vector:
  17:0.5393193716300061
  18:0.1734944795898721

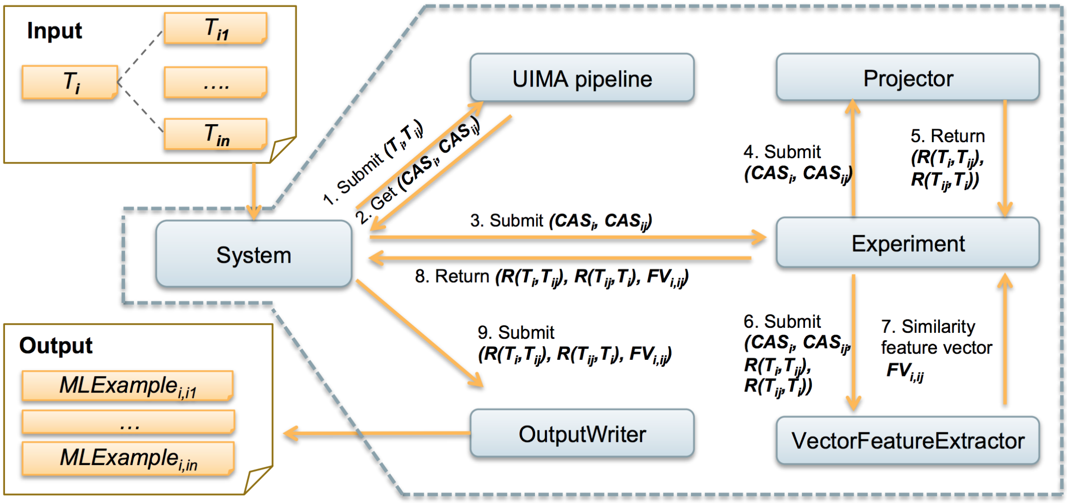

Input
T_i
T_{i1}
….
T_{in}
UIMA pipeline
Projector
1. Submit (T_i, T_{ij})
2. Get (CAS_i, CAS_{ij})
4. Submit (CAS_i, CAS_{ij})
5. Return $(R(T_i, T_{ij}), R(T_{ij}, T_i))$
System
3. Submit (CAS_i, CAS_{ij})
Experiment
8. Return $(R(T_i, T_{ij}), R(T_{ij}, T_i), FV_{i,ij})$
Output
$MLExample_{i,i1}$
…
$MLExample_{i,in}$
9. Submit $(R(T_i, T_{ij}), R(T_{ij}, T_i), FV_{i,ij})$
6. Submit $(CAS_i, CAS_{ij}, R(T_i, T_{ij}), R(T_{ij}, T_i))$
7. Similarity feature vector $FV_{i,ij}$
OutputWriter
VectorFeatureExtractor

FIGURE 3.15: Overall RelTextRank framework

19:0.44008622942335207 ...
36:0.6647577087403779

The list of similar examples is used for training SVM model. Similarly, for testing, the same structure is produced by adding all possible pairs of trees. Afterwards, they are ranked to obtain the best answer candidate.

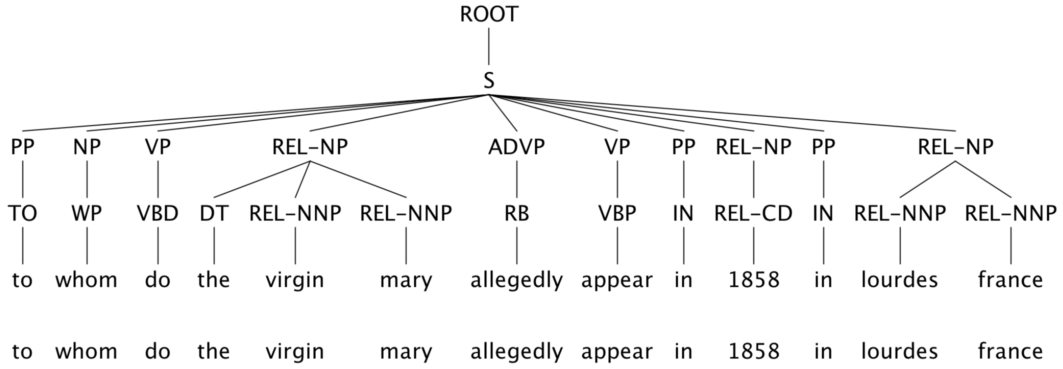


FIGURE 3.16: An example of a question syntactic tree

Error Analysis

For the model trained on 5,000 samples, we found, that in 6% of samples the predicted answer is wider than the ground truth, which is a result of the model predicting constituents, while real answer can contain any part of a constituent; +14% of predictions are shorter than a ground truth, i.e. the ground truth fully covers the prediction, which again can imply the necessity of a span span borders 'adjustment' module.

Qualitative analysis had revealed that majority of the cases of missed correct predictions are due to detecting either wrong noun phrase or errors in numerical answers, e.g. for the question *What is the mace displayed in?*, the predicted answer was *july 1999* while ground truths answers are *'glass case suspended from lid'*, *'glass case'*; and for the question *When was the Scottish Constitutional Convention held?* the prediction was *industry* for ground truth *' 1989'*.

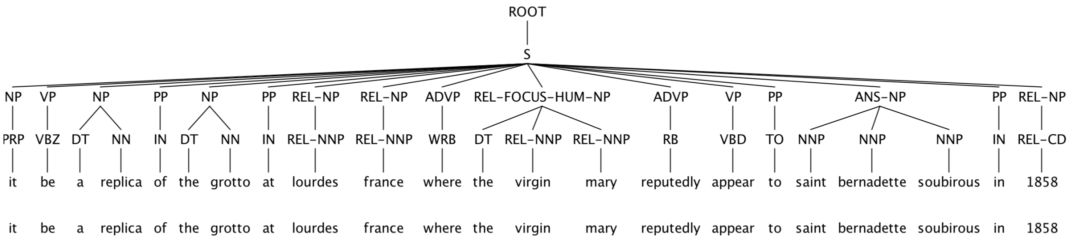

ROOT
S
NP VP NP PP NP PP REL−NP REL−NP ADVP REL−FOCUS−HUM−NP ADVP VP PP ANS−NP PP REL−NP
PRP VBZ DT NN IN DT NN IN REL−NNP REL−NNP WRB DT REL−NNP REL−NNP RB VBD TO NNP NNP NNP IN REL−CD
it be a replica of the grotto at lourdes france where the virgin mary reputedly appear to saint bernadette soubirous in 1858
it be a replica of the grotto at lourdes france where the virgin mary reputedly appear to saint bernadette soubirous in 1858

FIGURE 3.17: An example of an answer sentence syntactic tree

To set up an upper boundary and do a qualitative analysis of the errors, we further relaxed the task by employing pruning strategy (example of the pruned tree in Figures 3.18, 3.19), where the are less candidates of artificially generated incorrect answers, e.g. pruning ray = 0 mean that for training per question-answer pair we have two examples:

It yielded much better results: while training on 10,000 examples we got 50.956 for EM and 75.244 on F1, 488 questions were unanswered (answer is "none") and incorrectly answered were 4695.

In the same setting number of span predictions which were not correct, but were covered completely by ground truth is about 22% higher, while prediction was wider than the ground truth in 21% of cases, which shows the direction for future work, i.e. improvements on detecting exact boundaries which can cover a part of one or multiple sequential constituents. The rest of the errors are due to wrong parsing of the input, thus losing some characters, e.g. the prediction was *4 record*, while answer was '12-4 record', or *'31 24'*, when ground truth is *'31–24'*; another error type is a wrong boundary prediction, where the prediction has no intersection with a ground truth.

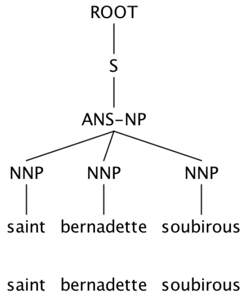


FIGURE 3.18: An example of an answer sentence syntactic tree pruned with a pruning ray equals to 0, which means that no neighboring REL constituents are considered

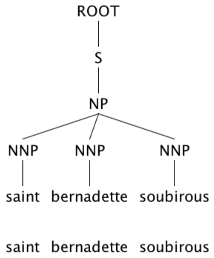


FIGURE 3.19: An example of 'incorrect' artificial sample of syntactic tree pruned with a pruning ray equals to 0. Differs from 3.18 by POS tag under **S** - NP.

Chapter 4

Conclusions and future work

Observing the rapid development of the field, in this work we have explored modern methods based on neural networks applied to large-scale machine reading comprehension datasets, and made an attempt to set up a new baseline based on the SVM+TK model, which was proven as an effective way of preserving syntactic information of language on traditional, comparatively low-scale datasets.

Discussion

Significant number research has recently done on creating datasets and according models for MRC task. Though many complex neural architectures report high performance results, it doesn't always mean that the tasks of MRC and NLU are solved, as better evaluation metrics show a significant decrease of performance. Also simpler models show that the introduced complexity of architectures is not always necessary. Deeper analysis of new datasets sometimes also provide hints on potential reasons of high performance, e.g. majority of questions do not go along with claims of required multi-sentence reasoning. Also, there are more and more neural network architectures, which attempt to incorporate structural information, like constituency or dependency trees, they show different level of success.

In the work we applied SVM+TK model, but realized that the main difficulties were mainly due to efficiency problems. In order to make the Tree Kernel model applicable and win some upper bounds, we used several relaxation approaches.

Another contribution is adapting RelTextRank framework to extract original surface form while detecting the answer span, while previously it was only lemmatized constituents.

Though we didn't get expected results facing computational challenges, there are many ways to try for the model improvement, which are listed below.

Future work

We consider the following steps to make next:

- Computation performance improvements optimization to train on the whole dataset
- Training on separate question types to get a better understanding on ability of the system to answer real explanatory questions (non-factoid)
- Test on real test dataset[1] (not available to public).

[1] https://worksheets.codalab.org/ is used as a platform of submission. We couldn't solve the issue of running our modular system on the platform due to inability of the platform to handle input files generated dynamically. The last we hope to solve by replacing C modules with equivalent Java modules with the help of machine learning library KeLP (Filice et al., 2015)

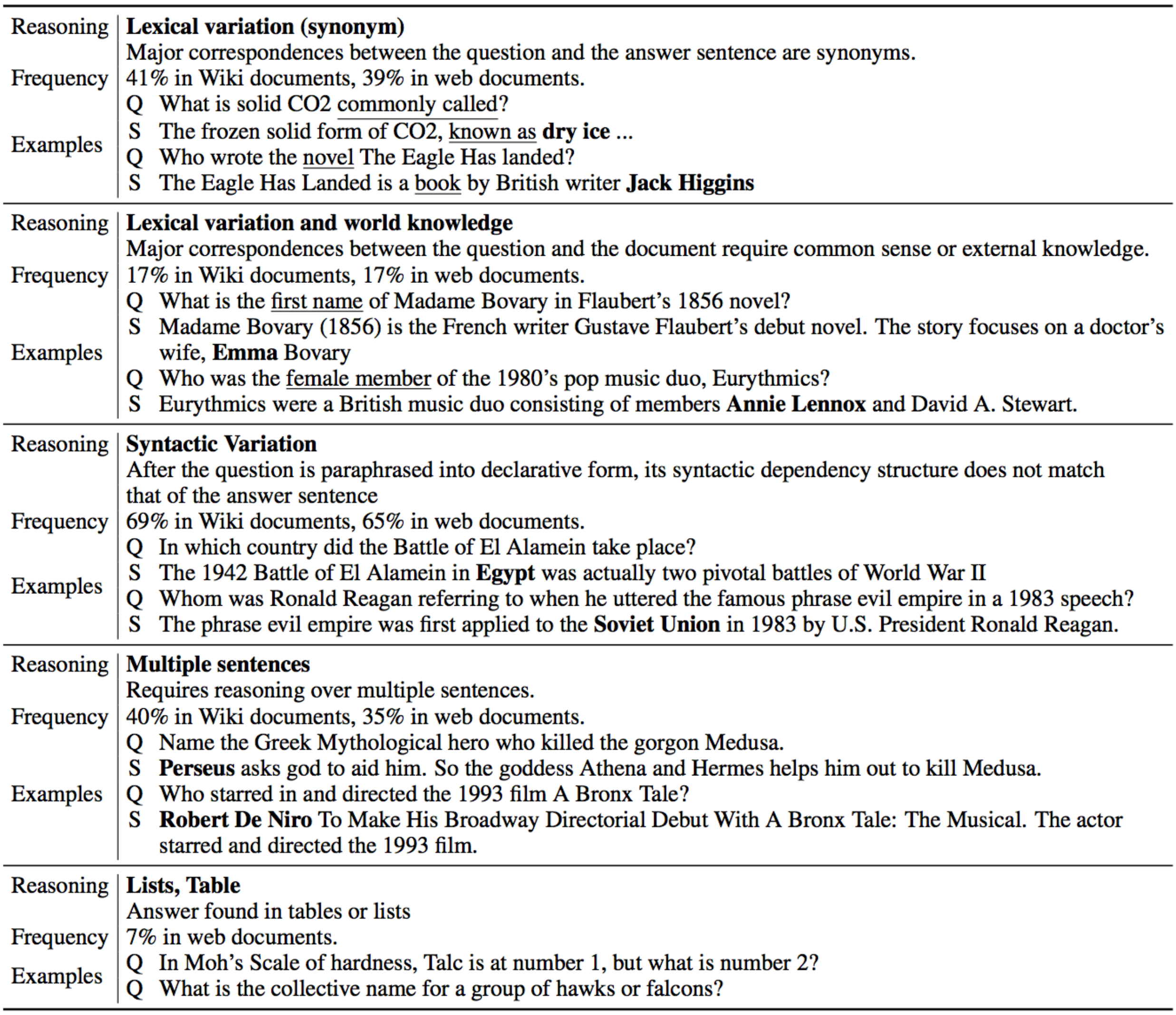
- Test on AddAny and AddSent metrics (Jia and Liang, 2017)
- We'd like to do ablation studies, mostly on features in the similarity feature vector used alongside syntactic trees. Depending on the results, we consider adding other similarity features to improve the model performance.
- We believe that one way to improve the accuracy could be also by a richer number of experiments on SVM parameters tuning (Regularization, Gamma and Margin)
- More experiments with depth of syntactic or constituency trees

Despite the reported performance, this work shapes some research directions to use TK models in large datasets like SQuAD. So we nevertheless see an optimism in future attempts to adopt SVM with PTKs model on SQuAD or similar dataset, as there are more research done in incorporating structural information into the models for MRC.

Appendix A

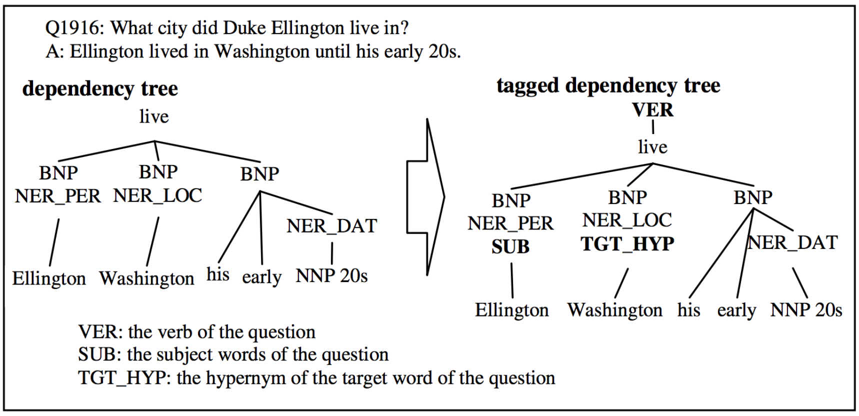
Appendix A

FIGURE A.1: Analysis of reasoning used to answer TriviaQA questions shows that a high proportion of evidence sentence(s) exhibit syntactic and lexical variation with respect to questions. Answers are indicated by boldfaced text (Joshi et al., 2017)



Reasoning		**Lexical variation (synonym)**
		Major correspondences between the question and the answer sentence are synonyms.
Frequency		41% in Wiki documents, 39% in web documents.
Examples	Q	What is solid CO2 commonly called?
	S	The frozen solid form of CO2, known as **dry ice** ...
	Q	Who wrote the novel The Eagle Has landed?
	S	The Eagle Has Landed is a book by British writer **Jack Higgins**
Reasoning		**Lexical variation and world knowledge**
		Major correspondences between the question and the document require common sense or external knowledge.
Frequency		17% in Wiki documents, 17% in web documents.
Examples	Q	What is the first name of Madame Bovary in Flaubert's 1856 novel?
	S	Madame Bovary (1856) is the French writer Gustave Flaubert's debut novel. The story focuses on a doctor's wife, **Emma** Bovary
	Q	Who was the female member of the 1980's pop music duo, Eurythmics?
	S	Eurythmics were a British music duo consisting of members **Annie Lennox** and David A. Stewart.
Reasoning		**Syntactic Variation**
		After the question is paraphrased into declarative form, its syntactic dependency structure does not match that of the answer sentence
Frequency		69% in Wiki documents, 65% in web documents.
Examples	Q	In which country did the Battle of El Alamein take place?
	S	The 1942 Battle of El Alamein in **Egypt** was actually two pivotal battles of World War II
	Q	Whom was Ronald Reagan referring to when he uttered the famous phrase evil empire in a 1983 speech?
	S	The phrase evil empire was first applied to the **Soviet Union** in 1983 by U.S. President Ronald Reagan.
Reasoning		**Multiple sentences**
		Requires reasoning over multiple sentences.
Frequency		40% in Wiki documents, 35% in web documents.
Examples	Q	Name the Greek Mythological hero who killed the gorgon Medusa.
	S	**Perseus** asks god to aid him. So the goddess Athena and Hermes helps him out to kill Medusa.
	Q	Who starred in and directed the 1993 film A Bronx Tale?
	S	**Robert De Niro** To Make His Broadway Directorial Debut With A Bronx Tale: The Musical. The actor starred and directed the 1993 film.
Reasoning		**Lists, Table**
		Answer found in tables or lists
Frequency		7% in web documents.
Examples	Q	In Moh's Scale of hardness, Talc is at number 1, but what is number 2?
	Q	What is the collective name for a group of hawks or falcons?

FIGURE A.2: Dependency tree and tagged dependency tree (Shen, Kruijff, and Klakow, 2005)


Q1916: What city did Duke Ellington live in?
A: Ellington lived in Washington until his early 20s.
dependency tree
live
BNP
NER_PER
BNP
NER_LOC
BNP
NER_DAT
Ellington
Washington
his
early
NNP 20s
tagged dependency tree
VER
live
BNP
NER_PER
SUB
BNP
NER_LOC
TGT_HYP
BNP
NER_DAT
Ellington
Washington
his
early
NNP 20s
VER: the verb of the question
SUB: the subject words of the question
TGT_HYP: the hypernym of the target word of the question

Dataset	Documents	Questions	Answers
MCTest (Richardson, 2013)	660 short stories, grade school level	2640 human generated, based on the document	multiple choice
CNN/Daily Mail (Hermann et al., 2015)	93K+220K news articles	387K+997K Cloze-form, based on highlights	entities
BookTest (Bajgar, Kadlec, and Kleindienst, 2016)	14.2M, similar to CBT	Cloze-form, similar to CBT	multiple choice
SQuAD (Rajpurkar et al., 2016)	23K paragraphs from 536 Wikipedia articles	108K human generated, based on the paragraphs	spans
NewsQA (Trischler et al., 2016b)	13K news articles from the CNN dataset	120K human generated, based on headline, highlights	spans
MS MARCO (Nguyen et al., 2016)	1M passages from 200K+ documents retrieved using the queries	100K search queries	human generated, based on the passages
SearchQA (Dunn et al., 2017)	6.9m passages retrieved from a search engine using the queries	140k human generated Jeopardy! questions	human generated Jeopardy! answers
NarrativeQA (Kočiský et al., 2017)	1,572 stories (books, movie scripts) & human generated summaries	46,765 human generated, based on summaries	human generated, based on summaries
WikiHOP (Welbl, Stenetorp, and Riedel, 2017)	598,103 support passages from Wikipedia & automatically generated summaries	43,738 human generated, based on Wikipedia paragraphs	human generated, based on summaries

TABLE A.1: Compiled table of MRC datasets. Extended, based on mentioned by Kočiský et al. (2017)

Bibliography

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473. arXiv: 1409.0473. URL: http://arxiv.org/abs/1409.0473.

Bajgar, Ondrej, Rudolf Kadlec, and Jan Kleindienst (2016). "Embracing data abundance: BookTest Dataset for Reading Comprehension". In: *CoRR* abs/1610.00956. arXiv: 1610.00956. URL: http://arxiv.org/abs/1610.00956.

Brill, E. et al. (2002). "An Analysis of the AskMSR Question-Answering System". In: *Proceedings of EMNLP 2002*. URL: https://www.microsoft.com/en-us/research/publication/an-analysis-of-the-askmsr-question-answering-system/.

Burges, Chris J.C. (2013). *Towards the Machine Comprehension of Text: An Essay*. Tech. rep. URL: https://www.microsoft.com/en-us/research/publication/towards-the-machine-comprehension-of-text-an-essay/.

Chen, Danqi, Jason Bolton, and Christopher D. Manning (2016). "A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task". In: *CoRR* abs/1606.02858. arXiv: 1606.02858. URL: http://arxiv.org/abs/1606.02858.

Chen, Zheqian et al. (2017). "Smarnet: Teaching Machines to Read and Comprehend Like Human". In: *CoRR* abs/1710.02772. arXiv: 1710.02772. URL: http://arxiv.org/abs/1710.02772.

Collins, Michael and Nigel Duffy (2002). "New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 263–270. DOI: 10.3115/1073083.1073128. URL: https://doi.org/10.3115/1073083.1073128.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Dunn, Matthew et al. (2017). "SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine". In: *CoRR* abs/1704.05179. arXiv: 1704.05179. URL: http://arxiv.org/abs/1704.05179.

Filice, Simone et al. (2015). "KeLP: a Kernel-based Learning Platform in Java". In: *The workshop on Machine Learning Open Source Software (MLOSS): Open Ecosystems*. Lille, France: International Conference of Machine Learning.

Ghigi, Fabrizio et al. (2017). *Sentence Answer Selection for Open Domain Question Answering via Deep Word Matching*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Heilman, Michael and Noah A Smith (2010). "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*. Association for Computational Linguistics, pp. 1011–1019.

Hermann, Karl Moritz et al. (2015). “Teaching machines to read and comprehend”. In: *Advances in Neural Information Processing Systems*, pp. 1693–1701.

Hewlett, Daniel et al. (2016). “WikiReading: A Novel Large-scale Language Understanding Task over Wikipedia”. In: *CoRR* abs/1608.03542. arXiv: 1608.03542. URL: http://arxiv.org/abs/1608.03542.

Hill, Felix et al. (2015). “The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representations”. In: *CoRR* abs/1511.02301. arXiv: 1511.02301. URL: http://arxiv.org/abs/1511.02301.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.

Hu, M., Y. Peng, and X. Qiu (2017). “Reinforced Mnemonic Reader for Machine Comprehension”. In: *ArXiv e-prints*. arXiv: 1705.02798 [cs.CL].

Huang, Hsin-Yuan et al. (2017). “FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension”. In: *CoRR* abs/1711.07341. arXiv: 1711.07341. URL: http://arxiv.org/abs/1711.07341.

Jia, Robin and Percy Liang (2017). “Adversarial Examples for Evaluating Reading Comprehension Systems”. In: *CoRR* abs/1707.07328. arXiv: 1707.07328. URL: http://arxiv.org/abs/1707.07328.

Joshi, Mandar et al. (2017). “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension”. In: *CoRR* abs/1705.03551. arXiv: 1705.03551. URL: http://arxiv.org/abs/1705.03551.

Kočiský, T. et al. (2017). “The NarrativeQA Reading Comprehension Challenge”. In: *ArXiv e-prints*. arXiv: 1712.07040 [cs.CL].

Lai, Guokun et al. (2017). “RACE: Large-scale ReAding Comprehension Dataset From Examinations”. In: *CoRR* abs/1704.04683. arXiv: 1704.04683. URL: http://arxiv.org/abs/1704.04683.

Lee, Kenton et al. (2016). “Learning Recurrent Span Representations for Extractive Question Answering”. In: *CoRR* abs/1611.01436. arXiv: 1611.01436. URL: http://arxiv.org/abs/1611.01436.

Liu, Rui et al. (2017). “Structural Embedding of Syntactic Trees for Machine Comprehension”. In: *CoRR* abs/1703.00572. arXiv: 1703.00572. URL: http://arxiv.org/abs/1703.00572.

Loper, Edward and Steven Bird (2002). “NLTK: The Natural Language Toolkit”. In: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP ’02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 63–70. DOI: 10.3115/1118108.1118117. URL: https://doi.org/10.3115/1118108.1118117.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press. ISBN: 0-262-13360-1.

Moschitti, Alessandro (2006). “Efficient convolution kernels for dependency and constituent syntactic trees”. In: *European Conference on Machine Learning*. Springer, pp. 318–329.

Nakov, Preslav et al. (2016). “SemEval-2016 Task 3: Community Question Answering”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pp. 525–545. URL: http://aclweb.org/anthology/S/S16/S16-1083.pdf.

Nguyen, Tri et al. (2016). “MS MARCO: A Human Generated MAchine Reading COmprehension Dataset”. In: *CoRR* abs/1611.09268. arXiv: 1611.09268. URL: http://arxiv.org/abs/1611.09268.

Pan, Boyuan et al. (2017). "MEMEN: Multi-layer Embedding with Memory Networks for Machine Comprehension". In: *CoRR* abs/1707.09098. arXiv: 1707.09098. URL: http://arxiv.org/abs/1707.09098.

Paperno, Denis et al. (2016). "The LAMBADA dataset: Word prediction requiring a broad discourse context". In: *CoRR* abs/1606.06031. arXiv: 1606.06031. URL: http://arxiv.org/abs/1606.06031.

Rajpurkar, Pranav et al. (2016). "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". In: *CoRR* abs/1606.05250. arXiv: 1606.05250. URL: http://arxiv.org/abs/Â§1606.05250.

Ravichandran, Deepak and Eduard H. Hovy (2002). "Learning surface text patterns for a Question Answering System". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.* Pp. 41–47. URL: http://www.aclweb.org/anthology/P02-1006.pdf.

MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text (2013). URL: https://www.microsoft.com/en-us/research/publication/mctest-challenge-dataset-open-domain-machine-comprehension-text/.

Riloff, Ellen and Michael Thelen (2000). "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*. Association for Computational Linguistics, pp. 13–19.

Seo, Min Joon et al. (2016). "Bidirectional Attention Flow for Machine Comprehension". In: *CoRR* abs/1611.01603. arXiv: 1611.01603. URL: http://arxiv.org/abs/1611.01603.

Severyn, Aliaksei and Alessandro Moschitti (2013). "Automatic Feature Engineering for Answer Selection and Extraction." In: *EMNLP*. Vol. 13, pp. 458–467.

Shen, Dan and Dietrich Klakow (2006). "Exploring correlation of dependency relation paths for answer extraction". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 889–896.

Shen, Dan, Geert-Jan M Kruijff, and Dietrich Klakow (2005). "Exploring syntactic relation patterns for question answering". In: *International Conference on Natural Language Processing*. Springer, pp. 507–518.

Shen, Yelong et al. (2017). "An Empirical Analysis of Multiple-Turn Reasoning Strategies in Reading Comprehension Tasks". In: *CoRR* abs/1711.03230. arXiv: 1711.03230. URL: http://arxiv.org/abs/1711.03230.

Shih, Cheng-Wei et al. (2005). "ASQA: Academia sinica question answering system for NTCIR-5 CLQA". In: *NTCIR-5 Workshop, Tokyo, Japan*, pp. 202–208.

Trischler, Adam et al. (2016a). "A Parallel-Hierarchical Model for Machine Comprehension on Sparse Data". In: *CoRR* abs/1603.08884. arXiv: 1603.08884. URL: http://arxiv.org/abs/1603.08884.

Trischler, Adam et al. (2016b). "NewsQA: A Machine Comprehension Dataset". In: *CoRR* abs/1611.09830. arXiv: 1611.09830. URL: http://arxiv.org/abs/1611.09830.

Ture, Ferhan and Oliver Jojic (2016). "Simple and Effective Question Answering with Recurrent Neural Networks". In: *arXiv preprint arXiv:1606.05029*.

Tymoshenko, Kateryna, Daniele Bonadiman, and Alessandro Moschitti (2016). "Learning to rank non-factoid answers: Comment selection in web forums". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 2049–2052.

Tymoshenko, Kateryna et al. (2017). "RelTextRank: An Open Source Framework for Building Relational Syntactic-Semantic Text Pair Representations". In: *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, pp. 79–84. URL: http://www.aclweb.org/anthology/P17-4014.

Vinyals, O., M. Fortunato, and N. Jaitly (2015). "Pointer Networks". In: *ArXiv e-prints*. arXiv: 1506.03134 [stat.ML].

Voorhees, Ellen M and DM Tice (2000). "Overview of the TREC-9 Question Answering Track." In: *TREC*.

Wang, Hai et al. (2015). "Machine comprehension with syntax, frames, and semantics". In: *In Proceedings of ACL: Short*.

Wang, Mengqiu (2006). "A survey of answer extraction techniques in factoid question answering". In: *Computational Linguistics* 1.1.

Wang, Mengqiu and Christopher D Manning (2010). "Probabilistic tree-edit models with structured latent variables for textual entailment and question answering". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, pp. 1164–1172.

Wang, Mengqiu, Noah A Smith, and Teruko Mitamura (2007). "What is the Jeopardy model? A quasi-synchronous grammar for QA". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Wang, Wenhui et al. (2017). "Gated self-matching networks for reading comprehension and question answering". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 189–198.

Weissenborn, Dirk, Georg Wiese, and Laura Seiffe (2017a). "FastQA: A Simple and Efficient Neural Architecture for Question Answering". In: *arXiv preprint arXiv:1703.04816*.

— (2017b). "Making neural qa as simple as possible but not simpler". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 271–280.

Welbl, Johannes, Pontus Stenetorp, and Sebastian Riedel (2017). "Constructing Datasets for Multi-hop Reading Comprehension Across Documents". In: *CoRR* abs/1710.06481. arXiv: 1710.06481. URL: http://arxiv.org/abs/1710.06481.

Weston, Jason et al. (2015). "Towards ai-complete question answering: A set of prerequisite toy tasks". In: *arXiv preprint arXiv:1502.05698*.

Xie, Pengtao and Eric Xing (2017). "A Constituent-Centric Neural Architecture for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 1405–1414.

Xiong, Caiming, Victor Zhong, and Richard Socher (2016). "Dynamic coattention networks for question answering". In: *arXiv preprint arXiv:1611.01604*.

Yao, Xuchen et al. (2013). "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*, pp. 858–867.

Yu, Lei et al. (2014). "Deep Learning for Answer Sentence Selection". In: *CoRR* abs/1412.1632. arXiv: 1412.1632. URL: http://arxiv.org/abs/1412.1632.

Yu, Yang et al. (2016). "End-to-End Reading Comprehension with Dynamic Answer Chunk Ranking". In: *CoRR* abs/1610.09996. arXiv: 1610.09996. URL: http://arxiv.org/abs/1610.09996.