Natural Language Processing with Deep Learning

(Textual) Question Answering
John Christian Watson

John Christian Watson (born John Christian Tanck; 9 April 1867 – 18 November 1941), commonly known as Chris Watson, was an Australian politician who served as the third Prime Minister of Australia.

Chris Watson - Wikipedia

People also search for
Andrew Fisher
George Reid
Billy Hughes
Edmund Barton
Alfred Deakin
Kevin Rudd
Julia Gillard

More about Chris Watson
Motivation: Question answering

• With massive collections of full-text documents, i.e., the web 😊, simply returning relevant documents is of limited use
• Rather, we often want answers to our questions
• Especially on mobile
• Or using a digital assistant device, like Alexa, Google Assistant, ...

• We can factor this into two parts:
  1. Finding documents that (might) contain an answer
     • Which can be handled by traditional information retrieval/web search

  2. Finding an answer in a paragraph or a document
     • This problem is often termed Reading Comprehension
     • It is what we will focus on today
A Brief History of Reading Comprehension

- Much early NLP work attempted reading comprehension
  - Schank, Abelson, Lehnert et al. c. 1977 – “Yale A.I. Project”
- Revived by Lynette Hirschman in 1999:
  - Could NLP systems answer human reading comprehension questions for 3rd to 6th graders? Simple methods attempted.
- Revived again by Chris Burges in 2013 with MCTest
  - Again answering questions over simple story texts
- Floodgates opened in 2015/16 with the production of large datasets which permit supervised neural systems to be built
  - Hermann et al. (NIPS 2015) DeepMind CNN/DM dataset
  - Rajpurkar et al. (EMNLP 2016) SQuAD
  - MS MARCO, TriviaQA, RACE, NewsQA, NarrativeQA, ...
Machine Comprehension (Burges 2013)

• “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.”
Alyssa got to the beach after a long trip. She's from Charlotte. She traveled from Atlanta. She's now in Miami. She went to Miami to visit some friends. But she wanted some time to herself at the beach, so she went there first. After going swimming and laying out, she went to her friend Ellen's house. Ellen greeted Alyssa and they both had some lemonade to drink. Alyssa called her friends Kristin and Rachel to meet at Ellen's house…….

Why did Alyssa go to Miami?

To visit some friends
A Brief History of Open-domain Question Answering

- Simmons et al. (1964) did first exploration of answering questions from an expository text based on matching dependency parses of a question and answer
- Murax (Kupiec 1993) aimed to answer questions over an online encyclopedia using IR and shallow linguistic processing
- The NIST TREC QA track begun in 1999 first rigorously investigated answering fact questions over a large collection of documents
- IBM’s Jeopardy! System (DeepQA, 2011) brought attention to a version of the problem; it used an ensemble of many methods
- DrQA (Chen et al. 2016) uses IR followed by neural reading comprehension to bring deep learning to Open-domain QA
Turn-of-the Millennium Full NLP QA:
[architecture of LCC (Harabagiu/Moldovan) QA system, circa 2003]
Complex systems but they did work fairly well on “factoid” questions
Question: Which team won Super Bowl 50?

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.
Stanford Question Answering Dataset (SQuAD)

Private schools, also known as independent schools, non-governmental, or nonstate schools, are not administered by local, state or national governments; thus, they retain the right to select their students and are funded in whole or in part by charging their students tuition, rather than relying on mandatory taxation through public (government) funding; at some private schools students may be able to get a scholarship, which makes the cost cheaper, depending on a talent the student may have (e.g. sport scholarship, art scholarship, academic scholarship), financial need, or tax credit scholarships that might be available.

Along with non-governmental and nonstate schools, what is another name for private schools?
Gold answers: ① independent ② independent schools ③ independent schools

Along with sport and art, what is a type of talent scholarship?
Gold answers: ① academic ② academic ③ academic

Rather than taxation, what are private schools largely funded by?
Gold answers: ① tuition ② charging their students tuition ③ tuition
SQuAD evaluation, v1.1

• Authors collected 3 gold answers

• Systems are scored on two metrics:
  • Exact match: 1/0 accuracy on whether you match one of the 3 answers
  • F1: Take system and each gold answer as bag of words, evaluate
    \[
    \text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN}, \quad \text{harmonic mean} \quad F1 = \frac{2PR}{P+R}
    \]
    Score is (macro-)average of per-question F1 scores

• F1 measure is seen as more reliable and taken as primary
  • It’s less based on choosing exactly the same span that humans chose, which is susceptible to various effects, including line breaks

• Both metrics ignore punctuation and articles (a, an, the only)
SQuAD v1.1 leaderboard, 2019-02-07 – it’s solved!

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Human Performance</td>
<td>82.304</td>
<td>91.221</td>
</tr>
<tr>
<td></td>
<td>Stanford University</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Rajpurkar et al. '16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BERT (ensemble)</td>
<td>87.433</td>
<td>93.160</td>
</tr>
<tr>
<td></td>
<td>Google AI Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BERT (single model)</td>
<td>85.083</td>
<td>91.835</td>
</tr>
<tr>
<td></td>
<td>Google AI Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="https://arxiv.org/abs/1810.04805">https://arxiv.org/abs/1810.04805</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>nlnet (ensemble)</td>
<td>85.356</td>
<td>91.202</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>nlnet (ensemble)</td>
<td>85.954</td>
<td>91.677</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>QANet (ensemble)</td>
<td>84.454</td>
<td>90.490</td>
</tr>
<tr>
<td></td>
<td>Google Brain &amp; CMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>r-net (ensemble)</td>
<td>84.003</td>
<td>90.147</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>QANet (ensemble)</td>
<td>83.877</td>
<td>89.737</td>
</tr>
<tr>
<td></td>
<td>Google Brain &amp; CMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>nlnet (single model)</td>
<td>83.468</td>
<td>90.133</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>nlnet (ensemble)</td>
<td>85.954</td>
<td>91.677</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SQuAD 2.0

- A defect of SQuAD 1.0 is that all questions have an answer in the paragraph
- Systems (implicitly) rank candidates and choose the best one
- You don’t have to judge whether a span answers the question
- In SQuAD 2.0, 1/3 of the training questions have no answer, and about 1/2 of the dev/test questions have no answer
  - For NoAnswer examples, NoAnswer receives a score of 1, and any other response gets 0, for both exact match and F1
- Simplest system approach to SQuAD 2.0:
  - Have a threshold score for whether a span answers a question
  - Or you could have a second component that confirms answering
    - Like Natural Language Inference (NLI) or “Answer validation”
Genghis Khan united the Mongol and Turkic tribes of the steppes and became Great Khan in 1206. He and his successors expanded the Mongol empire across Asia. Under the reign of Genghis' third son, Ögedei Khan, the Mongols destroyed the weakened Jin dynasty in 1234, conquering most of northern China. Ögedei offered his nephew Kublai a position in Xingzhou, Hebei. Kublai was unable to read Chinese but had several Han Chinese teachers attached to him since his early years by his mother Sorghaghtani. He sought the counsel of Chinese Buddhist and Confucian advisers. Möngke Khan succeeded Ögedei's son, Güyük, as Great Khan in 1251. He

When did Genghis Khan kill Great Khan?

**Gold Answers:** <No Answer>

**Prediction:** 1234 [from Microsoft nlnet]
# SQuAD 2.0 leaderboard, 2019-02-07

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Human Performance</td>
<td>86.831</td>
<td>89.452</td>
</tr>
</tbody>
</table>
|      | *Stanford University*  
  *(Rajpurkar & Jia et al. '18)* |      |      |
| 1    | BERT + MMFT + ADA (ensemble) | 85.082 | 87.615 |
|      | *Microsoft Research Asia* |      |      |
| 2    | BERT + Synthetic Self-Training (ensemble) | 84.292 | 86.967 |
|      | *Google AI Language*  
  [https://github.com/google-research/bert](https://github.com/google-research/bert) |      |      |
| 3    | BERT finetune baseline (ensemble) | 83.536 | 86.096 |
|      | *Anonymous* |      |      |
| 4    | Lunet + Verifier + BERT (ensemble) | 83.469 | 86.043 |
|      | *Layer 6 AI NLP Team* |      |      |
| 4    | PAML+BERT (ensemble model) | 83.457 | 86.122 |
|      | *PINGAN GammaLab* |      |      |
| 5    | Lunet + Verifier + BERT (single model) | 82.995 | 86.035 |
|      | *Layer 6 AI NLP Team* |      |      |
Good systems are great, but still basic NLU errors

The Yuan dynasty is considered both a successor to the Mongol Empire and an imperial Chinese dynasty. It was the khanate ruled by the successors of Möngke Khan after the division of the Mongol Empire. In official Chinese histories, the Yuan dynasty bore the Mandate of Heaven, following the Song dynasty and preceding the Ming dynasty. The dynasty was established by Kublai Khan, yet he placed his grandfather Genghis Khan on the imperial records as the official founder of the

What dynasty came before the Yuan?

Gold Answers: ① Song dynasty ② Mongol Empire ③ the Song dynasty

Prediction: Ming dynasty [BERT (single model) (Google AI)]
SQuAD limitations

- SQuAD has a number of other key limitations too:
  - Only span-based answers (no yes/no, counting, implicit why)
  - Questions were constructed looking at the passages
    - Not genuine information needs
  - Barely any multi-fact/sentence inference beyond coreference

- Nevertheless, it is a well-targeted, well-structured, clean dataset
  - It has been the most used and competed on QA dataset
  - It has also been a useful starting point for building systems in industry (though in-domain data always really helps!)
NewsQA dataset consists of 100,000 question-answer pairs from CNN news articles. For other datasets like WikiQA, the span is the entire sentence containing the answer (Yang et al., 2015); the task of choosing a sentence rather than a smaller answer span is sometimes called the sentence selection task.

These reading comprehension datasets are used both as a reading comprehension task in themselves and as a training set and evaluation set for the sentence extraction component of open question answering algorithms.

**Basic Reading Comprehension Algorithm.**

Neural algorithms for reading comprehension are given a question $q$ of $l$ tokens $q_1, \ldots, q_l$ and a passage $p$ of $m$ tokens $p_1, \ldots, p_m$. Their goal is to compute, for each token $p_i$, the probability $p_{\text{start}}(i)$ that $p_i$ is the start of the answer span, and the probability $p_{\text{end}}(i)$ that $p_i$ is the end of the answer span.

Fig. 23.8 shows the architecture of the Document Reader component of the DrQA system of Chen et al. (2017). Like most such systems, DrQA builds an embedding for the question, builds an embedding for each token in the passage, computes a similarity function between the question and each passage word in context, and then uses the question-passage similarity scores to decide where the answer span starts and ends.

Let's consider the algorithm in detail, following closely the description in Chen et al. (2017). The question is represented by a single embedding $q$, which is a weighted sum of representations for each question word $q_i$. It is computed by passing the series of embeddings $P E(q_1), \ldots, E(q_l)$ of question words through an RNN (such as a bi-LSTM shown in Fig. 23.8). The resulting hidden representations $\{q_1, \ldots, q_l\}$ are combined by a weighted sum:

$$q = \sum_j b_j q_j$$ (23.9)

**Training objective:**

$$\mathcal{L} = - \sum \log P_{\text{start}}(a_{\text{start}}) - \sum \log P_{\text{end}}(a_{\text{end}})$$
Stanford Attentive Reader++

- **p_i**: Vector representation of each token in passage
  Made from concatenation of
  - Word embedding (GloVe 300d)
  - Linguistic features: POS & NER tags, one-hot encoded
  - Term frequency (unigram probability)
  - Exact match: whether the word appears in the question
    - 3 binary features: exact, uncased, lemma
  - Aligned question embedding ("car" vs "vehicle")

\[
    f_{\text{align}}(p_i) = \sum_j a_{i,j} E(q_j)
\]

\[
    q_{i,j} = \frac{\exp(\alpha(E(p_i)) \cdot \alpha(E(q_j)))}{\sum_{j'} \exp(\alpha(E(p_i)) \cdot \alpha(E(q_{j'}))))}
\]

Where \(\alpha\) is a simple one layer FFNN
**BiDAF: Bi-Directional Attention Flow for Machine Comprehension**
(Seo, Kembhavi, Farhadi, Hajishirzi, ICLR 2017)
BiDAF

- There are variants of and improvements to the BiDAF architecture over the years, but the central idea is the **Attention Flow layer**

- **Idea:** attention should flow both ways – from the context to the question and from the question to the context

- Make similarity matrix (with \( \mathbf{w} \) of dimension \( 6d \)):

\[
S_{ij} = \mathbf{w}_{\text{sim}}^T [c_i; q_j; c_i \odot q_j] \in \mathbb{R}
\]

- Context-to-Question (C2Q) attention:
  (which query words are most relevant to each context word)

\[
\alpha^i = \text{softmax}(S_{i,:}) \in \mathbb{R}^M \quad \forall i \in \{1, \ldots, N\}
\]

\[
\mathbf{a}_i = \sum_{j=1}^{M} \alpha^i_j \mathbf{q}_j \in \mathbb{R}^{2h} \quad \forall i \in \{1, \ldots, N\}
\]
BiDAF

- **Attention Flow Idea**: attention should flow both ways – from the context to the question and from the question to the context

- **Question-to-Context (Q2C) attention**: (the weighted sum of the most important words in the context with respect to the query – slight asymmetry through max)

\[
m_i = \max_j S_{ij} \in \mathbb{R} \quad \forall i \in \{1, \ldots, N\}
\]

\[
\beta = \text{softmax}(m) \in \mathbb{R}^N
\]

\[
c' = \sum_{i=1}^{N} \beta_i c_i \in \mathbb{R}^{2h}
\]

- For each passage position, output of BiDAF layer is:

\[
b_i = [c_i; a_i; c_i \circ a_i; c_i \circ c'] \in \mathbb{R}^{8h} \quad \forall i \in \{1, \ldots, N\}
\]
BiDAF

- There is then a “modelling” layer:
  - Another deep (2-layer) BiLSTM over the passage
- And answer span selection is more complex:
  - Start: Pass output of BiDAF and modelling layer concatenated to a dense FF layer and then a softmax
  - End: Put output of modelling layer M through another BiLSTM to give $M_2$ and then concatenate with BiDAF layer and again put through dense FF layer and a softmax
Dynamic Coattention Networks for Question Answering
(Caiming Xiong, Victor Zhong, Richard Socher ICLR 2017)

- Flaw: Questions have input-independent representations
- Interdependence needed for a comprehensive QA model

**Dynamic Coattention Network**

**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?

**Coattention encoder**

**Dynamic pointer decoder**
- start index: 49
- end index: 51
- *U

*LSTM+HMN (highway maxout network)
Coattention Encoder

Figure 2: Coattention encoder. The affinity matrix $L$ is not shown here. We instead directly show the normalized attention weights $A_D$ and $A_Q$. We similarly compute the summaries $C_Q A_D$ of the question in light of each word of the document. Similar to Cui et al. (2016), we also compute the summaries $C_Q A_D$ of the previous attention contexts in light of each word of the document. These two operations can be done in parallel, as is shown in Eq. 3. One possible interpretation for the operation $C_Q A_D$ is the mapping of question encoding into space of document encodings.

\[ C_D = \langle Q; C_Q A_D \rangle; \quad (3) \]

We define $C_D$, a co-dependent representation of the question and document, as the coattention context. We use the notation $[a; b]$ for concatenating the vectors $a$ and $b$ horizontally.

The last step is the fusion of temporal information to the coattention context via a bidirectional LSTM:

\[ \begin{align*}
    u_t &= \text{Bi-LSTM}(u_{t-1}, u_{t+1}); \\
    c_{Dt} &= 2 \mathbb{R}^2 \langle \cdot \rangle.
\end{align*} \]

(4)

We define $U = [u_1, ..., u_m]$ $2 \mathbb{R}^2 \langle m \rangle$, which provides a foundation for selecting which span may be the best possible answer, as the coattention encoding.

2.3 Dynamic Pointing Decoder

Due to the nature of SQuAD, an intuitive method for producing the answer span is by predicting the start and end points of the span (Wang & Jiang, 2016). However, given a question-document pair, there may exist several intuitive answer spans within the document, each corresponding to a local maxima. We propose an iterative technique to select an answer span by alternating between predicting the start point and predicting the end point. This iterative procedure allows the model to recover from initial local maxima corresponding to incorrect answer spans.

Figure 3 provides an illustration of the Dynamic Decoder, which is similar to a state machine whose state is maintained by an LSTM-based sequential model. During each iteration, the decoder updates its state taking into account the coattention encoding corresponding to current estimates of the start and end positions, and produces, via a multilayer neural network, new estimates of the start and end positions.

Let $h_i$, $s_i$, and $e_i$ denote the hidden state of the LSTM, the estimate of the position, and the estimate of the end position during iteration $i$. The LSTM state update is then described by Eq. 5.

\[ h_i = \text{LSTM}_{dec}(h_i, \langle u_{s_i}, u_{e_i} \rangle); \quad (5) \]

where $u_{s_i}$ and $u_{e_i}$ are the representations corresponding to the previous estimate of the start and end positions in the coattention encoding $U$. 

3
Coattention layer

- Coattention layer again provides a two-way attention between the context and the question
- However, coattention involves a second-level attention computation:
  - attending over representations that are themselves attention outputs
- We use the C2Q attention distributions $\alpha_i$ to take weighted sums of the Q2C attention outputs $b_j$. This gives us second-level attention outputs $s_i$:

\[
s_i = \sum_{j=1}^{M+1} \alpha^i_j b_j \in \mathbb{R}^l \quad \forall i \in \{1, \ldots, N\}
\]
Attention functions

MLP (Additive) form:
\[ S_{ij} = s^T \tanh(W_1 c_i + W_2 q_j) \]

Bilinear (Product) form:
\[ S_{ij} = c_i^T W q_j \]
\[ S_{ij} = c_i^T U^T V q_j \]
\[ S_{ij} = c_i^T W^T D W q_j \]

1. Smaller space
2. Non-linearity

\[ S_{ij} = \text{Relu}(c_i^T W^T)D\text{Relu}(W q_j) \]

D: diagonal matrix

FusionNet (Huang, Zhu, Shen, Chen 2017)
FusionNet tries to combine many forms of attention

When Context is long, Self-boosted Fusion can be used
Recent, more advanced architectures

- Most of the work in 2016, 2017, and 2018 employed progressively more complex architectures with a multitude of variants of attention – often yielding good task gains.

<table>
<thead>
<tr>
<th>Architectures</th>
<th>(1)</th>
<th>(2)</th>
<th>(2’)</th>
<th>(3)</th>
<th>(3’)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match-LSTM (Wang and Jiang, 2016)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>DCN (Xiong et al., 2017)</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FastQA (Weissenborn et al., 2017)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FastQAExt (Weissenborn et al., 2017)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BiDAF (Seo et al., 2017)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RaSoR (Lee et al., 2016)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>DrQA (Chen et al., 2017)</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MPCM (Wang et al., 2016)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Mnemonic Reader (Hu et al., 2017)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>R-net (Wang et al., 2017b)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
ELMo and BERT preview

Contextual word representations
Using language model-like objectives

The transformer architecture used in BERT is sort of attention on steroids. More later!

Elmo
(Peters et al, 2018)

Bert
(Devlin et al, 2018)

Look at SDNet as an example of how to use BERT as submodule: https://arxiv.org/abs/1812.03593

(Vaswani et al, 2017)