Who did What: A Large-Scale Person-Centered Cloze Dataset

Takeshi Onishi  Hai Wang  Mohit Bansal  Kevin Gimpel  David McAllester
Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA
{tonishi,haiwang,mbansal,kgimpel,mcallester}@ttic.edu
to appear at EMNLP 2016.

Abstract
We have constructed a new “Who-did-What” dataset of over 200,000 fill-in-the-gap (cloze) multiple choice reading comprehension problems constructed from the LDC English Gigaword newswire corpus. The WDW dataset has a variety of novel features. First, in contrast with the CNN and Daily Mail datasets (Hermann et al., 2015) we avoid using article summaries for question formation. Instead, each problem is formed from two independent articles — an article given as the passage to be read and a separate article on the same events used to form the question. Second, we avoid anonymization — each choice is a person named entity. Third, the problems have been filtered to remove a fraction that are easily solved by simple baselines, while remaining 84% solvable by humans. We report performance benchmarks of standard systems and propose the WDW dataset as a challenge task for the community.1

1 Introduction
Researchers distinguish the problem of general knowledge question answering from that of reading comprehension (Hermann et al., 2015; Hill et al., 2016). Reading comprehension is more difficult than knowledge-based or IR-based question answering in two ways. First, reading comprehension systems must infer answers from a given unstructured passage rather than structured knowledge sources such as Freebase (Bollacker et al., 2008) or the Google Knowledge Graph (Singhal, 2012). Second, machine comprehension systems cannot exploit the large level of redundancy present on the web to find statements that provide a strong syntactic match to the question (Yang et al., 2015). In contrast, a machine comprehension system must use the single phrasing in the given passage, which may be a poor syntactic match to the question.

In this paper, we describe the construction of a new reading comprehension dataset that we refer to as “Who-did-What”. Two typical examples are shown in Table 1.2 The process of forming a problem starts with the selection of a question article from the English Gigaword corpus. The question is formed by deleting a person named entity from the first sentence of the question article. An information retrieval system is then used to select a passage with high overlap with the first sentence of the question article. An information retrieval system is then used to select a passage with high overlap with the first sentence of the question article, and an answer choice list is generated from the person named entities in the passage.

Our dataset differs from the CNN and Daily Mail comprehension tasks (Hermann et al., 2015) in that it forms questions from two distinct articles rather than summary points. This allows problems to be derived from document collections that do not contain manually-written summaries. This also reduces the syntactic similarity between the question and the relevant sentences in the passage, increasing the need for deeper semantic analysis.

To make the dataset more challenging we selectively remove problems so as to suppress four simple baselines — selecting the most mentioned person, 

1Available at tticnlp.github.io/who_did_what

2The passages here only show certain salient portions of the passage. In the actual dataset, the entire article is given. The correct answers are (3) and (2).
the first mentioned person, and two language model
baselines. This is also intended to produce problems
requiring deeper semantic analysis.

The resulting dataset yields a larger gap between
human and machine performance than existing ones.
Humans can answer questions in our dataset with
an 84% success rate compared to the estimates of
75% for CNN (Chen et al., 2016) and 82% for the
CBT named entities task (Hill et al., 2016). In spite
of this higher level of human performance, various
existing readers perform significantly worse on our
dataset than they do on the CNN dataset. For ex-
ample, the Attentive Reader (Hermann et al., 2015)
achieves 63% on CNN but only 55% on Who-did-
What and the Attention Sum Reader (Kadlec et al.,
2016) achieves 70% on CNN but only 59% on Who-
did-What.

In summary, we believe that our Who-did-What
dataset is more challenging, and requires deeper se-

demic analysis, than existing datasets.

2 Related Work

Our Who-did-What dataset is related to several re-
cently developed datasets for machine comprehen-
sion. The MCTest dataset (Richardson et al., 2013)
consists of 660 fictional stories with 4 multiple
choice questions each. This dataset is too small
to train systems for the general problem of reading
comprehension.

The bAbI synthetic question answering dataset
(Weston et al., 2016) contains passages describing a
series of actions in a simulation followed by a ques-
tion. For this synthetic data a logical algorithm can
be written to solve the problems exactly (and, in fact,
is used to generate ground truth answers).

The Children’s Book Test (CBT) dataset, created
by Hill et al. (2016), consists of 113,719 cloze-style
named entity problems. Each problem consists of 20
consecutive sentences from a children’s story, a 21st
sentence in which a word has been deleted, and a list
of ten choices for the deleted word. The CBT dataset
tests story completion rather than reading compre-

dension. The next event in a story is often not de-
determined — surprises arise. This may explain why
human performance is lower for CBT than for our
dataset — 82% for CBT vs. 84% for Who-did-What.
The 16% error rate for humans on Who-did-What
seems to be largely due to noise in problem forma-
tion introduced by errors in named entity recogni-
tion and parsing. Reducing this noise in future ver-
sions of the dataset should significantly improve hu-
man performance. Another difference compared to
CBT is that Who-did-What has shorter choice lists
on average. Random guessing achieves only 10%
on CBT but 32% on Who-did-What. The reduction
in the number of choices seems likely to be responsi-
ible for the higher performance of an LSTM system on Who-did-What – contextual LSTMs (the attentive reader of Hermann et al., 2015) improve from 44% on CBT (as reported by Hill et al., 2016) to 55% on Who-did-What.

Above we referenced the comprehension datasets created from CNN and Daily Mail articles by Hermann et al. (2015). The CNN and Daily Mail datasets together consist of 1.4 million questions constructed from approximately 300,000 articles. Of existing datasets, these are the most similar to Who-did-What in that they consists of cloze-style question answering problems derived from news articles. As discussed in Section 1, our Who-did-What dataset differs from these datasets in not being derived from article summaries, in using baseline suppression, and in yielding a larger gap between machine and human performance. The Who-did-What dataset also differs in that the person named entities are not anonymized, permitting the use of external resources to improve performance while remaining difficult for language models due to suppression.

3 Dataset Construction

We now describe the construction of our Who-did-What dataset in more detail. We sketch the procedure below and provide more specific details in the appendix. To generate a problem we first generate the question by selecting a random article — the “question article” — from the Gigaword corpus and taking the first sentence of that article — the “question sentence” — as the source of the cloze question. The hope is that the first sentence of an article contains prominent people and events which are likely to be discussed in other independent articles. To convert the question sentence to a cloze question, we first extract named entities using the Stanford NER system (Finkel et al., 2005) and parse the sentence using the Stanford PCFG parser (Klein and Manning, 2003).

The person named entities are candidates for deletion to create a cloze problem. For each person named entity we then identify a noun phrase in the automatic parse that is headed by that person. For example, if the question sentence is “President Obama met yesterday with Apple Founder Steve Jobs” we identify the two person noun phrases “President Obama” and “Apple Founder Steve Jobs”. When a person named entity is selected for deletion, the entire noun phrase is deleted. For example, when deleting the second named entity, we get “President Obama met yesterday with XXX” rather than “President Obama met yesterday with Apple founder XXX”. This increases the difficulty of the problems because systems cannot rely on descriptors and other local contextual cues. About 700,000 question sentences are generated from Gigaword articles (8% of the total number of articles).

Once a cloze question has been formed we select an appropriate article as a passage. The article should be independent of the question article but should discuss the people and events mentioned in the question sentence. To find a passage we search the Gigaword dataset using the Apache Lucene information retrieval system (McCandless et al., 2010), using the question sentence as the query. The named entity to be deleted is included in the query and required to be included in the returned article. We also restrict the search to articles published within two weeks of the date of the question article. Articles containing sentences too similar to the question in word overlap and phrase matching near the blanked phrase are removed. We select the best matching article satisfying our constraints. If no such article can be found, we abort the process and move on to a new question. See the appendix for details.

Given a question and a passage we next form the list of choices. We collect all person named entities in the passage except unblanked person named entities in the question. Choices that are subsets of longer choices are eliminated. For example the choice “Obama” would be eliminated if the list also contains “Barack Obama”. We also discard ambiguous cases where a part of a blanked NE appears in multiple candidate answers, e.g., if a passage has “Bill Clinton” and “Hillary Clinton” and the blanked phrase is “Clinton”. We found this simple coreference rule to work well in practice since news articles usually employ full names for initial mentions of persons. If the resulting choice list contains fewer than two or more than five choices, the process is aborted and we move on to a new question.3

---

3The maximum of five helps to avoid sports articles contain-
After forming an initial set of problems we then remove “duplicated” problems. Duplication arises because Gigaword contains many copies of the same article or articles where one is clearly an edited version of another. Our duplication-removal process ensures that no two problems have very similar questions. Here, similarity is defined as the ratio of the size of the bag of words intersection to the size of the smaller bag.

In order to focus our dataset on the most interesting problems, we remove some problems to suppress the performance of the following simple baselines:

- First person in passage: Select the person that appears first in the passage.
- Most frequent person: Select the most frequent person in the passage.
- \( n \)-gram: Select the most likely answer to fill the blank under a 5-gram language model trained on Gigaword minus articles which are too similar to one of the questions in word overlap and phrase matching.
- Unigram: Select the most frequent last name using the unigram counts from the 5-gram model.

To minimize the number of questions removed we solve an optimization problem defined by limiting the performance of each baseline to a specified target value while removing as few problems as possible, i.e.,

\[
\max_{\alpha(C)} \sum_{C \in \{0,1\}^{|B|}} \alpha(C)|T(C)|
\]

subject to

\[
\forall i \sum_{C:C_i=1} \frac{\alpha(C)|T(C)|}{N} \leq k
\]

\[
N = \sum_{C \in \{0,1\}^{|B|}} \alpha(C)|T(C)|
\]

where \( T(C) \) is the subset of the questions solved by the subset \( C \) of the suppressed baselines, and \( \alpha(C) \) is a keeping rate for question set \( T(C) \). \( C_i = 1 \) indicates \( i \)-th baseline is in the subset and \( |B| \) is a number of baselines. Then \( N \) is a total number of questions and \( k \) is an upper bound for the baselins after suppression. \( k \) is set to the random performance.

performance of these baselines before and after suppression are shown in Table 2. The suppression removed 49.9% of the questions.

Table 3 shows statistics of our dataset after suppression. We split the final dataset into train, validation, and test by taking the validation and test to be a random split of the most recent 20,000 problems as measured by question article date. In this way there is very little overlap in semantic subject matter between the training set and either validation or test. We also provide a larger “relaxed” training set formed by applying less baseline suppression (a larger value of \( k \) in the optimization). The relaxed training set then has a slightly different distribution from the train, validation, and test sets which are all fully suppressed.

4 Performance Benchmarks

We report the performance of several systems to characterize our dataset:

- Word overlap: Select the choice \( c \) inserted to the question \( q \) which is the most similar to any sentence \( s \) in the passage, i.e., \( \text{CosSim}(\text{bag}(c + q), \text{bag}(s)) \).
- Sliding window and Distance baselines (and their combination) from Richardson et al. (2013).
• Semantic features: NLP feature based system from Wang et al. (2015).
• Attentive Reader: LSTM with attention mechanism (Hermann et al., 2015).
• Stanford Reader: An attentive reader modified with a bilinear term (Chen et al., 2016).
• Attention Sum (AS) Reader: GRU with a point-attention mechanism (Kadlec et al., 2016).
• Gated-Attention (GA) Reader: Attention Sum Reader with gated layers (Dhingra et al., 2016).

Table 4 shows the performance of each system on the test data. For the Attention and Stanford Readers, we anonymized the Who-did-What data by replacing named entities with entity IDs as in the CNN and Daily Mail datasets.

We see consistent reductions in accuracy when moving from CNN to our dataset. The Attentive and Stanford Reader drop by up to 10% and the AS and GA reader drop by up to 17%. The ranking of the systems also changes. In contrast to the Attentive/Stanford readers, the AS/GA readers explicitly leverage the frequency of the answer in the passage, a heuristic which appears beneficial for the CNN and Daily Mail tasks. Our suppression of the most-frequent-person baseline appears to more strongly affect the performance of these latter systems.

5 Conclusion

We presented a large-scale person-centered cloze dataset whose scalability and flexibility is suitable for neural methods. This dataset is different in a variety of ways from existing large-scale cloze datasets and provides a significant extension to the training and test data for machine comprehension.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2358–2367.

Bhuwan Dhingra, Hanxiao Liu, William W. Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. CoRR, abs/1606.01549.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 363–370.

Karl Moritz Hermann, Tomáš Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems, pages 1684–1692.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks principle: Reading children’s books with explicit memory representations. In Proceedings of International Conference on Learning Representations.

Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 908–918.
6 Appendix

We include pseudocode for generating questions (Alg. 1) and multiple choice answer sets (Alg. 2).

Algorithm 1: Question Formation. Named entities are recognized by the Stanford NER system and parse trees are generated by the Stanford PCFG parser. Here $X.category$ is the syntactic category of parse node $X$, $X.parent$ is the parent-node of the node $X$, $X.descendant$ is the set of descendants of $X$ and $X.head$ is the head word of $X$. 

---

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, pages 423–430.

Michael McCandless, Erik Hatcher, and Otis Gospodnetic. 2010. Lucene in Action, Second Edition. Manning Publications Co.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 193–203.

Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. Official Google blog.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 700–706.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards AI-complete question answering: A set of prerequisite toy tasks. In Proceedings of International Conference on Learning Representations.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WIKIQA: A challenge dataset for open-domain question answering. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2013–2018.
Data: a pair \((q, e)\) returned by Algorithm 1.

Result: either null, if no appropriate passage can be found, or a passage \(a\) and multiple choice answer set \(C\).

\[
p \leftarrow \text{null}
\]

\[
\text{for } a \in \text{RankedArticles}(q, e) \text{ do}
\]

\[
C \leftarrow \text{The set of person NEs in } a \text{ different from } e \text{ and not in } q. \text{ Named entities appearing as sub-part of an earlier named entity are deleted.}
\]

\[
\text{if } 2 \leq |E| \leq 5 \text{ then return } (a, C)
\]

end

return null

\[
\text{RankedArticles}(q, e)\{
\]

\[
A_r \leftarrow \emptyset
\]

\[
\text{for } t \in \{1, 3, 7, 14\} \text{ do}
\]

\[
A \leftarrow \text{Articles}(q, e, t)
\]

\[
\text{for } a \in A \text{ do}
\]

\[
\text{if isValid}(a, q) \text{ then}
\]

\[
A_r \leftarrow A_r \text{ followed by } a
\]

end

end

return \(A_r\)}

\[
\text{Articles}(q, e, t)\{
\]

\[
\text{Result: articles containing the person NE } e, \text{ published within } t \text{ days of the article from which } q \text{ was taken, and ranked by Apache Lucene.}
\]

\[
\text{isValid}(a, q)\{
\]

\[
a \text{ is a valid passage for } q \text{ if the following hold:}
\]

\[
\text{• no sentence in } a \text{ shares more than 78\% of its words with the question } q.
\]

\[
\text{• no sentence in } a \text{ contains the sequence of five words to the left of the blank in } q, \text{ and similarly for the sequence to the right.}
\]

\[
\text{• } a \text{ contains at least one of the person NEs in } q. \text{(All person NEs in } q \text{ are different from } e. \text{ Two named entities are considered the same if they share some words.)}
\]

\[
\text{Algorithm 2: Passage Selection}
\]