Query-Focused Scenario Construction

Su Wang[1,2] **Greg Durrett**[3] **Katrin Erk**[1]
[1]Department of Linguistics
[2]Department of Statistics and Data Science
[3]Department of Computer Science
The University of Texas at Austin
shrekwang@utexas.edu gdurrett@cs.utexas.edu katrin.erk@mail.utexas.edu

Abstract

The news coverage of events often contains not one but multiple incompatible accounts of what happened. We develop a query-based system that extracts compatible sets of events (scenarios) from such data, formulated as one-class clustering. Our system incrementally evaluates each event's compatibility with already selected events, taking order into account. We use synthetic data consisting of article mixtures for scalable training and evaluate our model on a new human-curated dataset of scenarios about real-world news topics. Stronger neural network models and harder synthetic training settings are both important to achieve high performance, and our final scenario construction system substantially outperforms baselines based on prior work.

Query: Jamal Khashoggi was murdered.

Jamal Khashoggi entered the consulate of Saudi Arabia consulate in Istambul. He exited the Saudi consulate after a few minutes. The team wanted to arrest Khashoggi but botched it. He never exited the Saudi consulate but died there. Khashoggi, according to the reporter, was seen on a flight leaving Turkey for Estonia. A team flew from Saudi Arabia to Turkey prior to Khashoggi's appointment at the consulate specifically to intercept him. The team was sent by the Saudi crown prince with the order to murder Khashoggi. Jamal A. Khashoggi works for The Washington Post, and is the editor-in-chief of Al-Arab News.

Figure 1: An example for query-based scenario construction. Given the **query**, we want to select event-denoting sentences from a document mixture to build a target scenario with a sequence of compatible events. The mixture also contains sentences which may be irrelevant or part of an alternative scenario.

1 Introduction

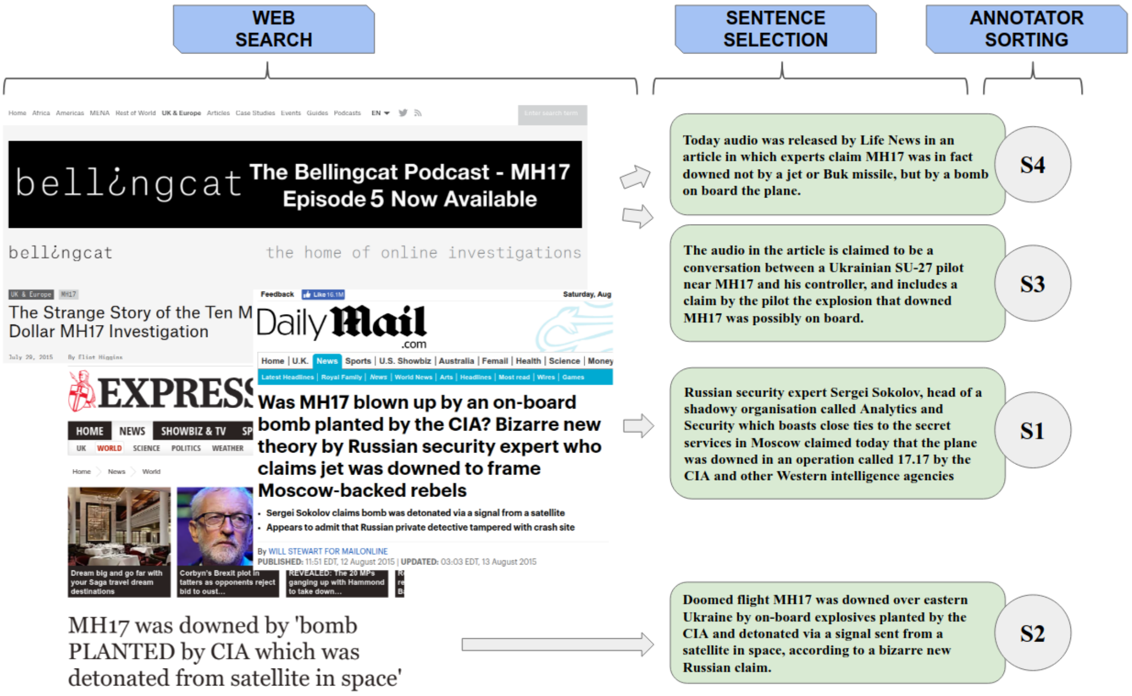
While a situation is developing, news reports often contain multiple contradictory stories (scenarios) of what happened, and it is hard to piece together the individual scenarios. For example, surrounding the disappearance of the Saudi journalist Jamal Khashoggi, there were initially multiple conflicting accounts of what happened. One states that he was the victim of a murder scheme; an alternative suggests that he walked out of the consulate alive. The task of identifying these individual scenarios is also being considered in the Active Interpretation of Disparate Alternatives (AIDA) program,[1] and in a recent Text Analysis Conference (TAC).[2]

We frame the task as query-based scenario discovery: given a *topic* (e.g., the disappearance of Jamal Khashoggi) and a *query* (e.g. *Jamal Khashoggi was murdered*), we want to retrieve a *scenario*, a set of compatible events, from the given reports. We formulate query-based scenario discovery as one-class clustering (Bekkerman and Crammer, 2008). We specifically focus on discovering a scenario of *compatible events* (Barzilay and Lapata, 2008; Chambers and Jurafsky, 2008, 2009; Mostafazadeh et al., 2017) in a collection of related and unrelated event-denoting sentences, which may contain conflicting and irrelevant information. We start with a query (see Figure 1) and then iteratively insert sentences into the "scenario-in-construction". Sentences are chosen based on overall compatibility as well as the ease with which scenario sentences can be arranged into an order. We additionally use an adapted relation network (Santoro et al., 2017) to assess connections between words.

For our evaluation, we collect a human-curated set of competing scenarios for real-world news topics. As collecting such data is costly, we follow past work in training our model on synthetic data consisting of document mixtures (Wang et al., 2018) and compare our models directly to theirs. We show that training on such synthetic data yields a model that can substantially outper-

[1] https://www.darpa.mil/program/active-interpretation-of-disparate-alternatives

[2] https://tac.nist.gov/2018/SM-KBP/index.html


WEB SEARCH
SENTENCE SELECTION
ANNOTATOR SORTING
bellingcat
The Bellingcat Podcast - MH17 Episode 5 Now Available
the home of online investigations
The Strange Story of the Ten M Dollar MH17 Investigation
Daily Mail .com
EXPRESS
S4
S3
S1
S2

Was MH17 blown up by an on-board bomb planted by the CIA? Bizarre new theory by Russian security expert who claims jet was downed to frame Moscow-backed rebels

- Sergei Sokolov claims bomb was detonated via a signal from a satellite
- Appears to admit that Russian private detective tampered with crash site

MH17 was downed by 'bomb PLANTED by CIA which was detonated from satellite in space'

Today audio was released by Life News in an article in which experts claim MH17 was in fact downed not by a jet or Buk missile, but by a bomb on board the plane.

The audio in the article is claimed to be a conversation between a Ukrainian SU-27 pilot near MH17 and his controller, and includes a claim by the pilot that the explosion that downed MH17 was possibly on board.

Russian security expert Sergei Sokolov, head of a shadowy organisation called Analytics and Security which boasts close ties to the secret services in Moscow claimed today that the plane was downed in an operation called 17.17 by the CIA and other Western intelligence agencies

Doomed flight MH17 was downed over eastern Ukraine by on-board explosives planted by the CIA and detonated via a signal sent from a satellite in space, according to a bizarre new Russian claim.

Figure 2: Phase 2 (cf. Table 1) of generating human evaluation data Human100: given the topic *Why did MH-17 crash?* and the scenario *MH-17 had a bomb on board*. The annotator searches the web and finds the webpages above. From these pages, she first selects 4 sentences which are relevant, then sorts them so that they make a consistent scenario that could be read from start to finish.

form lexical baselines and the strong neural model proposed in Wang et al.'s seminal work.

Our contribution is three-fold: (1) A query-based scenario construction task, for which we introduce a model to iteratively build a scenario with compatible events, exploiting ordering. (2) A human-curated evaluation set consisting of multiple accounts of real-world new events, along with a collection of scalably-built synthetic simulation datasets, which we show serve as an effective source of training data. (3) Comprehensive experiments and analysis that cast light on the properties of the task and data, as well as on the challenges.

2 Background

Our work traces its roots to research in *script* (Schank and Abelson, 1977; Mooney and DeJong, 1985) and *narrative schema* learning (Chambers and Jurafsky, 2008; Pichotta and Mooney, 2016). Early work explored tasks such as script modeling (Mooney and DeJong, 1985). Recent work built on the idea that compatibility of events can be learned from corpus data, evaluated on narrative cloze (Chambers and Jurafsky, 2009) and predicting-next-events (Pichotta and Mooney, 2016; Mostafazadeh et al., 2017).

We introduce a task with a more practical objective in mind: given a query or an information cue, extract the rest of the pieces to build a compatible scenario. The task is related to conversation disentanglement of multiple entangled conversations in a dialogue transcript (Elsner and Charniak, 2008, 2011; Jiang et al., 2018; Kummerfeld et al., 2019), and more closely to narrative clustering (Wang et al., 2018), i.e. identifying all the scenarios in an information source by grouping relevant sentences/events. Unlike Wang et al. (2018), we do not attempt to identify all the scenarios in the source, but are guided by one particular user's information need (e.g. the scenario about Khashoggi's murder, as opposed to all the theories regarding his disappearance, like in Wang et al. (2018)). Further, we do not assume the number of scenarios is known a priori (as Wang et al. (2018) do).

We phrase query-based scenario construction as one-class clustering. One-class models assume the availability of data for only one (positive) class, rather than positive and negative data. In particular, one-class clustering assumes that the data consists of few positive cases among a large group of outliers (Bekkerman and Crammer, 2008; Banu

Topic question: ***Why did MH-17 crash?***
scenario 1. ***MH-17 had a bomb on board***
(1) Doomed flight MH17 was downed over eastern Ukraine by on-board explosives planted by the CIA and detonated via a signal sent from a satellite in space, according to a bizarre new Russian claim. (2) Russian security expert Sergei Sokolov head of a shadowy organisation called Analytics and Security which boasts close ties to the secret services in Moscow claimed today that the plane was downed in an operation called 17.17 by the CIA and other Western intelligence agencies. (3) Today audio was released by Life News in an article in which experts claim MH17 was in fact downed not by a jet or Buk missile, but by a bomb on board the plane. (4) The audio in the article is claimed to be a conversation between a Ukrainian SU-27 pilot near MH17 and his controller, and includes a claim by the pilot the explosion that downed MH17 was possibly on board.
scenario 2. ***MH-17 was shot down by missiles.***
(1) The shoot-down occurred in the War in Donbass, during the Battle of Shakhtarsk, in an area controlled by pro-Russian rebels. (2) The responsibility for investigation was delegated to the Dutch Safety Board (DSB) and the Dutch-led joint investigation team (JIT), who concluded that the airliner was downed by a Buk surface-to-air missile launched from pro-Russian separatist-controlled territory in Ukraine. (3) According to the JIT, the Buk that was used originated from the 53rd Anti-Aircraft Missile Brigade of the Russian Federation, and had been transported from Russia on the day of the crash, fired from a field in a rebel-controlled area, and the launcher returned to Russia after it was used to shoot down MH17. (4) Previously, the investigative website Bellingcat has pointed to involvement of the same brigade using open-source information.

Table 1: An example from the crowdsourced Human100 dataset. Phase 1 (in bold): an MTurk worker writes (by web searching and editing) a topic question and two scenarios that answers the question. Phase 2: an annotator elaborates on the two scenarios (also through search-and-edit) with a compatible scenario.

and Karthikeyan, 2014). Finally, our task is superficially similar to query-based summarization (Otterbacher et al., 2005; Baumel et al., 2018) but has a different goal: we want to distinguish potentially conflicting narrative scenarios rather than conduct single-topic information compression. We also distinguish our work from multi-document summarization (McKeown et al., 2002; Radev et al., 2005), as we are explicitly drawing distinctions among conflictive scenarios rather than summarizing the entire (single) topic.

3 Data

This section first introduces our human-curated, realistic evaluation data for our objective. Then we describe how we synthesized various types of training data for our model.

3.1 Human-curated Data

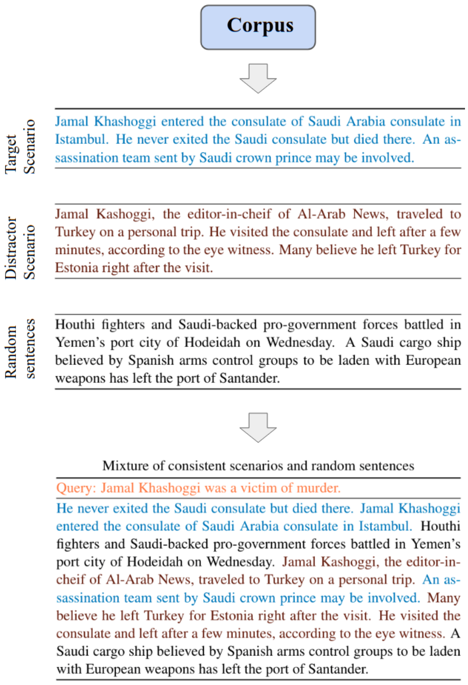
Realistic data on this task is hard to obtain, as after a time, a single scenario tends to dominate in the news. The Linguistics Data Consortium (LDC) has data for the AIDA project, and the 2018 Text Analysis Conference (TAC) had a hypothesis generation task, but both use a single topic only (the Russia-Ukraine conflict of 2014), with no hypotheses available (TAC) or no hypotheses at sentence level (LDC). As a step in the direction of realistic data for the task, we had human annotators collect news items that have multiple scenarios of what happened around the same topic (Table 1).

We collected data in two phases: in phase 1, we asked workers on Amazon Mechanical Turk (MTurk) to provide (1) a topic in the form of an English wh-question; (2) two scenarios that answer the topic question mutually exclusively. See the bold text in Table 1 for an example. In phase 2 (Figure 2), a group of non-Turk annotators[3] pieced together English sentences from the web to elaborate on the scenarios from phase 1 (Table 1, non-bold). Annotators were instructed to build scenarios which can be read fluently. Sentences that could be copy-and-pasted directly from web search results were prioritized. When such sentences were unattainable, annotators were allowed to edit the style of the text to fit the scenario in the tone of a news report. We allowed this relaxation because, as mentioned above, often one scenario is dominant in web search results. For instance, in the Jamal Khashoggi story, English media almost unanimously report that he was a victim of murder, whereas the alternative scenario – he "disappeared and is still alive" is harder to find. A set of 100 mixtures were collected this way. We refer to the dataset as **Human100** (stats in Table 2).[4]

Wang et al. (2018) gauge the difficulty of a mixture by measuring the *topic similarity* between the target and the distractor(s) using the cosine distance between the average word embeddings. A mixture of documents (or scenarios) will typically be more difficult to separate if the scenarios/documents are more topically similar, since lexical cues are less reliable in this case. Specif-

[3] In our pilots for phase 2, the data Turkers created were not ideal, therefore we opted for hiring local annotators which produced higher quality results.

[4] Available before main conference: http://www.katrinerk.com/home/software-and-data/query-focused-scenario-construction



Jamal Khashoggi entered the consulate of Saudi Arabia consulate in Istambul. He never exited the Saudi consulate but died there. An assassination team sent by Saudi crown prince may be involved.

Jamal Kashoggi, the editor-in-cheif of Al-Arab News, traveled to Turkey on a personal trip. He visited the consulate and left after a few minutes, according to the eye witness. Many believe he left Turkey for Estonia right after the visit.

Houthi fighters and Saudi-backed pro-government forces battled in Yemen's port city of Hodeidah on Wednesday. A Saudi cargo ship believed by Spanish arms control groups to be laden with European weapons has left the port of Santander.

Query: Jamal Khashoggi was a victim of murder.

He never exited the Saudi consulate but died there. Jamal Khashoggi entered the consulate of Saudi Arabia consulate in Istambul. Houthi fighters and Saudi-backed pro-government forces battled in Yemen's port city of Hodeidah on Wednesday. Jamal Kashoggi, the editor-in-cheif of Al-Arab News, traveled to Turkey on a personal trip. An assassination team sent by Saudi crown prince may be involved. Many believe he left Turkey for Estonia right after the visit. He visited the consulate and left after a few minutes, according to the eye witness. A Saudi cargo ship believed by Spanish arms control groups to be laden with European weapons has left the port of Santander.

Figure 3: Synthetic data generation illustrated with our running example. First randomly sample news paragraphs (one is randomly assigned as the target scenario, the rest are distractors) and random sentences, then mix the target scenario with (a) other complete distractor scenarios (red) and (b) random sentences.

Corpus	#scenarios	Vocab	Words/scenario	Sents/scenario
Human100	200 (2/topic)	-	127.9	4
NYT	1.14m	50,000	189.5	9.5
ROC	113k	39,954	46.5	5.0

Table 2: Corpora statistics. Top: human-curated data; Bottom: synthetic data. For NYT, we truncated the vocabulary to the most frequent 50k (the full vocabulary is over 100k). For NYT and ROC we apply a 85%/5%/10% split to construct train/dev/test sets. The two datasets share 27,795 words in vocabulary.

ically, Wang et al. found that their models struggled to separate scenarios even at 0.6 topical similarity, with accuracy on a binary clustering task dropping from 85% (all) to 68% (the hard ones). By this criterion Human100, at 0.8 average topic similarity cosine, is a hard dataset that challenges NLP models for their abilities to perform beyond shallow textual inference. Human-level performance is nonetheless fairly strong: 0.81 with distractor scenarios, and 0.97 with randomly sampled sentences (both are F1 scores, more details in Section 5).

3.2 Training with Proxy Synthetic Data

The idea we follow for synthetic data creation is the same as in Wang et al. (2018): we can use different articles as proxies for different scenarios, even though articles in the synthetic setting do not necessarily belong to the same topic. *Our hypothesis is that a model trained to predict which sentences came from the same news article will also do better at predicting which sentences come from the same scenario in the human-curated data.*

We synthesize data from two source corpora: the New York Times portion of English Gigaword (Graff et al., 2003) (**NYT**), which Wang et al. (2018) used to construct their document mixtures, and ROCStories (Mostafazadeh et al., 2017) (**ROC**). NYT is chosen for having the same domain as the human-constructed news data.[5] With ROC we want to gauge the generality of our approach out-of-domain: on news-only data a model could "cheat" by memorizing common named entities. We want to see to what extent models go beyond that to learn general event compatibility.

The synthesis method is summarized in Figure 3, with corpus statistics in Table 2. In the first condition we mix a randomly sampled target scenario with a distractor scenario (also randomly sampled), following Wang et al. (2018). The mixtures are denoted **NYT/ROC-w18**. For the second condition, we replace the distractor scenario with unconnected randomly sampled sentences (corpus-wide), hence **NYT/ROC-rand**. We also combine both conditions, giving a mixture with both types of distractors. We also vary the number of distractor scenarios in a mixture (2, 3, or, 4, including the target scenario). To equalize the number of sentences in mixtures, we pad them all to a fixed number of sentences. We call these **NYT/ROC-2/3/4**.

4 Models

Given a query q and a mixture of sentences, we want to select sentences that form a compatible scenario with the query. Our models select the sentences iteratively: the process begins with a *target set* $\mathcal{T}^{(1)} = \{q\}$ (i.e. initialized with only the query in the set) and a *candidate set* $\mathcal{C}^{(1)}$ (i.e.

[5] The Human100 dataset is created based on search results, which could conceivably be from a variety of domains, but annotators are largely selecting news articles about given topics. The topics themselves are general news with a skew towards politics, which is reflected in the NYT dataset as well. So both datasets consist mostly of political newswire writing, which we view as similar domain.

the mixture), and terminates at some time step i with a predicted scenario $\mathcal{T}^{(i)}$.

We experiment with two termination conditions: (1) **fixed #sentences**: a pre-specified number of sentences are extracted; (2) **dynamic #sentences**: a special end-of-scenario token `<end>` is predicted as the next candidate. (1) simulates the case where the user desires to specify the amount of information to be extracted (i.e. a consistent yet not necessarily all-inclusive scenario), and (2) the case where the model finds a complete scenario.[6]

Notation We describe one step of candidate selection without loss of generality, thus whenever no confusion arises, we drop the timestep superscripts to use $\mathcal{T}, \mathcal{C}$ for simplification. $t \in \mathcal{T}$ denotes a sentence in the target scenario, and $c \in \mathcal{C}$ a candidate. We use bold lower case letters for embeddings. The acronyms for the models are introduced at the beginning of model description (e.g. COMP for compatibility-attention).

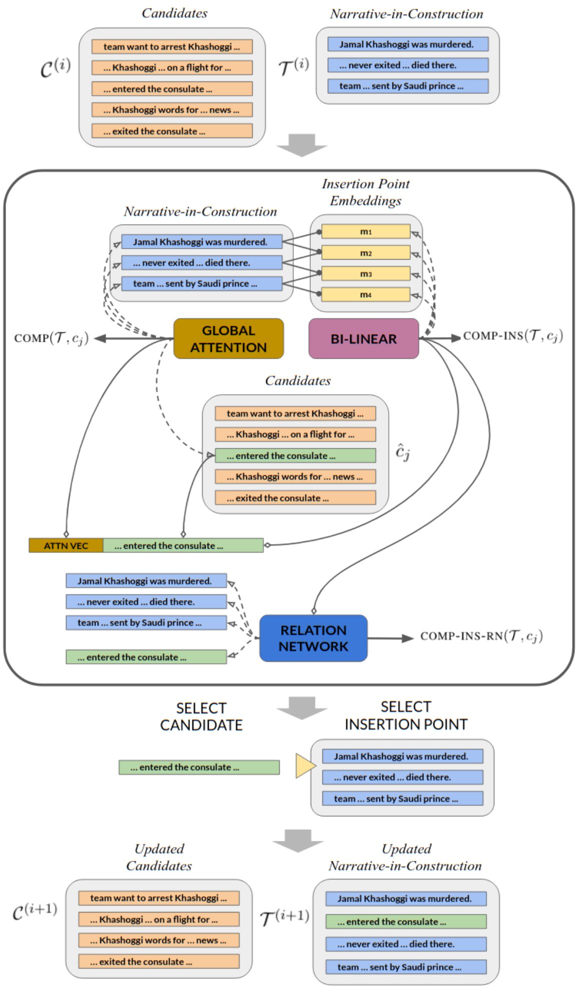
4.1 Architectures

Compatibility-Attention (COMP) In scenario building, intuitively we select a candidate c_j that fits best with $\mathcal{T}$ such that the updated target scenario is $\mathcal{T} \cup \{c_j\}$, the most compatible scenario-in-construction possible. The prediction $\hat{c}_j$ is then:

$$\hat{c}_j = \underset{c_j \in \mathcal{C}}{\text{argmax}}\ \text{COMP}(\mathcal{T}, c_j) \qquad (1)$$

For example, take the example in Figure 1, *He never exited the consulate but died there* is a good candidate, as it clearly relates to the scenario that Khashoggi was murdered, compatible with the current scenario-in-construction $\mathcal{T}^{(1)} = \{$*Jamal Khashoggi was murdered*$\}$.

Now note for $\mathcal{T}^{(i)}$, where $i > 1$ (i.e. multiple sentences in the scenario-in-construction), its member sentences do not contribute equally to the decision on a $\hat{c}_j$. For example, say $\mathcal{T}^{(2)} = \{$*Jamal Khashoggi was murdered*; *A team flew from Saudi Arabia ... to intercept him*$\}$, the first sentence is more informative for us to pick out *He never exited the consulate but died there* as a good candidate. We implement this with a bi-linear attention



Candidates
Narrative-in-Construction
$\mathcal{C}^{(i)}$
$\mathcal{T}^{(i)}$
team want to arrest Khashoggi ...
... Khashoggi ... on a flight for ...
... entered the consulate ...
... Khashoggi words for ... news ...
... exited the consulate ...
Jamal Khashoggi was murdered.
... never exited ... died there.
team ... sent by Saudi prince ...
Insertion Point Embeddings
m1
m2
m3
m4
GLOBAL ATTENTION
BI-LINEAR
COMP($\mathcal{T}, c_j$)
COMP-INS($\mathcal{T}, c_j$)
$\hat{c}_j$
ATTN VEC
RELATION NETWORK
COMP-INS-RN($\mathcal{T}, c_j$)
SELECT CANDIDATE
SELECT INSERTION POINT
Updated Candidates
Updated Narrative-in-Construction
$\mathcal{C}^{(i+1)}$
$\mathcal{T}^{(i+1)}$

Figure 4: Iterative models for one step of candidate selection. Embeddings of scenario sentences in $\mathcal{T}$ are shown in light blue, candidate embeddings in $\mathcal{C}$ in orange, and the sentence embedding being processed and selected is in light green. COMP-ATT: only runs GLOBAL ATTENTION to produce COMP($\mathcal{T}, c_j$), which selects a candidate *but not an insertion point*. COMP-INS runs GLOBAL ATTENTION and BI-LINEAR to get COMP-INS($\mathcal{T}, c_j$), which selects a candidate as well as an insertion point. COMP-INS-RN additionally employs a RELATION NETWORK on top of COMP-INS.

layer with parameters U:

$$\alpha_{j,k} = \text{softmax}_k(\boldsymbol{c}_j^\intercal U \boldsymbol{t}_k)$$

$$\text{COMP}(\mathcal{T}, c_j) = \text{softmax}_j\left(\text{linear}\left(\sum_k \alpha_{j,k} \boldsymbol{t}_k\right)\right)$$

i.e. c_j attends to the sentences $\{t_k\}$ in the current scenario, which computes a vector that is scored to compute the weight score of c_j under COMP.

Insertion-sort based selection (COMP-INS) If $\mathcal{T}$ is an ordered scenario, it is possible to further

[6] We select the same number of sentences as in the target scenario for the fixed #sents condition. This is extra supervision compared to the dynamic #sents condition. Changing this number changes the precision/recall tradeoff, but it would not lead to significant gains in the F1 values we report.

improve the quality of the candidate selection by selecting a c_j that is *easy-to-insert* into $\mathcal{T}$. For instance, for the example in Figure 1, the most readable update can be made by inserting *He never exited the consulate but died there* to the right of *Jamal Khashoggi was murdered*. Essentially, we imagine an insertion-sort based candidate selection technique: we iteratively pick out the easiest candidate to insert and maintain $\mathcal{T}$ as ordered. Crucially, note that we want to use ordering to aid clustering, rather than aiming for ordered scenarios: the model performance is only evaluated on clustering the correct set of sentences.

Let m_k be the insertion point to the left of the sentence t_k. For $\mathcal{T}$ we have insertion points $\{m_1, \cdots, m_{|\mathcal{T}|+1}\}$. For each $\langle$insertion point, candidate$\rangle$ tuple $\langle m_k, c_j\rangle$, we want to compute a weight $z_{k \leftarrow j}$ to indicate the "easy-to-insert-ness" of c_j to insertion point m_k. First we embed m_k and c_j: $\boldsymbol{c}_j$ is embedded with a BiLSTM, and $\boldsymbol{m}_k$ is computed from the embedded sentences in $\mathcal{T}$:

$$\boldsymbol{m}_k = \begin{cases} \boldsymbol{t}_1 & \text{if } k = 1 \\ \boldsymbol{t}_{|\mathcal{T}|} & \text{if } k = |\mathcal{T}| + 1 \\ \text{average}(\boldsymbol{t}_{k-1}, \boldsymbol{t}_k) & \text{otherwise} \end{cases} \tag{2}$$

Finally, applying a bilinear function:

$$z_{k \leftarrow j} = \boldsymbol{m}_k^{\intercal} W [\boldsymbol{a}_j; \boldsymbol{c}_j] \tag{3}$$

where $[;]$ is concatenation. This results in a model $\text{COMP-INS}(\mathcal{T}, c_j)$ s.t.

$$\hat{c}_j = \underset{c_j \in \mathcal{C}}{\text{argmax}}\ \text{COMP-INS}(\mathcal{T}, c_j) \tag{4}$$

$$\text{COMP-INS}(\mathcal{T}, c_j) = \max_k (z_{k \leftarrow j}) \tag{5}$$

i.e. the easy-to-insert-ness of c_j is indicated with its highest z score over all the available insertion points: the larger the largest $z_{k \leftarrow j}$ is, the clearer the model is about where to insert c_j.

Relation Networks (COMP-INS-RN) Oftentimes word tokens in $\mathcal{T}$ and $\mathcal{C}$ are also indicative of which candidate is the best. E.g. in Figure 1, the candidate *He never exited the consulate but died there* has the event *died* which relates to the *murder* in the query. Similarly the entity *he* is likely anaphoric to *Jamal Khashoggi*. The relations make the sentence an ideal candidate.[7]

Adapting the *Relation Network* as per Santoro et al. (2017),[8] we summarize the relation between the word tokens in c_j and $\mathcal{T}$ with a summary score r_j, i.e. how strongly the candidate is connected to the scenario-in-construction, based on the relations between its tokens and that of the scenario-in-construction. The process breaks down to three steps: first, we pair c_j with each of the sentences $t_k \in \mathcal{T}$ and compute a sentence-sentence relational vector $\boldsymbol{v}_{j,k}$ which is the sum over all the word-word bi-linear contraction (the bi-linear contraction of two word embeddings $\boldsymbol{w}_a, \boldsymbol{w}_b$ is $\boldsymbol{w}_a^{\intercal} V \boldsymbol{w}_b$, where $\boldsymbol{w} \in \mathbb{R}^d, V \in \mathbb{R}^{d \times l \times d}$). Then, we average over the sentence-sentence relation vectors to obtain a summary vector $\boldsymbol{p}_j = \underset{k}{\text{average}}(\boldsymbol{v}_{j,k})$. Finally we compute $r_j = \text{linear}(\boldsymbol{p}_j)$. We incorporate the relation network with COMP-INS by adding r_j to all the $z_{k \leftarrow j}$ over k. Patterning after the model descriptions above, we get a model $\text{COMP-INS-RN}(\mathcal{T}, c_j)$:

$$\hat{c}_j = \underset{c_j \in \mathcal{C}}{\text{argmax}}\ \text{COMP-INS-RN}(\mathcal{T}, c_j) \tag{6}$$

$$\text{COMP-INS-RN}(\mathcal{T}, c_j) = \max_k (z_{k \leftarrow j} + r_j) \tag{7}$$

4.2 Optimization

Since our models' supervision comes in the form of whole scenarios, supervising our iterative insertion clustering model is not completely straightforward. In particular, there may be multiple correct sentences that can be added to the current scenario. For example, in Figure 1, all the sentences labeled with (B) are possible candidates. We thus optimize the marginal log-likelihood of making any correct decision.[9]

Specifically, we minimize: $\mathcal{L} = -\log \sum_j p(c_j)$, which maximizes the sum of the probabilities of the correct candidates. For Eq. (5, 7), we optimize for

$$\mathcal{L} = -\log \sum_{(k,j)} p(\langle m_k, c_j \rangle) \tag{8}$$

i.e. maximizing the probability sum over all correct (insertion point, candidate) pairs.[10]

[7] Note that Wang et al. (2018) apply a mutual attention mechanism (Hermann et al., 2015) for the similar modeling purpose, but with many more parameters than our relation net. For practicality we believe the latter is a better option.

[8] Santoro et al. (2017) abstract word tokens as *objects* to summarize the relation between two sentences (or, in general, sequences) with a compound function $f(\sum_{i,j} g(o_{1,i}, o_{2,j}))$, where $o_{1,i}$ is the i-th token in sentence 1 (similar for $o_{2,j}$), and f, g can be any model (e.g. a feedforward net).

[9] E.g. Durrett and Klein (2013) employ the same technique to optimize for multiple viable coreference candidates.

[10] Maximizing marginal likelihood has the attractive property that it allows our model to learn the "easiest" way to build the desired gold scenarios, rather than forcing one analysis.

At train time, we treat each timestep of scenario construction as a training example. We use a form of teacher forcing where we assume that a correct partial scenario has been built and we want to make some correct decision from there. The partial scenarios are built by adding *random* correct sentences, so the model learns to make correct decisions in a range of settings.

5 Experiments

We use scalably synthesized data for training and reserve the realistic data, which is expensive to produce, for evaluation. To better understand which techniques work, we also conduct evaluation on the synthetic data.

Data Preparation The statistics of our source corpora are summarized in Table 2. For all the synthetic datasets, we make 100k mixtures for training, 5k for validation and 10k for test, mixed from a 0.85/0.05/0.10 split of the total 113k scenarios. For NYT/ROC-w18, to properly compare with Wang et al. (2018), all mixtures have 2 scenarios. For NYT/ROC-2/3/4, we first mix 2/3/4 randomly sampled scenarios, then pad the resulting mixtures to a fixed lengths. For NYT/ROC-rand, we sample equal #sentences as for NYT/ROC-2/3/4.

Baselines We apply three baselines: (1) **UNIF**, which randomly selects $n-1$ candidates ($n-1$: #sentences in the target scenario minus the query). (2) **AVG**, an iterative model that always selects the candidate the embedding of which is the closest (in cosine) to the average embedding of the sentences in the scenario-in-construction. A sentence embedding is the average over each of its word embeddings. (3) **PAIRWISE**, which adapts Wang et al. (2018)'s best model. It predicts a probability for a pair of sentences to indicate how likely they are from the same scenario. PAIRWISE replaces the cosine in AVG with the pairwise model.

Implementation All the models are constructed with PyTorch 0.4.1 (Paszke et al., 2017). We use pretrained 1024-dim ELMo embeddings (Peters et al., 2018). The hidden size for the BiLSTM and the relation network are 200. We run 10 epochs with an initial learning rate of 1e-4 (with Adam (Kingma and Ba, 2014)).

Evaluation For clustering performance, we use macro-averaged F1, comparing our recovered cluster for each query to the gold cluster. For sorting-clustering correlation, Spearman's Rho (ρ); and for sorting per se, Kendall's Tau (τ).[11]

Condition	Model	NYT-w18	NYT-rand	NYT-4
Fixed#sent	UNIF	0.45	0.18	0.18
	AVG	0.51	0.46	0.29
	PAIRWISE	0.68	0.64	0.53
	COMP	0.86	0.84	0.76
	COMP-INS	0.87	0.84	0.81
	COMP-INS-RN	**0.93**	**0.92**	**0.84**
Dyn. #sent	COMP	0.70	0.66	0.58
	COMP-INS	0.75	0.70	0.61
	COMP-INS-RN	**0.78**	**0.73**	**0.65**

Table 3: Intrinsic evaluation: F1 scores (testing) for models trained on different NYT mixtures in fixed and dynamic #sentences conditions. In the dynamic #sentences condition, the baselines no longer apply because they do not model a stopping condition.

5.1 Constructing Effective Training Mixtures

Which method for synthetic mixture creation leads to the best results on Human100? We first run our models in the three mixing conditions – scenario distractor only, random sentence distractor only, and both distractors (see Section 3.2), then evaluate both intrinsically and on the human-curated Human100. Here we only use our domain-proxy NYT-* datasets.

COMP-INS-RN	NYT-w18	NYT-rand	NYT-4
Fixed #sent	0.65	0.60	0.70
Dyn. #sent	0.60	0.56	0.62
Human benchmark		0.82	

Table 4: Which scenario mixture method is the best? F1 scores on the Human100 data of the COMP-INS-RN model trained in different mixing conditions. The most complex condition (NYT-4) gives the best results.

Examining the results in Table 3,[12] we observe the hybrid mixtures with both types of distractors are the most difficult, with substantially lower performance. But how does this translate into human evaluation? In Table 4, we evaluate the best-performing model (COMP-INS-RN, trained on NYT-* sets) on Human100. We find that harder training conditions (i.e., the hybrid mixing) give stronger results on Human100. Our initial conclusion is: the more challenging hybrid mixing serves

[11] More discussion on the ρ-τ mix is at the end of sec 5.2.

[12] As we test on examples with more mixtures (i.e., going from NYT/ROC-2 to NYT/ROC-4), test accuracy steadily decreases (0.93/0.86/0.84 for NYT-2/3/4, 0.92/0.91/0.90 for ROC), as is to be expected. To avoid cluttering we only report scores on *-4 data.

better as a training proxy to the realistic data. We also see in both Tables 3 and 4 that the dynamic #sent setting, where the model needs to decide when to stop adding events, is considerably more difficult throughout.

5.2 Do Our Modules All Contribute?

We additionally evaluate our proposed modules – insertion-sort based selection and relation nets – to see which contributes substantially in the intrinsic evaluation (Table 3). The COMP-INS module achieves a gain of 3 points of F1 on average over COMP, and COMP-INS-RN improves 4.5 F1 on average over COMP-INS. In addition we see a clear and large margin of the models over the baselines. To evaluate the modules on Human100, we use the models trained on the best hybrid mixtures (Section 5.1). The results are summarized in Table 5.

	COMP	COMP-INS	COMP-INS-RN
Fixed #sent	0.62	0.68	0.70
Dyn. #sent	0.53	0.61	0.62
Human benchmark		0.82	

Table 5: How do modeling modules contribute? F1 scores on Human 100 of different models with the best hybrid training mixtures (NYT-4).

Similar to the intrinsic evaluation, both modules improve performance across fixed and dynamic conditions. While in intrinsic evaluation, relation nets are the stronger contributor, insertion-sort based selection leads to a higher performance gain on Human100.

COMP-INS	sorting	corr. clustering
Fixed #sent	0.31	0.38
Dyn. #sent	0.30	0.40

Table 6: Sorting performance (τ) and its correlation with clustering accuracy (ρ)

While sorting performance in itself is not very high, it has a reasonable correlation with clustering performance (Table 6): following Cui et al. (2018), we use Kendall's τ to compute sorting performance (as correlation of predicted and gold ordering). We then calculate the correlation between sorting performance and model performance, using Spearman's ρ as the most widely used correlation measure in NLP.[13]

5.3 Do We Learn Compatibility that Generalizes?

As argued previously (Section 3.1), NYT contains plenty of shallow textual cues, meaning an expressive model can do well at the task doing bag-of-words clustering of the data rather than more sophisticated event compatibility inference.

COMP-INS-RN	ROC-w18	ROC-rand	ROC-4
Train-on-ROC (in-domain)	0.95	0.95	0.90
Train-on-NYT (out-domain)	0.85	0.87	0.81

Table 7: Generalization out-of-domain text: train on NYT-*/ROC-* and evaluate on ROC-* (fixed #sents).

The first question is: do the models generalize out-of-domain, particularly when textual cues are much fewer? We train our strongest COMP-INS-RN on NYT-* and evaluate on the corresponding ROC-* datasets (Table 7): in in-domain evaluation (i.e., train and test on ROC) our model produces excellent performance, and in out-of-domain evaluation (i.e., train on NYT test on ROC) it manages to keep up with fairly strong results. This indicates our model captures information beyond surface cues.

5.4 Final Model

From the domain-generalization test in Table 7, we see there is likely NYT-* sets do not subsume all the information in ROC-* sets. Exploiting all the data we have available, we combine ROC and NYT in a domain-joint training. The results in Table 8 show that in both fixed and dynamic #sent conditions, the model improves on the performance with single-domain training (Table 4, 5).

6 Analyzing Human-curated Data

To set up a human-level performance benchmark, we asked two additional annotators (they did not participate in the collection of Human100) to manually perform the same task as the models in fixed #sentences condition on a sample of 30 with randomly chosen query and target scenario. On average the F1 is 0.82 (one worker 0.81, the other 0.83). While even our best model (COMP-INS-RN) is inferior to the human-level performance, it draws quite close.

This however does not tell the whole story: qualitatively comparing the scenarios built by COMP-INS-RN vs. annotators, we observe human annotators tend to construct much more reasonable scenarios even when they include sentences

[13] Sorting and clustering performance are calculated one pair per instance. In computing τ we drop incorrectly extracted candidates as they do not have gold ordering with target sentences.

COMP-INS-RN	R&N-w18	R&N-rand	R&N-4	rand-scenario
Fixed #sents	0.70	0.61	0.74	0.90
Dynamic #sents	0.65	0.59	0.67	0.79
UNIF	0.50	0.50	0.50	0.42
AVG	0.49	0.50	0.51	0.50
PAIRWISE	0.56	0.60	0.56	0.64
Human benchmark		0.82		0.97

Table 8: **Left table**: F1 scores for Human100 evaluation with the best model (COMP-INS-RN) in fixed and dynamic #sentences conditions. The model is trained with three domain-joint training datasets. The model clears a sizable margin over the baselines, but falls short from human-level. **Right table**: F1 for COMP-INS-RN with *modified Human100*, i.e. the distractor scenario is now a random sample from NYT rather than one collected by an annotator. Human100 has a topic similarity of over 0.8 but the modified version only 0.45. This demonstrates *high* topic similarity is a strong contributor to the difficulty of Human100, which is true for both models and humans.

Target scenario: *Trump says Russia is the sole party to be blamed.*
(1) Trump says that Russia is the sole party to be blamed. (2) White House issued a statement which says Moscow is violating the Reagan-era agreement. (3) Secretary of State Mike Pompeo announced the decision to suspend the accord, declaring that countries must be held scenarioable when they break the rules. (4) "We can no longer be restricted by the treaty while Russia shamelessly violates it," Mr. Pompeo said.
Distractor scenario: *Russians accused the wrong doing on the part of the Trump administration*
(1) Russians accuse the wrong doing on the part of the Trump administration. (2) This is the latest step in the Trump administrations pattern of abandoning the diplomatic tools that have prevented nuclear war for 70 years. (3) Russia has also complained about the alleged lack of U.S. diplomacy. (4) Russian foreign minister Sergei Lavrov accused the U.S. of being obstinate. U.S. representatives arrived with a prepared position that was based on an ultimatum and centered on a demand for us to destroy this rocket, its launchers and all related equipment under US supervision.

Table 9: Humans make more reasonable mistakes (the query is underscored): the annotator selected sentence (2) as a part of the target scenario, which, while not part of the gold, does make a comperent scenario. The model however chose sentence (1), which is in direct contradiction with the target scenario.

are not from the gold scenario (Table 9). This indicates that models for the task could benefit from textual inference capabilities (Cases et al., 2017), or from deeper meaning representations.

Discussion The results on Human100 are largely in line with those on synthetic datasets, which indicates that results on synthetic data gives a reasonable estimate of results on more realistic data. While the performance on Human100 is lower overall, the findings are encouraging. Further, realistic cases of scenario discovery have the property that different scenarios for the same topic have a high vocabulary overlap. This can be seen in Human100. This property of the data penalizes shallow processing based models while encouraging learning deeper semantics.

One caveat about Human100 is that the dataset is still relatively small; a larger dataset would be useful to strengthen quantitative analysis. Also, variable-size scenarios would be more realistic for evaluating the more general case. Finally, we would like to improve the crowdsourcing technique in future work: in phase 1, some collected scenarios are not entirely in conflict with each other, for example two talking points Trump made in his State of the Union Address. An extra step where additional crowdworkers rate the compatibility of scenarios could be useful. In phase 2, we would like to have scenarios which exhibit more nuanced conflicting points that capture a wider range of cues that distinguish different scenarios.

7 Conclusion

Identifying an individual scenario in a blend of contradictory stories is a task that is targeted by the Active Interpretation of Disparate Alternatives (AIDA) program as well in a recent Text Analysis Conference (TAC). We address this task through query-based scenario construction, and find sizable performance improvements both from taking sentence order into account (INS) and from encoding connections between words (RN). Evaluating on a new human-curated dataset, we find that the synthetic training data serves as a reasonable proxy for the human-curated data.

Our current model sometimes gets misled by superficially similar sentences, and it will be an important future direction to move towards deeper reasoning for the task. In addition, we plan to create larger human-curated datasets with variable size scenarios.

Acknowledgments

This research was supported by the DARPA AIDA program under AFRL grant FA8750-18-2-0017. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, DoD or the US government. We acknowledge the Texas Advanced Computing Center and Chameleon (Keahey et al., 2019) for providing grid resources that contributed to these results. We are grateful to the anonymous reviewers for helpful discussions.

References

E. Afreen Banu and R. Karthikeyan. 2014. Text data linkage of different entities using OCCT one class clustering tree. In *Proceedings of IJCSIT*.

Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. In *CoRR:1801.07704*.

Ron Bekkerman and Koby Crammer. 2008. One-class Clustering in the Text Domain. In *Proceedings of EMNLP*, pages 41–50.

Ignacio Cases, Minh-Thang Luong, and Christopher Potts. 2017. On the effective use of pretraining for natural language inference. In *CoRR:1710.02076*.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *Proceedings of ACL*.

Nathanael Chambers and Daniel Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and Their Participants. In *Proceedings of ACL*.

Zhiyong Cui, Ruimin Ke, and Yinhai Wang. 2018. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In *Proceeings of IEEE*.

Greg Durrett and Dan Klein. 2013. Easy Victories and Uphill Battles in Coreference Resolution . In *Proceedings of NAACL*.

Micha Elsner and Eugene Charniak. 2008. You Talking to Me? A Corpus and Algorithm for Conversation Disentanglement. In *Proceedings of ACL*.

Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of ACL*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English Gigaword. *Linguistic Data Consortium, Philadelphia*, 4:1.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Proceedings of NeurIPS*.

Jyun-Yu Jiang, Francine Chen, Yang-Ying Chen, and Wei Wang. 2018. Learning to Disentangle Interleaved Conversational Threads with a Siamese Hierarchical Network and Similarity Ranking. In *Proceedings of NAACL*.

Kate Keahey, Pierre Riteau, Dan Stanzione, Tim Cockerill, Joe Mambretti, Paul Rad, and Paul Ruth. 2019. Chameleon: a scalable production testbed for computer science research. In Jeffrey Vetter, editor, *Contemporary High Performance Computing: From Petascale toward Exascale*, 1 edition, volume 3 of *Chapman Hall/CRC Computational Science*, chapter 5, pages 123–148. CRC Press, Boca Raton, FL.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: a Method for Stochastic Optimization. In *Proceedings of ICLR*.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A Large-Scale Corpus for Conversation Disentanglement. In *Proceedings of ACL*.

Kathleen R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with Columbias Newsblaster. In *Proceedings of NAACL*.

Raymond J. Mooney and Gerald DeJong. 1985. Learning Schemata for Natural Language Processing. In *Proceedings of IJCAI*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James F. Allen. 2017. LSDSem 2017 Shared Task: The Story Cloze Test. In *LSDSem 2017 Shared Task*.

Jahna Otterbacher, Günes Erkan, and Dragomir R. Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of EMNLP*.

Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. 2017. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL*.

Karl Pichotta and Raymond Mooney. 2016. Using Sentence-Level LSTM Language Models for Script Inference. In *Proeedings of ACL*.

Dragomir Radev, Jahna Otterbacher, Adam Winkel, and Sasha Blair-Goldensohn. 2005. NewsInEssence: Summarizing Online News Topics. In *Communications of the ACM*.

Adam Santoro, David Raposo, David G.T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A Simple Neural Network Module for Relational Reasoning. In *CoRR*.

Roger C. Schank and Robert P. Abelson. 1977. Scripts, Plans, Goals and Understanding. *Lawrence Erlbaum*.

Su Wang, Eric Holgate, Greg Durrett, and Katrin Erk. 2018. Picking Apart Story Salads. In *Proceedings of EMNLP*.