Picking Apart Story Salads

Su Wang\textsuperscript{1,2} Eric Holgate\textsuperscript{1} Greg Durrett\textsuperscript{3} Katrin Erk\textsuperscript{1}

\textsuperscript{1}Department of Linguistics
\textsuperscript{2}Department of Statistics and Data Science
\textsuperscript{3}Department of Computer Science
The University of Texas at Austin
shrekwang@utexas.edu holgate@utexas.edu
gdurrett@cs.utexas.edu katrin.erk@mail.utexas.edu

Abstract

During natural disasters and conflicts, information about what happened is often confusing, messy, and distributed across many sources. We would like to be able to automatically identify relevant information and assemble it into coherent narratives of what happened. To make this task accessible to neural models, we introduce \textit{Story Salads}, mixtures of multiple documents that can be generated at scale. By exploiting the Wikipedia hierarchy, we can generate salads that exhibit challenging inference problems. Story salads give rise to a novel, challenging clustering task, where the objective is to group sentences from the same narratives. We demonstrate that simple bag-of-words similarity clustering falls short on this task and that it is necessary to take into account global context and coherence.

1 Introduction

When a natural disaster strikes or a conflict arises, it is often hard to determine what happened. Information is messy and confusing, spread out over many messages, buried in irrelevant text, and even conflicting. For example, when flight MH-17 crashed in Ukraine in 2014, there were initially many theories of what happened, including a missile strike initiated by Russia-affiliated militants, a missile strike by the Ukrainian military, and a terrorist attack. There was no single coherent interpretation of what happened, but multiple, messy narratives, a \textit{story salad}. We would like to be able to automatically identify relevant information and assemble it into coherent narratives of what happened. This task is also the subject of an upcoming task at the Text Analysis Conference\textsuperscript{1}.

Picking apart a story salad is a hard task that could in principle make use of arbitrary amounts of inference. But it is also a task in which coherence judgments could play a large role, the simplest being topical coherence, but also narrative coherence (Chambers and Jurafsky, 2008, 2009; Pichotta and Mooney, 2016; Mostafazadeh et al., 2017), overall textual coherence (Barzilay and Lapata, 2008; Logeswaran et al., 2018), and coherence in the description of entities. This makes it an attractive task for neural models.

To make the task accessible to neural models, we propose a simple method for creating simulated story salad data at scale: we mix together sentences from different documents. Figure 1 shows an example mixture of two articles from Wikipedia, one on the Russia-Chechnya conflict and one on a conflict between the U.S. and Afghanistan. By controlling how similar the source documents are, we can flexibly adjust the difficulty of the task. In particular, while we do not focus on creating mixtures with conflicting information, it can often be found in mixtures created based on Wikipedia categories, as shown in Figure 2.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example_story_salad.png}
\caption{A story salad involving two articles, about a Russian military operation in Chechnya (A) and about a U.S. operation in Afghanistan (B). These two articles are topically similar but their mixture can still be disentangled based on narrative coherence.}
\end{figure}

\footnote{The story salad data is available at \url{http://www.katrinerk.com/home/software-and-data/picking-apart-story-salads-1}. The Wikipedia categories determine how related the two stories are.}

1\textsuperscript{https://tac.nist.gov/2018/SM-KBP/index.html}
We explore some initial models for our Story Salad task. As the aim of the task is to group story pieces into stories, we start with straightforward clustering based on topic similarity. But topic similarity is clearly not enough to group the right pieces together. For example, the two articles in Figure 1 are both about armed conflicts, but the Russia-Chechnya sentences in the example form a group in contrast to the U.S.-Afghanistan sentences. To model this, we learn sentence embeddings adapted to the clustering task and with access to global information about the salad at hand. We also test an extension where to decide whether to group two sentences together, the model mutually attends to the sentences during encoding in order to better focus on the commonalities and differences of these two sentences. Both extensions lead to better models (6-13% improvement in accuracy with a model incorporating both), confirming that the task requires more than just general topical similarity. But there is much room for improvement, in particular on salads generated to be more difficult, where performance is around 15 points lower than on arbitrary mixtures.

2 Related Work

Building on early work in script learning (Schank and Abelson, 1977), Chambers and Jurafsky (2008) introduce narrative schema and propose the “narrative cloze” task where the modeling objective is to predict the event happening next. The topic has since seen many extensions and variants coupled with increasingly sophisticated models (Chambers and Jurafsky, 2009) including neural networks (Granroth-Wilding and Clark, 2016; Pichotta and Mooney, 2016; Mostafazadeh et al., 2017). This line of work is related to story salads in that our aim of separating entangled narratives in a document mixture also leverages within-narrative coherence. Our work, however, is very different from narrative cloze: (i) we group sentences/events rather than predicting what happens next; (ii) crucially, the narrative coherence in story salads is in context, in that a narrative clustering is only meaningful with respect to a particular document mixture (see Section 5, 6), while in narrative cloze the next event is predicted on a “global” level.

Working with labeled story salad examples, we draw inspiration from previous work on supervised clustering (Bilenko et al., 2004; Finley and Joachim, 2005). We also take advantage of the recent success of deep learning in leveraging a continuous semantic space (Pennington et al., 2014; Kiros et al., 2015; Mekala et al., 2017; Wieting and Gimpel, 2017; Wieting et al., 2017) for word/sentence/event encoding; neural components for enhanced supervised clustering (Bilenko et al., 2004), in particular LSTMs (Hochreiter and Schmidhuber, 1997; Dai and Le, 2015), CNNs (Kim, 2014; Conneau et al., 2017), and attention mechanisms (Bahdanau et al., 2015; Hermann et al., 2015; Lin et al., 2017). By exploring our ability to pick apart story salads with these state-of-the-art NLP modeling tools, we attempt to (i) show the value of the story salad task as a new NLP task that warrants extensive research; (ii) understand the nature of the task and the challenges it sets forth for NLP research in general.

The task of picking apart story salads is related to the task of conversation disentanglement (Elsner and Charniak, 2008; Wang and Oard, 2009; Jiang et al., 2018), which is a clustering task of dividing a transcript into a set of distinct conversations. While superficially similar to our Story Salad task, conversation disentanglement focuses on dialogues and has many types of metadata available, such as time stamps, discourse information, and chat handles. Existing systems draw heavily on this metadata. Another related task is the distinction of on-topic and off-topic documents (Bekkerman and Crammer, 2008), which is defined in terms of topical relatedness. In comparison, the story salad task offers opportunities for more in-depth reasoning, as we show below.

3 Data

Natural story salads arise when multiple messy narratives exist to describe the same event or outcome. Often this is because each contribution to the explanation only addresses a small aspect of the larger picture. We can directly simulate the confusion this kind of discourse creates by taking multiple narratives, cutting them into small pieces, and mixing them together.
### Data generation

Story salads are generated by combining content from source documents and randomizing the sentence order of the resulting mixture. In order to ensure appropriately sized salads, we require that each source document contain at least eight sentences. Furthermore, to avoid problematically large salads, we pull paragraphs from source documents one at a time until the eight sentence minimum is met. While this procedure can be used to mix any number of documents, we currently present mixtures of two documents.

We utilize two different corpora as sources for story salad generation: (i) the subset of New York Times articles presented within English Gigaword (Graff and Cieri, 2003) and (ii) English Wikipedia (Wikipedia contributors, 2004). An overview of the datasets is available in Table 1.

### Gigaword

From the New York Times subset of Gigaword, we compiled a set of 573,681 mixtures we call NYT. Each mixture in this set is constructed from source articles pulled from the same month and year. Because this temporal constraint is the only restriction put on what articles can be mixed, it is possible for a salad to be constructed from topically disparate source documents (e.g., a restaurant review and a political story). We intend NYT to be relatively easy on the whole as a result of this design choice.

However, it is also possible for articles about dominant news stories and trends (e.g., the OJ Simpson trial in the summer of 1994) to be mixed as a result of the same temporal constraint. We therefore pulled out a curated subset of NYT consisting only of salads generated from highly topically similar source documents which we call NYT-HARD. This subset consists of the 1,000 salads where the source documents are most topically similar. We calculate topic similarity scores by computing the cosine similarity between the average word embeddings for each source document (denoted \( \cos \) hereafter)

\[
\cos(d) = \frac{g(\omega_1) \cdot g(\omega_2)}{\|g(\omega_1)\| \|g(\omega_2)\|}
\]

where \( \omega_1 \) and \( \omega_2 \) are the source documents, \( g \) is a function that computes the average word embedding of a document, and \( d \) is the salad under evaluation. The \( \cos \) scores on the test portion of the datasets are presented in Table 1.

### Wikipedia

From Wikipedia, we present an additional set of 500k salads constructed by combining random articles which we call WIKI.

We also leverage Wikipedia category membership as a form of human-annotated topic information. We use this to create a set of 50,374 salads, henceforth called WIKI-HARD, by restricting the domain of articles to only those appearing in categories containing the words conflict and war. Each mixture in this set is generated from source articles from the same category in order to produce highly difficult mixtures. We intend this to be a challenge set in this domain as the constituent articles for a given mixture are intentionally selected to be closely related. While we have used the category information to construct an intentionally very difficult set for this paper, we note that this procedure can be used to create sets of varying difficulty.

The fact that WIKI-HARD is generated from human-annotated category labels differentiates it from NYT-HARD in the source of its difficulty. After manually reviewing 20 samples from each WIKI-HARD dataset, we found that NYT-HARD more frequently contains salads that are impossible for humans to pick apart while WIKI-HARD more frequently contains salads that are possible, though challenging. In particular, in 9 out of 20 WIKI-HARD salads we found that access to world knowledge and inference would be beneficial. Nevertheless, the two WIKI-HARD datasets are both high in topic similarity (Table 1).

In Figure 2 we present sentences from a sample WIKI-HARD salad that can be solved with world knowledge. In this salad, we learn about two individuals. We can tell that Randle, born in 1855, is unlikely to also have been enrolled in high school in 1913 at the age of 58. We also learn that Randle was a doctor, while Martins, the other individual, was involved in theater. From this, we can deduce that the individual who “also worked as a wrestler” is more likely to be Martins than Randle.

### Table 1: Statistics of the datasets we present.

| Dataset | Salads | Total Words | \( \mu \) Words/Salad | \( \cos \) (test) |
|---------|--------|-------------|------------------------|-------------------|
| NYT     | 573,681| 217,841,716 | 379,726                | 0.33              |
| WIKI    | 1,000  | 20,149      | 438,220                | 0.56              |
| WIKI-HARD | 50,374| 21,266,243  | 422,167                | 0.64              |
| HARD    | 50,374 | 21,266,243  | 422,167                | 0.64              |
| WIKI-HARD | 1,000 | 20,149      | 438,220                | 0.56              |

Wikipedia dump pulled on January 20, 2018.

The average topic cosine similarity scores (\( \cos \)) between the two narratives in document mixtures are computed from the test sets. The NYT, WIKI and WIKI-HARD salads are divided into 80%/20% train/test splits, while the smaller HARD salads are divided into 90%/10% train/test splits. The cosine topic similarity scores (\( \cos \)) between the two narratives in document mixtures are computed from the test sets. The \( \cos \) scores on the test portion of the datasets are presented in Table 1.
Event Representation. Finally, we explore a form of document representation that has been shown to be useful in narrative schema learning, a related task. We include variants of NYT and NYTHARD with story salads consisting of event tuple representations instead of natural language sentence representations, as in Pichotta and Mooney (2016). We label these variants as NYT-EVENT and NYT-EVENT-HARD. Event tuples are in the form \(<\text{VERB}, \text{SUBJ}, \text{DOBJ}, \text{PREP}, \text{POBJ}>\), where as many preposition and prepositional object pairs as necessary are allowed.

Summary. The story salads we present here are, in the end, simpler than those that occur naturally in the news or on social media: for one thing, sentences drawn from a document written by a single author should exhibit a high degree of coherence. We have also shown that we can use Wikipedia category annotations to produce large-scale story salad datasets with customizable levels of difficulty, enabling us to increase the difficulty of the task as performance increases. In the following section, we see that both our standard and *-HARD mixtures are challenging for current models. Furthermore, our WIKI-HARD dataset contains salads featuring conflicting information and is an attractive setting for building models with deeper reasoning capabilities.

4 Models

We treat the story salad task as a narrative clustering task where, in our dataset, each salad is comprised of two clusters. Accordingly, the first baselines we consider are standard clustering approaches.

Baselines. Our first baseline is a simple uniform baseline (hereafter UNIF), where we assign all sentences in a document mixture to a single cluster. Under UNIF the clustering accuracy is the percentage of the majority-cluster sentences, e.g. if a mixture has 7 sentences from one narrative and 3 from the other, then the accuracy is 0.7.

Additionally, we explore a family of baselines that consist of clustering off-the-shelf sentence embeddings. We choose k-medoids (hereafter KM) as our clustering algorithm. For sentence embeddings, we experimented with (i) averaged 300D GloVe embeddings (Pennington et al., 2014), which have been shown to produce surprisingly strong performance in a variety of text classification tasks (Iyyer et al., 2015; Coates and Bolligala, 2018); (ii) skip-thought embeddings (Kiros et al., 2015); and (iii) SCDV (Mekala et al., 2017), a multisense-aware sentence embedding algorithm which builds upon pretrained GloVe embeddings using a Gaussian mixture model. Averaged GloVe embeddings gave the best performance in our experiments; to avoid clutter, we only report those results henceforth.

Neural supervised clustering. Our baselines work directly on sentence embeddings and therefore ignore the task-specific supervision available in our labeled training data. Inspired by the work in Bilenko et al. (2004) and Finley and Joachim (2005) on supervised clustering, we aim to exploit this supervision using a learned distance metric in our clustering.

Figure 3 shows our model, which produces a distribution \(P(\text{same} \mid s_1, s_2, d)\): the probability that two sentences \(s_1\) and \(s_2\) taken from document mixture \(d\) are in the same cluster. We train this model as a binary classifier on sampled pairs of sentences to distinguish same-narrative sentence pairs (positive examples) from different-narrative pairs (negative examples). \(1 - P(\text{same} \mid s_1, s_2, d)\) is then used by the clusterer as the pairwise distance metric. Given the pairwise distance between all sentence pairs in a mixture, the KM algorithm

---

1K-medoids is chosen as a substitute for k-means because the latter does not extend easily to our classifier-aided neural models: it does not work when only pairwise distances are available. In empirical evaluation we found k-means and k-medoids to produce very similar accuracy scores when using off-the-shelf embeddings. Experiments with hierarchical agglomerative clustering (not reported here) showed it to perform worse than either method.

2In early experiments, another strong candidate we tried is a joint model of a sentence autoencoder and a clustering algorithm (Yang et al., 2017). However, this produces subpar performance (weaker than the strongest baseline), due partially to scalability issues in learning these jointly.
Figure 3: BiLSTM sentence pair classifier to determine whether \( s_1 \) and \( s_2 \) are from the same narrative, augmented with a mutual attention and a context reader. The three subcomponents — the BiLSTM, the mutual attention mechanism, and the context reader — each produce vectors, denoted as \( z, m, c \) respectively. In the basic BiLSTM model, only \( z \) is fed to the bilinear layer (Eq. 2), while more sophisticated models incorporate the additional mutual attention and context vectors.

Our classifier is a neural network model built on top of LSTM sentence encoders, which perform well at similar text classification tasks (Dai and Le, 2015; Liu et al., 2016).\(^9\) Denoting a sentence as the list of embeddings of its constituent words: \( s = \{w_1, \ldots, w_M\} \), we first encode it as a sentence embedding \( z \) with a bidirectional LSTM \( z = \text{BiLSTM}(s) \) and then compute the probability score with a bilinear layer:

\[
P(\text{same} \mid s_1, s_2) = \sigma(z_1^T W z_2) \quad (2)
\]

This model corresponds to the green subset of Figure 3.

**Stronger models.** There are two additional effects we might want our model to capture. First, whether two sentences are from the same narrative cannot be determined globally: there aren’t two “globally-contrasted”\(^10\) narratives (or bag-of-words based topics) from which sentences are sampled. In other words, sentences are always (pairwise) compared in the context of the document mixture from which they are drawn. Second, we want to capture more in-depth interactions between sentences: our sentence embedding scheme for a sentence \( s_1 \) should exploit its point of comparison \( s_2 \) and encode \( s_1 \) with a view of similarities to and differences with \( s_2 \). This type of technique has been useful in tasks like natural language inference (NLI) (Bowman et al., 2015; Peters et al., 2018).

To improve contextualization, we add a CNN-based context encoder to the BiLSTM classifier: the reader embeds the whole document salad at hand into a vector. Formally, we compute \( c = \text{CNN}(d) \), where in this case CNN denotes a single convolution layer with max pooling in the style of Kim (2014) and \( d \) is the concatenation of all sentences in the mixture. This component is shown in blue in Figure 3. The context vector \( c \) is then appended to \( z \) and fed into the bilinear layer.

To capture the interaction between two sentences in a pair, we employ a mutual attention mechanism, which is similar to the attentive reader (Hermann et al., 2015). Let \( e_{i,1\ldots n} \) denote the BiLSTM outputs for the tokens of sentence \( i \). Given the encoding \( z_1 \) of sentence \( s_1 \), we compute attention weights and a representation of \( s_2 \) as follows:

\[
\alpha_{1\rightarrow 2} = \text{softmax}_j(z_1^T e_{2,j})
\]

\[
m_{1\rightarrow 2} = \sum_j \alpha_{1\rightarrow 2,j} e_{2,j}
\]

We compute \( m_{2\rightarrow 1} \) analogously. This process is shown in purple in Figure 3. The \( m \) vectors are used as additional inputs to the bilinear layer.

For comprehensive ablation, we experiment with four variants of neural classifiers: (i) BiLSTM;
STM alone (BILSTM); (ii) BiLSTM + mutual attention (BILSTM-MT); (iii) BiLSTM + context (BILSTM-CTX); and (iv) BiLSTM + mutual attention and context (BILSTM-MT-CTX).

**Event-based models.** For the event-based variants of the datasets, NYT-EVENT and NYT-EVENT-HARD, we build three models: (i) FFNN-BILSTM: we input a sentence as a sequence of embeddings rather than word embeddings as in BILSTM, where a feedforward layer maps the words in an event tuple to an event embedding; (ii) FFNN-BILSTM-MT-CTX: replacing the base BILSTM in (i) with our best model which is enhanced with mutual attention and contextualization; (iii) FFNN-BILSTM-MT-CTX-PRETRAIN: a variant of (ii) that is based on the event embedding pretraining method\(^{11}\) described in Weber et al. (2018), where events are encoded with a feedforward net (same as (i)) and trained with a word2Vec-like objective, encouraging events that co-occur in the same narrative to have more similar embeddings.

5 Experiments and Analysis

**Experimental setup.** To stave off sparsity, we impose a vocabulary cut by using only the 100k most frequent lemmas. To evaluate on NYT, NYT-EVENT, WIKI and WIKI-HARD, we sample 20k unique salads (from their respective test portions\(^{12}\)) to use for both the sentence and event versions of the experiments. For WIKI-HARD, the training combines the training portions of both WIKI and WIKI-HARD. For NYT-HARD, we train on the training portion of NYT and evaluate on NYT-HARD in full as a test set.

All the neural components are constructed with TensorFlow and use the same hyperparameters across variants: a 2-layer BiLSTM, learning rate 1e-5 with Adam (Kingma and Ba, 2014), dropout (Srivastava et al., 2014) rate 0.3 on all layers, and Xavier initialization (Glorot and Bengio, 2010). To create training pairs for the neural classifiers, we randomly sample sentence pairs balanced between same-narrative and different-narrative pairs. We train with a batch size of 32 and stop when an epoch yields less than 0.001% accuracy improvement on the validation set, which is 5% of mixtures sampled from the training data beforehand (the models are not trained on the validation sample). For KM we use the default configurations of off-the-shelf software.\(^{13}\)

**Evaluation.** We evaluate all models in terms of a clustering accuracy metric (hereafter CA), which is a simple extension from the conventional accuracy metric: we calculate the ratio of correctly clustered sentences in a document mixture, averaged over test mixtures. Given a document mixture \(d_i\), we call its component documents \(A\) and \(B\). Let \(\text{pred}\) be a function that does the clustering by mapping each sentence \(s_{i,n}\) of mixture \(d_i\) to either \(A\) or \(B\), and \(\text{true}_{AB}\) a function that returns the original pseudo-labels (i.e. \(\{A, B\}\)) as they are, and \(\text{true}_{BA}\) flips the pseudo-labels, i.e. \(A \rightarrow B\) and \(B \rightarrow A\). Then the clustering accuracy for document \(d_i\) by \(\text{pred}\) is

\[
\text{CA}(d_i, \text{pred}, \text{true}) = \max\{\text{CA}(d_i, \text{pred}, \text{true}_{AB}), \text{CA}(d_i, \text{pred}, \text{true}_{BA})\}
\]

\[
\text{CA}(d_i, \text{pred}, \text{true}) = \frac{1}{N_i} \sum_n \mathbb{1} [\text{pred}(s_{i,n}) = \text{true}(s_{i,n})]
\]

where \(s_{i,n}\) is the \(n\)-th sentence of mixture \(d_i\).

**Sentence based models.** First we evaluate the sentence based models. We first run the UNIF baseline on all our datasets, where we obtain near-50% clustering accuracy. This indicates that the data are all balanced in the number of sentences in the two narratives of the mixtures. We then run k-medoids (KM) on sentence embeddings as a baseline to compare to the classifier-aided models. The results are summarized in Table 2.

![Table 2: Clustering accuracy (CA) results from the sentence based experiments. More sophisticated models do better across all datasets, particularly on *-HARD tasks, which are substantially more challenging.](github.com/letiantian/kmedoids)

---

\(^{11}\)In Weber et al. (2018), a more complex tensor-based model is applied. Using exactly same method in our experiments we obtain weaker results.

\(^{12}\)Test sets available with data release.

\(^{13}\)github.com/letiantian/kmedoids
Figure 4: Attention weight heatmaps for a random sample with BILSTM-MT-CTX. Lighter color indicates higher attention weights. The two heatmaps in the same block are for the attention weights of one sentence attending to the other. In (a), we see related concepts being identified (video game and sega group), while in (b), we see a contrast (family station wagon and sega group).

Why would the mutual attention mechanism help? Plotting the attention weights of randomly selected samples, we see distributionally similar words being attended to in Figure 4a. Intuitively, a BiLSTM compresses a sentence into a single vector, leading to information loss (Conneau et al., 2018). Mutual attention enriches this representation by allowing us access to detailed information in sentences at word-level resolution by capturing lexical similarity. Even more interestingly, we observe a synergistic effect between mutual attention and contextualization: with the context reader added, we see high attention weights on words/phrases which bear little distributional similarity but are important for connecting/contrasting different narratives. For example, in Figure 4b, sega group and family station wagon are selected by the attention, despite not having similar words in the other sentences. These words are crucial in identifying the two narratives in this mixture: one is about a Japanese video game company, the other is on vehicle manufacturing in the U.S.

Another observation is that all models see drastic reduction in accuracy in the *-HARD version of the data. In fact, the clustering accuracy corresponds well with our topic similarity metric (cos, Eq. 1; Table 1) across models. In addition, cos is negatively correlated with clustering accuracy for all mixtures (Table 3).

From the results we also see that contextualization brings clear performance improvement. This supports our hypothesis that the Story Salad task is a nonstandard clustering task where the contrast of two narratives is only meaningful in the context of the particular mixture where they reside, rather than on a corpus-general level. Taking the example in Figure 1, the Russian-Chechnya and the U.S.-Afghanistan narratives are contrasted in that mixture, but one can easily imagine a mixture where they are in the same narrative and are contrasted to another narrative on business affairs. Further, contextualized models are less vulnerable to the performance reduction on mixtures with high topic similarity: for one thing, contextualization improves performance over the base BILSTM on both regular and *-HARD datasets. Secondly, computing the correlation between clustering accuracy and topic similarity, we see a lower negative correlation for contextualized models, true for both NYT and WIKI datasets (Table 3).

**Event based models.** While the accuracy scores in the event based experiments are in general lower than those in the sentence based (Table 4), overall we observe the same pattern that
**Table 4:** Clustering accuracy (CA) results from the event based experiments. *-MT-CTX is a short hand for FFNN-BILSTM-MT-CTX. The same notation applies for the following models.

| Model          | NYT-EVENT | NYT-EVENT-HARD |
|----------------|-----------|----------------|
| KM             | 64.7      | 55.3           |
| FFNN-BILSTM    | 64.9      | 54.8           |
| *-MT-CTX       | 66.8      | 59.1           |
| *-MT-CTX-PRETRAIN | 70.2    | 61.4           |

Figure 5: An example of (preprocessed) sentences from two unrelated documents being that have been clustered into a single cluster by the base model. Document (A) is an article about proposed privatization of public assets, while Document (B) is an article about happenings in Major League Baseball.

models not only have many fewer bad clusterings, they also show low accuracy almost exclusively in mixtures of related documents (2 cases of distinct documents for BILSTM-CTX, none for BILSTM-MT or BILSTM-MT-CTX). Figure 5 shows an example of a bad clustering of two unrelated documents, produced by the base BILSTM model.

In a second study, we rated the same 60 samples by their difficulty for a human, focusing in particular on mixtures that went from low performance (0.5-0.65) in the BILSTM model to high performance (0.8-1.0) in another model. For BILSTM-CTX we find that only 2 out of 11 mixtures with such marked improvement over BILSTM were hard for humans; for BILSTM-MT only 1 out of 9 markedly improved mixtures was hard for humans. But for BILSTM-MT-CTX, 8 out of 17 markedly improved mixtures were hard for humans, indicating that more sophisticated models do better not only on easy but also on hard cases.

In a third small study, we compare NYT-HARD and WIKI-HARD for their difficulty for humans, looking at 20 mixtures each. Here, very interestingly, we find more mixtures that are impossible for humans in NYT-HARD (10 cases, example in Figure 6) than WIKI-HARD (3 cases). This presents a clear discrepancy between difficulty for humans and difficulty for models: the models do better on NYT-HARD which is harder for us. While we would not want to draw strong conclusions from a small sample, this hints at possibilities of future work where world knowledge, which is likely to be orthogonal to the information picked up by the models, can be introduced to improve performance (e.g. Wang et al. (2018)).

Note that unlike many other NLP tasks where
We have presented a technique to generate Story Salads, mixtures of multiple narratives, at scale. We have demonstrated that the difficulty of these mixtures can be manipulated either based on document similarity or based on human-created document categories. This data gives rise to a challenging binary clustering task (but easily extended to $n$-ary), where the aim is to group sentences that come from the same original narrative. As coherence plays an important role in this task, the task is related to work on narrative schemas (Chambers and Jurafsky, 2008; Pichotta and Mooney, 2016) and textual coherence (Barzilay and Lapata, 2008; Logeswaran et al., 2018). The automated and scalable data generation technique allows for the use of neural models, which need large amounts of training data.

Conducting a series of preliminary experiments on the data with common unsupervised clustering algorithms (Cao and Yang, 2010) and variants of neural network-based (Kim, 2014; Dai and Le, 2015; Liu et al., 2016) supervised clustering (Bilenko et al., 2004; Finley and Joachim, 2005) models, we have (i) verified the validity of the task where generalizable patterns can be learned through machine learning techniques; (ii) shown that this is a nonstandard clustering task in which the contrast between narratives is in context as opposed to global; (iii) found that there is a class of mixtures that are doable for humans but very difficult for our current models, and that in particular the category-based method creates a high proportion of such mixtures.

Our work opens up a large number of directions for future research. First, while our models obtain strong results on simpler story salads, they have low performance on more difficult mixtures with high topical similarity. Second, there are many intuitively promising sources of information that we have not explored, such as coreference. And third, our models rely on pairwise similarity-based coherence learning, which leads to the natural question of whether structured prediction would improve performance.

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 for providing grid resources that contributed to these results. We are grateful to the anonymous reviewers for helpful discussions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of NIPS.
Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Embeddings. In Proceedings of EMNLP.

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 Proceedings of ACL.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of EMNLP.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a Simple Way to Prevent Neural Network from Overfitting. JMLR, 15:1929–1958.

Lidan Wang and Douglas W. Oard. 2009. Context-based Message Expansion for Disentanglement of Interleaved Text Conversation. In Proceedings of NAACL.

Su Wang, Greg Durrett, and Katrin Erk. 2018. Modeling Semantic Plausibility by Injecting World Knowledge. In Proceedings of NAACL.

Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Event Representations with Tensor-based Compositions. In Proceedings of AAAI.

John Wieting and Kevin Gimpel. 2017. Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings. In Proceedings of ACL.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext. In Proceedings of EMNLP.

Wikipedia contributors. 2004. Plagiarism — Wikipedia, the free encyclopedia. [Online; accessed 20-January-2018].

Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. 2017. Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. In Proceedings of ICML.