When and Why is Document-level Context Useful in Neural Machine Translation?

Yunsu Kim  Duc Thanh Tran  Hermann Ney
Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
{surname}@cs.rwth-aachen.de

Abstract

Document-level context has received lots of attention for compensating neural machine translation (NMT) of isolated sentences. However, recent advances in document-level NMT focus on sophisticated integration of the context, explaining its improvement with only a few selected examples or targeted test sets. We extensively quantify the causes of improvements by a document-level model in general test sets, clarifying the limit of the usefulness of document-level context in NMT. We show that most of the improvements are not interpretable as utilizing the context. We also show that a minimal encoding is sufficient for the context modeling and very long context is not helpful for NMT.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017) has been originally developed to work sentence by sentence. Recently, it has been claimed that sentence-level NMT generates document-level errors, e.g. wrong coreference of pronouns/articles or inconsistent translations throughout a document (Guillou et al., 2018; Läubli et al., 2018).

A lot of research addresses these problems by feeding surrounding context sentences as additional inputs to an NMT model. Modeling of the context is usually done with fully-fledged NMT encoders with extensions to consider complex relations between sentences (Bawden et al., 2018; Voita et al., 2018; Zhang et al., 2018; Miculicich et al., 2018; Maruf et al., 2019). Despite the high overhead in modeling, translation metric scores (e.g. BLEU) are often only marginally improved, leaving the evaluation to artificial tests targeted for pronoun resolution (Jean et al., 2017; Tiedemann and Scherrer, 2017; Bawden et al., 2018; Voita et al., 2018, 2019). Even if the metric score gets significantly better, the improvement is limited to specific datasets or explained with only a few examples (Tu et al., 2018; Maruf and Haffari, 2018; Kuang and Xiong, 2018; Cao and Xiong, 2018; Zhang et al., 2018; Maruf et al., 2019).

This paper systematically investigates when and why document-level context improves NMT, asking the following research questions:

- In general, how often is the context utilized in an interpretable way, e.g. coreference?
- Is there any other (non-linguistic) cause of improvements by document-level models?
- Which part of a context sentence is actually meaningful for the improvement?
- Is a long-range context, e.g. in ten consecutive sentences, still useful?
- How much modeling power is necessary for the improvements?

To answer these questions, we conduct an extensive qualitative analysis on non-targeted test sets. According to the analysis, we use only the important parts of the surrounding sentences to facilitate the integration of long-range contexts. We also compare different architectures for the context modeling and check sufficient model complexity for a significant improvement.

Our results show that the improvement in BLEU is mostly from a non-linguistic factor: regularization by reserving parameters for context inputs. We also verify that very long context is indeed not helpful for NMT, and a full encoder stack is not necessary for the improved performance.

2 Document-level NMT

In this section, we review the existing document-level approaches for NMT and describe our strategies to filter out uninteresting words in the context.
input. We illustrate with an example of including
one previous source sentence as the document-
level context, which can be easily generalized also
to other context inputs such as target hypotheses
(Agrawal et al., 2018; Bawden et al., 2018; Vo-
ita et al., 2019) or decoder states (Tu et al., 2018;
Maruf and Haffari, 2018; Miculicich et al., 2018).

For the notations, we denote a source sentence
by $f$ and its encoded representations by $H$. A sub-
script distinguishes the previous (pre) and current
(cur) sentences. $e_i$ indicates a target token to be
predicted at position $i$, and $e_i^{\bot}$ are already pre-
predicted tokens in previous positions. $Z$ denotes en-
coded representations of a partial target sequence.

2.1 Single-Encoder Approach

The simplest method to include context in NMT is
to just modify the input, i.e. concatenate surroun-
ding sentences to the current one and put the ex-
tended sentence in a normal sentence-to-sentence
model (Tiedemann and Scherrer, 2017; Agrawal
et al., 2018). A special token is inserted between
context and current sentences to mark sentence
boundaries (e.g. $\_\text{BREAK}_\_$.)

Figure 1 depicts this approach. Here, a single
encoder processes the context and current sen-
tences together as one long input. This requires
no change in the model architecture but worsens
a fundamental problem of NMT: translating long
inputs (Koehn and Knowles, 2017). Apart from the
data scarcity of a higher-dimensional input space,
it is difficult to optimize the attention component
to the long spans (Sukhbaatar et al., 2019).

2.2 Multi-Encoder Approach

Alternatively, multi-encoder approaches encode
each additional sentence separately. The model
learns representations solely of the context sen-
tences which are then integrated into the baseli-
ne model architecture. This tackles the integration
of additional sentences on the architecture level,
in contrast to the single-encoder approach. In the
following, we describe two methods of integrating
the encoded context sentences. The descriptions
below do not depend on specific types of context
encoding; one can use recurrent or self-attentive
encoders with a variable number of layers, or just
word embeddings without any hidden layers on
top of them (Section 3.1).

2.2.1 Integration Outside the Decoder

The first method combines encoder representation
of all input sentences before being fed to the de-
coder (Maruf and Haffari, 2018; Voita et al., 2018;
Miculicich et al., 2018; Zhang et al., 2018; Maruf
et al., 2019). It attends from the representations of
the current sentence ($H_{\text{cur}}$) to those of the previous
sentence ($H_{\text{pre}}$), yielding $\bar{H}$. Afterwards, a linear
interpolation with gating is applied:

$$g\bar{H} + (1-g)H_{\text{cur}}$$

where $g = \sigma(W_g [H; H_{\text{cur}}] + b_g)$ is gating acti-
vation and $W_g, b_g$ are learnable parameters. This
type of integration is depicted in Figure 2. By
using such a gating mechanism, the model is ca-
pable of learning how much additional context in-
formation shall be included.

Figure 1: Single-encoder approach.

Figure 2: Multi-encoder approach integrating context
outside the decoder.
2.2.2 Integration Inside the Decoder

Another method integrates the context inside the decoder; the partial target history $e_{i-1}^{t-1}$ is available during the integration. Here, using the (encoded) target history as a query, the decoder attends directly to the context representations. It also has the original attention to the current sentence. Depending on the order of these two attention components, this type of integration has two variants.

**Sequential Attentions** The first variant is stacking the two attention components, with the output of one component being the query of another (Tu et al., 2018; Zhang et al., 2018).

Figure 3 shows the case when the current sentence is attended by the decoder first, which is then used to attend to the context sentence. This refines the regular attention to the current source sentence with additional context information. The order of the attention components may be switched. To block signals of potentially unimportant context information, a gating mechanism can be employed between the regular and context attention outputs like Section 2.2.1.

Figure 3: Multi-encoder approach integrating context inside the decoder with sequential attentions.

For each category above, we have described a common architecture shared by previous works in that category. There are slight variations but they do not diverge much from our descriptions.

2.3 Filtering of Words in the Context

Document-level NMT inherently has heavy computations due to longer inputs and additional processing of context. However, intuitively, not all of the words in the context are actually useful in translating the current sentence. For instance, in most literature, the improvements from using document-level context are explained with coreference, which can be resolved with just nouns, articles, and the conjugated words affected by them.

Under the assumption that we do not need the whole context sentence in document-level NMT, we suggest to retain only the context words that are likely to be useful. This makes the training easier with a smaller input space and less memory requirement. Concretely, we filter out words in the context sentences according to pre-defined word lists or predicted linguistic tags:

- Remove stopwords using a pre-defined list
- Remove $n \in \mathbb{N}$ most frequent words
- Retain only named entities
- Retain only the words with specific parts-of-speech (POS) tags

The first method has the same motivation as Kuang et al. (2018) to ignore function words. The second method aims to keep infrequent words that

---

1. https://github.com/explosion/spaCy
in recent years, I correctly foresaw that, in the absence of stronger fiscal stimulus (which was not forthcoming in either Europe or the United States), recovery from the Great Recession of 2008 would be slow.

Table 1: Examples for filtering of words in the context (News Commentary v14 English→German).

<table>
<thead>
<tr>
<th>Original source</th>
<th>Remove stopwords</th>
<th>Remove most frequent words</th>
<th>Retain named entities</th>
<th>Retain specific POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>in recent years, I correctly foresaw that, in the absence of stronger fiscal stimulus (which was not forthcoming in either Europe or the United States), recovery from the Great Recession of 2008 would be slow.</td>
<td>recent years, I correctly foresaw absence stronger fiscal stimulus (forthcoming Europe United States), recovery Great Recession 2008 slow.</td>
<td>recent correctly foresaw absence stronger fiscal stimulus forthcoming either States recovery Great Recession 2008 slow</td>
<td>years I foresaw the absence stimulus was forthcoming either Europe or the United States recovery the Great Recession 2008 would be</td>
<td></td>
</tr>
</tbody>
</table>

3 Experiments

We evaluate the document-level approaches in IWSLT 2017 English→Italian and WMT 2018 English→German translation tasks. We used TED talk or News Commentary v14 dataset as the training data respectively, preprocessed with the Moses tokenizer\footnote{https://github.com/zalandoresearch/flair} and byte pair encoding (Sennrich et al., 2016) trained with 32k merge operations.\footnote{https://sites.google.com/site/iwsltevaluation2017} All our experiments, one previous source sentence was given as the document-level context. A special token was inserted at each document boundary, which was also fed as context input when translating sentences around the boundaries. Detailed corpus statistics are given in Table 2.

All experiments were carried out with \textsc{Sockeye} (Hieber et al., 2018). We used Adam optimizer (Kingma and Ba, 2015) with the default parameters. The learning rate was reduced by 30\% when the perplexity on a validation set was not improving for four checkpoints. When it did not improve for ten checkpoints, we stopped the training. Batch size was 3k tokens, where the bucketing was done for a tuple of current/context sentence lengths. All other settings follow a 6-layer base Transformer model (Vaswani et al., 2017).

In all our experiments, a sentence-level model was pre-trained and used to initialize document-level models, which was crucial for the performance. We also shared the source word embeddings over the original and context encoders.

### Table 2: Training data statistics.

\begin{tabular}{l|c|c}
\hline
Running Words & \texttt{en-it} & \texttt{en-de} \\
Sentences & 4.3M & 8.1M \\
Documents & 227k & 329k \\
Document Length (avg. #sent) & 2,045 & 8,891 \\
\hline
\end{tabular}

3.1 Model Comparison

**Model Architecture** Firstly, we compare the performance of existing single-encoder and multi-encoder approaches (Table 3). For each category of document-level methods (Section 2), we test one representative architecture (Figures 2, 3, 4) which encompasses all existing work in that category except slight variations. The tested methods are equal or closest to:

<table>
<thead>
<tr>
<th>Approach</th>
<th>Context Encoder</th>
<th>en-it</th>
<th>en-de</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Architecture</td>
<td>BLEU [%]</td>
<td>TER [%]</td>
</tr>
<tr>
<td>Baseline</td>
<td>-</td>
<td>31.4</td>
<td>56.1</td>
</tr>
<tr>
<td>Single-Encoder</td>
<td>Transformer</td>
<td>31.5</td>
<td>57.2</td>
</tr>
<tr>
<td>Multi-Encoder (Out.)</td>
<td>Transformer</td>
<td>31.3</td>
<td>56.1</td>
</tr>
<tr>
<td>Multi-Encoder (Seq.)</td>
<td>Transformer</td>
<td>32.6</td>
<td>55.2</td>
</tr>
<tr>
<td>Multi-Encoder (Para.)</td>
<td>Transformer</td>
<td>32.7</td>
<td>54.7</td>
</tr>
<tr>
<td></td>
<td>word embeddings</td>
<td>32.5</td>
<td>54.8</td>
</tr>
</tbody>
</table>

Table 3: Comparison of document-level model architectures and complexity.

- Integration outside the decoder: Voita et al. (2018) without sharing the encoder hidden layers over current/context sentences
- Integration inside the decoder
  - Sequential attention: Decoder integration of Zhang et al. (2018) with the order of attentions (current/context) switched
  - Parallel attention: Gating version of Bawden et al. (2018)

The training of the single-encoder method was quite unstable. It took about twice as long as other document-level models, yet yielding no improvements, which is consistent with Kuang and Xiong (2018). Longer inputs make the encoder-decoder attention widely scattered and harder to optimize. We might need larger training data, massive pre-training, and much larger batches to train the single-encoder approach effectively (Junczys-Dowmunt, 2019); however, these conditions are often not realistic.

For the multi-encoder models, if the context is integrated outside the decoder (“Out.”), it barely improves upon the baseline. By letting the decoder directly access context sentences with a separate attention component, they all outperform the single-encoder method, improving the sentence-level baseline up to +1.4% BLEU and -1.9% TER. Particularly, when attending to current and context sentences in parallel (“Para.”), it provides more flexible and selective information flow from multiple source sentences to the decoder, thus producing better results than the sequential attentions (“Seq.”).

**Model Complexity** In the linguistic sense, surrounding sentences are useful in translating the current sentence mostly by providing case distinctions of nouns or topic information (Section 4). The sequential relation of tokens in the surrounding sentences is important for neither of them. Therefore we investigate how many levels of sequential encoding is actually needed for the improvement by the context. From a 6-layer Transformer encoder, we gradually reduce the model complexity of the context encoder: 2-layer, 1-layer, and only using word embeddings without any sequential encoding. We remove positional encoding (Vaswani et al., 2017) when we encode only with word embeddings.

The results are shown in the lower part of Table 3. Context encoding without any sequential modeling (the last row) shows indeed comparable performance to using a full 6-layer encoder. This simplified encoding eases the memory-intensive document-level training by having 22% fewer model parameters, which allows us to adopt a larger batch size without accumulating gradients. For the remainder of this paper, we stick to using the multi-encoder approach with parallel attention components in the decoder and restricting the context encoding to only word embeddings.

### 3.2 Filtering Words in the Context
To make the context modeling even lighter, we analyze the effectiveness of the filtered context (Section 2.3) in Table 4. All filtering methods shrink the context input drastically without a significant loss of performance. Each method has its own motivation to retain only useful tokens in the
context; the results show that they are all reasonable in practice. In particular, using only named entities as context input, we achieve the same level of improvement with only 13% of tokens in the full context sentences. By filtering words in the context sentences, we can use more examples in each batch for a robust training.

### 3.3 Context Length

Filtered context inputs (Section 3.2) with a minimal encoding (Section 3.1) make it also feasible to include much longer context without much difficulty. Most of previous works on document-level NMT have not examined context inputs longer than three sentences.

Figure 5 shows the translation performance with an increasing number of context sentences. If we concatenate full context sentences (plain curves), the performance deteriorates severely. We found that it is hard to fit such long sequences in memory as the training becomes very erratic.

The training is much more stable with filtered context; the dashed/dotted curves do not drop significantly even when using 20 context sentences. In the English→Italian task, the performance slightly improves up to 15 context sentences. In the English→German task, there is no improvement by extending the context length over 5 sentences. This discrepancy can be explained with document lengths in each dataset (Table 2). The TED talk corpus for English→Italian has much longer documents, thus it is probable to benefit from larger context windows. However, in general we observe only marginal improvements by enlarging the context length to more than one sentence, as seen also in Bawden et al. (2018), Miculicich et al. (2018), or Zhang et al. (2018).

### 4 Analysis

Simplifying the context encoder (Section 3.1) and filtering the context input (Section 3.2) are both inspired by the intuition that only a small part of the context is useful for NMT. In order to verify this intuition rigorously, we conduct an extensive analysis on how document-level context helps the translation process, manually checking every output of sentence-level/document-level NMT.
models; automatic metrics are inherently not suitable for distinguishing document-level behavior. Our analysis is not constrained to certain discourse phenomena which are favored in evaluating document-level models. We quantify various causes of the improvements 1) regardless of its linguistic interpretability and 2) in a realistic scenario where not all the test examples require document-level context. Here are the steps we take:

1. Translate a test set with a sentence-level baseline and a document-level model.
2. Compute per-sentence TER scores of outputs from both models.
3. Select those cases where the document-level model improves the per-sentence TER over the sentence-level baseline.
4. Examine each case of 3 by looking at:
   - Source, context, and translation outputs
   - Attention distribution over the context tokens for each target token: averaged over all decoder layers/heads
   - Gating activation (Equation 1)
5. Classify each case into “coreference”, “topic-aware lexical choice”, or “not interpretable”.

Statistics of each category on the test sets are reported in Table 5. The manual inspection of translation outputs is done by a native-level speaker of Italian or German, respectively.

Only a couple of cases belong to coreference, which is ironically the most advocated improvement in the literature on document-level NMT. One of them is shown in Table 6a. In the document-level NMT, the English word “said” is translated to a correct conjugation of “sagen” (= say) for the third person noun “der Präsident” (= the President). This can be explained by the high attention energy on “Trump” (Figure 7a) in the context sentence.

Another interpretable cause is topic-aware lexical choice (Table 6b). The document-level model actively attends to “seized” and “cocaine” in the context sentence (Figure 7b), and does not miss the source word “raids” in the translation (“Razzien”). When it corrects the translation of polysemous words, it is related to word sense disambiguation (Gonzales et al., 2017; Marvin and Koehn, 2018; Pu et al., 2018). This category includes also a coherence of text style in the translation outputs, depending on the context topic.

<table>
<thead>
<tr>
<th>Category</th>
<th>en-it</th>
<th>en-de</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coreference</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>Topic-aware lexical choice</td>
<td>66</td>
<td>33</td>
</tr>
<tr>
<td>Not interpretable</td>
<td>292</td>
<td>1,211</td>
</tr>
<tr>
<td>Total TER improved</td>
<td>379</td>
<td>1,246</td>
</tr>
<tr>
<td>Total</td>
<td>1,147</td>
<td>2,998</td>
</tr>
</tbody>
</table>

Table 5: Causes of improvements by document-level context.

We found that only 7.5% of the TER-improved cases can be interpreted as utilizing document-level context. The other cases are mostly general improvements in adequacy or fluency which are not related to the given context. Table 6c shows such an example. It improves the translation by a long-range reordering and rephrasing some nouns, whose clues do not exist in the previous source sentence. Its attention distribution over the context words is totally random and blurry (Figure 7c).

A possible reason for the non-interpretable improvements is regularization of the model, since the training data of our experiments are relatively small. Figure 6 shows that, for most of the improved cases, the model has non-negligible gating activation towards document-level context, even if the output seems not to benefit from the context. It means that, when combining the encoded representations of context/current sentences, the model can reserve some of its capacity to the information from context inputs. This might effectively mitigate overfitting to the given training data.

Figure 6: Gating activation for all TER-improved cases of the English→German task, averaged over all layers and target positions.
inside the White House, Trump addressed Sikorsky representatives, joking with the media about his own fleet of company products.

“I know Sikorsky very well,” the President said, “I have three of them.”

in addition, officials seized large quantities of marijuana and cocaine, firearms and several hundred thousand euros.

at simultaneous raids in Italy, two people were detained.

die Rede ist von einem regelrechten „Wettbewerb der öffentlichen Arbeitgeber“.


die Rede ist von einem herben „Wettbewerb zwischen öffentlichen Arbeitgebern“.

Table 6: Example translation outputs for each analysis category (WMT English→German newstest2018).

Figure 7: Attention distribution over context words from target hypothesis.
We argue that the linguistic improvements with document-level NMT have been sometimes oversold, and the document-level components should be tested on top of a well-regularized NMT system. In our experiments, we obtain a much stronger sentence-level baseline by applying a simple regularization (dropout), which the document-level model cannot outperform (Table 7).

On a larger scale, we also built a sentence-level model with all parallel training data available for the WMT 2019 task and fine-tuned only with document-level data (Europarl, News Commentary, newstest2008-2014/2016). The document-level training does not give any improvement in BLEU (last two rows of Table 7). There may exist document-level improvements which are not highlighted by the automatic metrics, but the amount of such improvements must be very small without a clear gain in BLEU or TER.

### 5 Conclusion

In this work, we critically investigate the advantages of document-level NMT with a thorough qualitative analysis and expose the limit of its improvements in terms of context length and model complexity. Regarding the questions asked in Section 1, our answers are:

- In general, document-level context is utilized rarely in an interpretable way.
- We conjecture that a dominant cause of the improvements by document-level NMT is actually the regularization of the model.
- Not all of the words in the context are used in the model; we leave out redundant tokens without loss of performance.

- A long-range context gives only marginal additional improvements.
- Word embeddings are sufficient to model document-level context.

For a fair evaluation of document-level NMT methods, we argue that one should make a sentence-level NMT baseline as strong as possible first, i.e. by using more data or applying proper regularization. This will get rid of by-product improvements from additional information flows and help to focus only on document-level errors in translation. In this condition, we show that document-level NMT can barely improve translation metric scores against such strong baselines. Targeted test sets (Bawden et al., 2018; Voita et al., 2019) might be helpful here to emphasize the document-level improvements. However, one should bear in mind that a big improvement in such test sets may not carry over to practical scenarios with general test sets, where the number of document-level errors in translation is inherently small.

Given these conclusions, a future research direction would be building a lightweight post-editing model to correct only document-level errors, not complicating the sentence-level model too much for a very limited amount of document-level improvements. To strengthen our arguments, we also plan to conduct the same qualitative analysis on other types of context inputs (e.g. translation history) and different domains.

Our implementation of document-level NMT methods is publicly available on the web.\(^6\)

### Acknowledgments

This work has received funding from the European Research Council (ERC) (under the European Union’s Horizon 2020 research and innovation programme, grant agreement No 694537, project “SEQCLAS”). The work reflects only the authors’ views and none of the funding agencies is responsible for any use that may be made of the information it contains.

The authors thank Tina Raissi for analyzing English→Italian translations.

\(^6\)https://github.com/ducthanhtran/sockeye_document_context
References


A  Context Word Filtering Details

The tables below follow the tagging conventions of FLAIR (https://github.com/zalandoresearch/flair).

<table>
<thead>
<tr>
<th>Tag</th>
<th>Examples</th>
<th>Tag</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>mountain ranges, bodies of water</td>
<td>Law</td>
<td>named laws</td>
</tr>
<tr>
<td>Organization</td>
<td>companies, agencies</td>
<td>Language</td>
<td>any named language</td>
</tr>
<tr>
<td>Person</td>
<td>People, fictional characters</td>
<td>Dates</td>
<td>important holidays</td>
</tr>
<tr>
<td>NORP</td>
<td>nationalities, political groups</td>
<td>Time</td>
<td>times smaller than a day</td>
</tr>
<tr>
<td>Facility</td>
<td>buildings, airports</td>
<td>Percent</td>
<td>percentage values</td>
</tr>
<tr>
<td>GPE</td>
<td>countries, cities</td>
<td>Money</td>
<td>monetary values</td>
</tr>
<tr>
<td>Product</td>
<td>vehicles, food</td>
<td>Ordinal</td>
<td>first, second, ...</td>
</tr>
<tr>
<td>Event</td>
<td>named hurricanes, battles</td>
<td>Cardinal</td>
<td>other numeric values</td>
</tr>
</tbody>
</table>

Table 8: Retained named entities.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Coordinating conjunction</td>
<td>PDT</td>
<td>Predeterminer</td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal number</td>
<td>PRP</td>
<td>Personal pronoun</td>
</tr>
<tr>
<td>CODE</td>
<td>Code IDs</td>
<td>PRPS</td>
<td>Possessive pronoun</td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
<td>VBG</td>
<td>Verb, gerund/present particle</td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
<td>VBN</td>
<td>Verb, past participle</td>
</tr>
<tr>
<td>NN</td>
<td>Noun, singular</td>
<td>VBP</td>
<td>Verb, non-3rd person singular present</td>
</tr>
<tr>
<td>NNP</td>
<td>Proper noun, singular</td>
<td>VBZ</td>
<td>Verb, 3rd person singular present</td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
<td>VB</td>
<td>Verb, base form</td>
</tr>
<tr>
<td>NNPS</td>
<td>Proper noun, plural</td>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
</tbody>
</table>

Table 9: Retained parts-of-speech.