Inter-document Contextual Language Model

Quan Hung Tran and **Ingrid Zukerman** and **Gholamreza Haffari**
Faculty of Information Technology
Monash University, Australia
hung.tran,ingrid.zukerman,gholamreza.haffari@monash.edu

Abstract

In this paper, we examine the impact of employing contextual, structural information from a tree-structured document set to derive a language model. Our results show that this information significantly improves the accuracy of the resultant model.

1 Introduction

Conventional Language Models (LMs) are based on n-grams, and thus rely upon a limited number of preceding words to assign a probability to the next word in a document. Recently, Mikolov *et al.* (2010) proposed a Recurrent Neural Network (RNN) LM which uses a vector representation of all the preceding words in a sentence as the context for language modeling. This model, which theoretically can utilize an infinite context window within a sentence, yields an LM with lower perplexity than that of n-gram-based LMs. However, the model does not leverage the wider contextual information provided by words in other sentences in a document or in related documents.

Several researchers have explored extending the contextual information of an RNN-based LM. Mikolov and Zweig (2012) proposed a context-dependent RNN LM that employs Latent Dirichlet Allocation for modeling a long span of context. Wang and Cho (2015) offered a bag-of-words representation of preceding sentences as the context for the RNN LM. Ji *et al.* (2015) used a Document-Context LM (DCLM) to leverage both intra- and inter-sentence context.

These works focused on contextual information at the document level for LM, but did not consider information at the inter-document level. Many document sets on the Internet are structured, which means there are connections between different documents. This phenomenon is prominent in social media, where all the posts are directly linked to several other posts. We posit that these related documents could hold important information about a particular post, including the topic and language use, and propose an RNN-based LM architecture that utilizes both intra- and inter-document contextual information. Our approach, which was tested on the social media dataset `reddit`, yielded promising results, which significantly improve on the state of the art.

2 Dataset

We used pre-collected `reddit` data,[1] which as of December, 2015, consists of approximately 1.7 billion comments in JSON format. A comment thread starts with a "topic", which might be a link or an image. The users then begin to comment on the topic, or reply to previous comments. Over time, this process creates a tree-structured document repository (Figure 1), where a level indicator is assigned to each comment, e.g., a response to the root topic is assigned level 1, and the reply to a level n comment is assigned level $n + 1$. We parsed the raw data in JSON format into a tree structure, removing threads that have less than three comments, contain deleted comments, or do not have comments above

[1] `https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment`

Proceedings of NAACL-HLT 2016, pages 762–766,
San Diego, California, June 12-17, 2016. ©2016 Association for Computational Linguistics

Figure 1: `reddit` example

Table 1: Dataset statistics

	# of threads	# of posts	# of sentences	# of tokens
training	1500	14592	40709	648624
testing	500	5007	13612	217164
validation	100	968	2762	44575

level 2. We randomly selected 2100 threads that fit these criteria. The data were then split into training/testing/validation sets. Table 1 displays some statistics of our dataset.

3 Baseline Neural Language Models

Our inter-document contextual language model scaffolds on the RNN LM (Mikolov et al., 2010) and DCLMs (Ji et al., 2015), as described below.

RNN-LSTM. Given a sentence $\{x_t\}_{t\in[1,\ldots,N]}$, where x_t is the vector representation of the t-th word in the sentence, and N is the length of the sentence, Mikolov *et al.*'s (2010) RNN LM can be defined as:

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) \quad (1)$$

$$\boldsymbol{y}_t \sim \text{softmax}\left(\boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \quad (2)$$

where $\boldsymbol{h}_t$ is the hidden unit at word t, and $\boldsymbol{y}_t$ is the prediction of the t-th word given the previous hidden unit $\boldsymbol{h}_{t-1}$. The function $\boldsymbol{f}$ in Equation 1 can be any non-linear function. Following the approach in (Sundermeyer et al., 2012) and (Ji et al., 2015), we make use of Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997) rather than the simple hidden units used in the original RNN LM. In our work, the word representation $\boldsymbol{x}_t$ is obtained from the one-hot representation using an affine transformation, as follows:

$$\boldsymbol{x}_t = \boldsymbol{W}_p \boldsymbol{o}_t + \boldsymbol{b}_p \quad (3)$$

where $\boldsymbol{o}_t$ is the one-hot representation, $\boldsymbol{W}_p$ is the projection matrix, and $\boldsymbol{b}_p$ is a bias term.

Document Context LMs (DCLMs). We re-implemented two of Ji *et al.*'s (2015) DCLMs as our baselines,[2] viz Context-to-context (Figure 2a) and Context-to-output (Figure 2b). These models extend the RNN-LSTM model by leveraging information from preceding sentences.

The context-to-context model (ccDCLM) concatenates the final hidden unit of the previous sentence with the word vectors of the current sentence. Thus, Equation 1 becomes:

$$\boldsymbol{h}_t^i = \boldsymbol{f}(\boldsymbol{h}_{t-1}^i, \boldsymbol{x}'_{i,t}) \quad (4)$$

$$\boldsymbol{x}'_{i,t} = \text{concat}\left(\boldsymbol{x}_{i,t}, \boldsymbol{h}_{N_{i-1}}^{i-1}\right) \quad (5)$$

where N_{i-1} is the length of the previous sentence in the document, $\boldsymbol{x}_{i,t}$ is the vector representation of the t-th word in the i-th sentence, $\boldsymbol{x}'_{i,t}$ is the concatenation of the vector representation $\boldsymbol{x}_{i,t}$ and the previous sentence's final hidden unit $\boldsymbol{h}_{N_{i-1}}^{i-1}$.

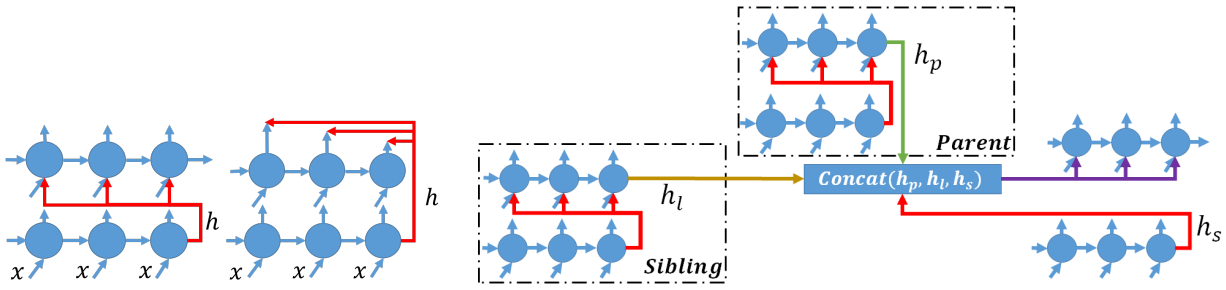
The context-to-output model (coDCLM) applies the additional information directly to the word-decoding phase. Thus, Equation 2 becomes:

$$\boldsymbol{y}_{i,t} \sim \text{softmax}\left(\boldsymbol{W}_o \boldsymbol{h}_{t-1}^i + \boldsymbol{W}'_o \boldsymbol{h}_{N_{i-1}}^{i-1} + \boldsymbol{b}_o\right) \quad (6)$$

4 Inter-document Context Language Model

We now extend the DCLM by leveraging the information at the inter-document level, taking advantage of the structure of the repository — a tree in `reddit`. Specifically, by harnessing the information in documents related to a target document, i.e., its siblings and parent, the LM is expected to contain additional relevant information, and hence lower perplexity. Formally, let's call the sentence-level context vector h_s, the parent document context

[2] Ji *et al.*'s three options performed similarly.



(a) ccDCLM (b) coDCLM (c) Our model (PS-ccDCLM)

Figure 2: Contextual language models; see Sections 3 and 4 for detailed descriptions.

vector h_p, the sibling context vector h_l, and the overall context vector h_c. Our framework is defined as:

$$\boldsymbol{h}_c = \boldsymbol{g}_h(\boldsymbol{h}_s, \boldsymbol{h}_l, \boldsymbol{h}_p) \quad (7)$$

$$\boldsymbol{x}'_{i,t} = \boldsymbol{g}_i(\boldsymbol{x}_{i,t}, \boldsymbol{h}_c) \quad (8)$$

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{h}_{t-1}, \boldsymbol{x}'_t) \quad (9)$$

$$\boldsymbol{y}_t \sim \text{softmax}\left(\boldsymbol{g}_o(\boldsymbol{h}_{t-1}, \boldsymbol{x}'_t, \boldsymbol{h}_c)\right) \quad (10)$$

We use the last hidden vector of the RNNs as the representation of the parent post, the older-sibling, and the previous sentence. The definition of the context function ($\boldsymbol{g}_h$), the input function ($\boldsymbol{g}_i$), and the word-decoding function ($\boldsymbol{g}_o$) yields different configurations.

We also explored two strategies of training the models: *Disconnected (disC)* and *Fully Connected (fulC)*. In the disC-trained models, the error signal within a time step (i.e. a post or sentence) only affects the parameters in that time step. This is in contrast to the fulC-trained models, where the error signal is propagated to the previous time steps, hence influencing parameters in those time steps too.

4.1 Analysis of our modelling approach

In this section, we empirically analyze different training and modelling decisions within our framework, namely DC vs FC training, as well as contextual information from parent vs sibling.

The Setup. For our analysis, we employed a subset of the data described in Table 1 which contains 450 threads split into training/testing/validation sets with 300/100/50 threads respectively. The hidden-vector and word-vector dimensions were set to 50 and 70, respectively. The models were implemented in Theano (Bastien et al., 2012; Bergstra et al., 2010), and trained with RMSProp (Tieleman and Hinton, 2012).

Table 2: disC/fulC-trained models vs the baselines.

Model	Training	Perplexity
6-gram	na	205
RNN-LSTM	na	182
ccDCLM	disC	185
coDCLM	disC	181
ccDCLM	fulC	176
coDCLM	fulC	172

disC vs fulC. We first compared the disC and fulC strategies, at the sentence level only, in order to select the best strategy in a known setting. To this effect, we re-implemented Ji *et al.*'s (2015) DCLMs with the disC strategy, noting that Ji *et al.*'s original sentence-based models are fulC-trained. The results of this experiment appear in Table 2 which further compares these models with the following baselines: (1) vanila RNN-LSTM, and (2) a 6-gram LM with Kneser-Ney smoothing[3] (Kneser and Ney, 1995). The disC-trained models showed no improvement over the RNN-LSTM, and lagged behind their fulC-trained counterparts. The lower performance of the disC-trained models may be due to not fully leveraging the contextual information; disC-training lose information, as the error signal from the current time step is not used to calibrate the parameters of previous time steps. Therefore, we make use of fulC strategy to train our models in the rest of this paper.

Parent vs Sibling Context. The inter-document information in `reddit`'s case may come from a parent post, sibling posts or both. We tested our models with different combinations of inter-document con-

[3]Tested with the SRILM toolkit (Stolcke et al., 2011).

text information to reflect these options. At present, we consider only the closest older-sibling of a post, as it is deemed the most related; different combinations of sibling posts are left for future work. We tested the following three context-to-context configurations: parent only (P-ccDCLM), sibling only (S-ccDCLM), and parent and sibling (PS-ccDCLM), which define the context function as Equation 11, 12 and 13 respectively. The three configurations use the same word-decoding function (Equation 15) and the same input function (Equation 14).

$$\boldsymbol{h}_c = \text{concat}\left(\boldsymbol{h}_s, \boldsymbol{h}_p\right) \quad (11)$$

$$\boldsymbol{h}_c = \text{concat}\left(\boldsymbol{h}_s, \boldsymbol{h}_l\right) \quad (12)$$

$$\boldsymbol{h}_c = \text{concat}\left(\boldsymbol{h}_s, \boldsymbol{h}_l, \boldsymbol{h}_p\right) \quad (13)$$

$$\boldsymbol{x}'_{i,t} = \text{concat}\left(\boldsymbol{x}_{i,t}, \boldsymbol{h}_c\right) \quad (14)$$

$$\boldsymbol{y}_{i,t} \sim \text{softmax}\left(\boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \quad (15)$$

The results of this experiment appear in the first three rows of Table 3, which shows that the best-performing model is PS-ccDCLM.

As discussed by Ji *et al.* (2015), the coDCLM makes the hidden units of the previous sentence have no effect on the hidden units of the current sentence. While this configuration might have some advantages (Ji et al., 2015), applying it directly to a larger context may lead to complications. Suppose we use the last hidden unit of the previous document as the context for the next document. With the context-to-output approach, the last hidden unit summarizes only the information in the last sentence of the previous document, and doesn't reflect the entire document. We address this problem by not using the context-to-output approach in isolation. Instead, we use the context-to-output approach in tandem with the context-to-context approach of ccDCLM. This approach was tested in an additional parent-sibling configuration (PS-ccoDCLM), as an alternative to the best performing context-to-context configuration. The PS-ccoDCLM is similar to the PS-ccDCLM except for the decoding equation, which is changed into Equation 16.

$$\boldsymbol{y}_{i,t} \sim \text{softmax}\left(\boldsymbol{W}_o \boldsymbol{h}^i_{t-1} + \boldsymbol{W}'_o \boldsymbol{h}_c + \boldsymbol{b}_o\right) \quad (16)$$

Based on the results of these trials, we chose the best-performing PS-ccDCLM (Figure 2c) as our final system.

Table 3: Comparing models incorporating parent (P) and/or sibling (S) contextual information.

Systems	Perplexity
P-ccDCLM	172
S-ccDCLM	174
PS-ccDCLM	**168**
PS-ccoDCLM	175

Table 4: Results on the entire dataset.

Systems	Perplexity
6-gram	209
RNN-LSTM	184
ccDCLM	168
coDCLM	176
PS-ccDCLM	**159**

4.2 Results

The model perplexity obtained by the baselines and our best-performing model for the test set (Table 1) is shown in Table 4 — our system (PS-ccDCLM) statistically significantly outperforms the best baseline (ccDCLM), with $\alpha = 0.01$, using the Friedman test. The inter-sentence contextual information under the context-to-context regime (ccDCLM) decreases model perplexity by 9% compared to the original RNN-LSTM, while the inter-document contextual information (PS-ccDCLM) reduces perplexity by a further 5% compared to ccDCLM.

5 Discussion and Future Work

Our results show that including inter-document contextual information yields additional improvements to those obtained from inter-sentence information. However, as expected, the former are smaller than the latter, as sentences in the same post are more related than sentences in different posts. At present, we rely on the final hidden-vector of the sentences and the posts for contextual information. In the future, we propose to explore other options, such as additional models to combine the contextual information from all siblings in the tree structure, and extending our model to structures beyond trees.

References

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June. Oral Presentation.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yangfeng Ji, Trevor Cohn, Lingpeng Kong, Chris Dyer, and Jacob Eisenstein. 2015. Document context language models. *arXiv preprint arXiv:1511.03962*.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *ICASSP-95*, volume 1, pages 181–184. IEEE.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *SLT*, pages 234–239.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, page 5.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *INTERSPEECH*, pages 194–197.

Tijmen. Tieleman and Geoffrey Hinton. 2012. Neural Networks for Machine Learning. `http://www.youtube.com/watch?v=O3sxAc4hxZU`. [Online].

Tian Wang and Kyunghyun Cho. 2015. Larger-context language modelling. *arXiv preprint arXiv:1511.03729*.