Incorporating Side Information into Recurrent Neural Network Language Models

Cong Duy Vu Hoang
University of Melbourne
Melbourne, VIC, Australia
vhoang2@student.unimelb.edu.au

Gholamreza Haffari
Monash University
Clayton, VIC, Australia
gholamreza.haffari@monash.edu

Trevor Cohn
University of Melbourne
Melbourne, VIC, Australia
t.cohn@unimelb.edu.au

Abstract

Recurrent neural network language models (RNNLM) have recently demonstrated vast potential in modelling long-term dependencies for NLP problems, ranging from speech recognition to machine translation. In this work, we propose methods for conditioning RNNLMs on external side information, e.g., metadata such as keywords, description, document title or topic headline. Our experiments show consistent improvements of RNNLMs using side information over the baselines for two different datasets and genres in two languages. Interestingly, we found that side information in a foreign language can be highly beneficial in modelling texts in another language, serving as a form of cross-lingual language modelling.

1 Introduction

Neural network approaches to language modelling (LM) have made remarkable performance gains over traditional count-based $n$gram LMs (Bengio et al., 2003; Mnih and Hinton, 2007; Mikolov et al., 2011). They offer several desirable characteristics, including the capacity to generalise over large vocabularies through the use of vector space representation, and – for recurrent models (Mikolov et al., 2011) – the ability to encode long distance dependencies that are impossible to include with a limited context windows used in conventional $n$gram LMs. These early papers have spawned a cottage industry in neural LM based applications, where text generation is a key component, including conditional language models for image captioning (Kiros et al., 2014; Vinyals et al., 2015) and neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015).

Inspired by these works for conditioning LMs on complex side information, such as images and foreign text, in this paper we investigate the possibility of improving LMs in a more traditional setting, that is when applied directly to text documents. Typically corpora include rich side information, such as document titles, authorship, time stamp, keywords and so on, although this information is usually discarded when applying statistical models. However, this information can be highly informative, for instance, keywords, titles or descriptions, often include central topics which will be helpful in modelling or understanding the document text. We propose mechanisms for encoding this side information into a vector space representation, and means of incorporating it into the generating process in a RNNLM framework. Evaluating on two corpora and two different languages, we show consistently significant perplexity reductions over the state-of-the-art RNNLM models.

The contributions of this paper are as follows:

1. We propose a framework for encoding structured and unstructured side information, and its incorporation into a RNNLM.
2. We introduce a new corpus, the RIE corpus, based on the Europarl web archive, with rich annotations of several types of meta-data.
3. We provide empirical analysis showing consistent improvements from using side information across two datasets in two languages.

Proceedings of NAACL-HLT 2016, pages 1250–1255,
San Diego, California, June 12-17, 2016.

2 Problem Formulation & Model

We first review RNNLM architecture (Mikolov et al., 2011) before describing our extension in §2.2.

2.1 RNNLM Architecture

The standard RNNLM consists of 3 main layers: an input layer where each input word has its embedding via one-hot vector coding; a hidden layer consisting of recurrent units where a state is conditioned recursively on past states; and an output layer where a target word will be predicted. RNNLM has an advantage over conventional n-gram language model in modelling long distance dependencies effectively.

In general, an RNN operates from left-to-right over the input word sequence; i.e.,

$$\begin{aligned}\boldsymbol{h}_t &= \mathrm{RU}\left(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}\right) \\ &= \mathrm{f}\left(\boldsymbol{W}^{(hh)}\boldsymbol{h}_{t-1} + \boldsymbol{W}^{(ih)}\boldsymbol{x}_t + \boldsymbol{b}^{(h)}\right) \\ \boldsymbol{x}_{t+1} &\sim \mathrm{softmax}\left(\boldsymbol{W}^{(ho)}\boldsymbol{h}_t + \boldsymbol{b}^{(o)}\right);\end{aligned}$$

where $f(.)$ is a non-linear function, e.g., $\tanh$, applied element-wise to its vector input; $\boldsymbol{h}_t$ is the current RNN hidden state at time-step t; and matrices $\boldsymbol{W}$ and vectors $\boldsymbol{b}$ are model parameters. The model is trained using gradient-based methods to optimise a (regularised) training objective, e.g. the likelihood function. In principle, a recurrent unit (RU) can be employed using different variants of recurrent structures such as: Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Gated Recurrent Unit (GRU) (Cho et al., 2014), or recently deeper structures, e.g. Depth Gated Long Short Term Memory (DGLSTM) – a stack of LSTMs with extra connections between memory cells in deep layers (Yao et al., 2015). It can be regarded as being a generalisation of LSTM recurrence to both time and depth. Such deep recurrent structure may capture long distance patterns at their most general. Empirically, we found that RNNLM with DGLSTM structure appears to be best performer across our datasets, and therefore is used predominantly in our experiments.

2.2 Incorporating Side Information

Nowadays, many corpora are archived with side information or contextual meta-data. In this work, we argue that such information can be useful for language modelling (and presumably other NLP tasks). By providing this auxiliary information directly to the RNNLM, we stand to boost language modelling performance.

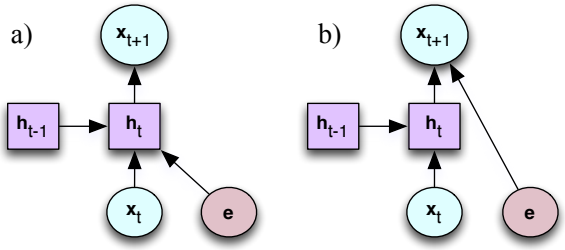

Figure 1: Integration methods for auxiliary information, $\boldsymbol{e}$: a) as input to the RNN, or b) as part of the output $\mathrm{softmax}$ layer.

The first question in using side information is how to encode these unstructured inputs, $\boldsymbol{y}$, into a vector representation, denoted $\boldsymbol{e}$. We discuss several methods for encoding the auxiliary vector:

BOW additive bag of words, $\boldsymbol{e} = \sum_t \boldsymbol{y}_t$, and

average the average embedding vector, $\boldsymbol{e} = \frac{1}{T}\sum_t \boldsymbol{y}_t$, both inspired by (Hermann and Blunsom, 2014a);

bigram convolution with sum-pooling, $\boldsymbol{e} = \sum_t \tanh\left(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t\right)$ (Hermann and Blunsom, 2014b); and

RNN a recurrent neural network over the word sequence (Sutskever et al., 2014), using the final hidden state(s) as $\boldsymbol{e}$.

From the above methods, we found that BOW worked consistently well, outperforming the other approaches, and moreover lead to a simpler model with faster training. For this reason we report only results for the BOW encoding. Note that when using multiple auxiliary inputs, we use a weighted combination, $\boldsymbol{e} = \sum_i \boldsymbol{W}^{(ai)}\boldsymbol{e}^{(i)}$.

The next step is the integration of $\boldsymbol{e}$ into the RNNLM. We consider two integration methods: as input to the hidden state (denoted **input**), and connected to the output $\mathrm{softmax}$ layer (**output**), as shown in Figure 1 a and b, respectively. In both cases, we compare experimentally the following integration strategies:

add adding the vectors together, e.g., using $\boldsymbol{x}_t + \boldsymbol{e}$ as the input to the RNN, such that

$\boldsymbol{h}_t = \mathrm{RU}\left(\boldsymbol{x}_t + \boldsymbol{e}, \boldsymbol{h}_{t-1}\right)$;

stack concatenating the vectors, e.g., using $\left[\boldsymbol{x}_t^\top \boldsymbol{e}^\top\right]^\top$ for generating the RNN hidden state, such that $\boldsymbol{h}_t = \mathrm{RU}\left(\begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{e} \end{bmatrix}, \boldsymbol{h}_{t-1}\right)$; and

mlp feeding both vectors into an extra perceptron with single hidden layer, using a $\tanh$ nonlinearity and projecting the output to the required dimensionality; i.e.,

$$\boldsymbol{h}'_t = \tanh\left(\boldsymbol{W}^{(hh')}\boldsymbol{h}_t + \boldsymbol{W}^{(he)}\boldsymbol{e} + \boldsymbol{b}^{(h')}\right)$$
$$\boldsymbol{x}_{t+1} \sim \mathrm{softmax}\left(\boldsymbol{W}^{(ho)}\boldsymbol{h}'_t + \boldsymbol{b}^{(o)}\right).$$

Note that *add* requires the vectors to be the same dimensionality, while the other two methods do not. The *stack* method can be quite costly, given that it increases the size of several matrices, either in the recurrent unit (for *input*) or the *output* mapping for word generation. This is a problem in the latter case: given the large size of the vocabulary, the matrix $\boldsymbol{W}^{(ho)}$ is already very large and making it larger (doubling the size, to become $\boldsymbol{W}^{(h'o)}$) has a sizeable effect on training time (and presumably also propensity to over-fit). The *output+stack* method does however have a compelling interpretation as a jointly trained product model between a RNNLM and a unigram model conditioned on the side information, where both models are formulated as $\mathrm{softmax}$ classifiers. Considered as a product model (Hinton, 2002; Pascanu et al., 2013), the two components can concentrate on different aspects of the problem where the other model is not confident, and allowed each model the ability to 'veto' certain outputs, by assigning them a low probability.

3 Experiments

Datasets. We conducted our experiments on two datasets with different genres in two languages. As the first dataset, we use the IWSLT2014 MT track on TED Talks[1] due to its self-contained rich auxiliary information, including: title, description, keywords, and author related information. We chose the English-French pair for our experiments[2] . The statistics of the training set is shown in Table 1. We used dev2010 (7 talks/817 sentences) for early stopping of training neural network models. For evaluation, we used different testing sets over years, including tst2010 (10/1587), tst2011 (7/768), tst2012 (10/1083).

	tokens (M)	types (K)	docs	sents (K)
TED-en	4.0	18.3	1414	179
TED-fr	4.3	22.6	1414	179
RIE-en	13.7	15.0	200	460
RIE-fr	14.9	19.4	200	460

Table 1: Statistics of the training sets, showing in each cell the number of word tokens, types, documents (talks or plenaries), and sentences. Note that "types" here refers to word frequency thresholded at 5 and 15 for TED Talks and RIE datasets, respectively.

As the second dataset, we crawled the entire European Parliament[3] website, focusing on plenary sessions. Such sessions contain useful structural information, namely multilingual texts divided into speaker sessions and topics. We believe that those texts are interesting and challenging for language modelling tasks. Our dataset contains 724 plenary sessions over 12.5 years until June 2011 with multilingual texts in 22 languages[4]. We refer to this dataset by RIE[5] (**R**ich **I**nformation **E**uroparl). We randomly select 200/5/30 plenary sessions as the training/development/test sets, respectively. We believe that the new data including side information pose another challenge for language modelling. Furthermore, the sizes of our working datasets are an order of magnitude larger than the standard Penn Treebank set which is often used for evaluating neural language models.

Set-up and Baselines. We have used cnn[6] to implement our models. We use the same configurations for all neural models: 512 input embedding and hidden layer dimensions, 2 hidden layers, and vocabulary sizes as given in Table 1. We used the same vocabulary for the auxiliary and modelled text. We trained a conventional $5-$gram language model using modified Kneser-Ney smoothing, with the KenLM toolkit (Heafield, 2011). We used the

[1] `https://wit3.fbk.eu/` (IWSLT'14 MT Track)

[2] Our method can be also applied to other language pairs.

[3] `http://www.europarl.europa.eu/`

[4] We ignored the period from June 2011 onwards, as from this date the EU stopped creating manual human translations.

[5] This dataset will be released upon publication.

[6] `https://github.com/clab/cnn/`

Method	test2010	test2011	test2012
5-gram LM	79.9	77.4	89.9
RNNLM	65.8	63.9	73.0
LSTM	54.1	52.2	58.4
DGLSTM	53.1	52.1	58.8
input+add+k	52.9	52.1	57.5
input+mlp+k	53.3	51.5	57.3
input+stack+k	53.7	51.9	58.1
output+mlp+k	**51.7**	**50.6**	**55.8**
output+mlp+t	**52.3**	53.5	58.3
output+mlp+d	**52.0**	**49.8**	**56.3**
output+mlp+k+t	**51.4**	**51.1**	**56.8**
output+mlp+k+d	**51.2**	**49.7**	**55.1**
output+mlp+t+d	**52.6**	51.5	**57.2**
output+mlp+k+t+d	**51.1**	**50.6**	**56.3**

Table 2: Perplexity scores based on the **English** part of TED talks dataset in IWSLT14 MT. +k, +t, +d: with keywords, title, and description as auxiliary side information respectively. **bold**: Statistically significant better than the best baseline.

Method	test2010	test2011	test2012
5-gram LM	65.1	60.3	64.8
LSTM	45.0	42.5	44.0
DGLSTM	44.0	41.9	43.0
output+mlp+t	**42.1**	**40.6**	42.5
output+mlp+d	**40.9**	**38.9**	**40.3**
output+mlp+t+d	**41.7**	**39.8**	42.8
output+mlp+k	**40.8**	**38.3**	**39.7**
output+mlp+d+k	**40.2**	**38.3**	**39.4**

Table 3: Perplexity scores based on the **French** part of TED talks dataset in IWSLT14 MT. Note that +k means with keywords in **English**.

Wilcoxon signed-rank test (Wilcoxon, 1945) to measure the statistical significance ($p < 0.05$) on differences between sentence-level perplexity scores of improved models compared to the best baseline. Throughout our experiments, punctuation, stop words and sentence markers ($\langle$s$\rangle$, $\langle$/s$\rangle$, $\langle$unk$\rangle$) are filtered out in all auxiliary inputs. We observed that this filtering was required for BOW to work reasonably well. For each model, the best perplexity score on development set is used for early stopping of training models, which was obtained after 2-5 and 2-3 epochs on TED Talks and RIE datasets, respectively.

Results & Analysis. The perplexity results on TED Talks dataset are presented in Table 2 and 3. RNNLM variants consistently achieve substantially better perplexities compared to the conventional $5-$gram language model baseline.[7] Of the basic RNNLM models (middle), the DGLSTM works consistently better than both the standard RNN and the LSTM. This may be due to better interactions of memory cells in hidden layers. Since the DGLSTM outperformed others[8], we used it for all subsequent experiments. For TED Talks dataset, there are three kinds of side information, including keywords, title, description. We attempted to inject those into different RNNLM layers, resulting in model variants as shown in Table 2. First, we chose "keywords" (+k) information as an anchor to figure out which incorporation method works well. Comparing *input*+add+k, *input*+mlp+k and *input*+stack+k, the largest decrease is obtained by *output*+mlp+k consistently across all test sets (and development sets, not shown here). We further evaluated the addition of other side information (e.g., "description" (+d), "title" (+t)), finding that +d has similar effect as +k whereas +t has a mixed effect, being detrimental for one test set (test2011). We suspect that it is due to often-times short sentences of titles in that test, after our filtering step, leading to a shortage of useful information fed into neural network learning. Interestingly, the best performance is obtained when incorporating both +k and +d, showing that there is complementary information in the two auxiliary inputs. Further, we also achieved the similar results in the counterpart of English part (in French) using *output*+mlp with both +t and +d as shown in Table 3. In French data, no "keywords" information is available. For this reason, we run additional experiments by injecting English keywords as side information into neural models of French. Interestingly, we found that "keywords" side information in English effectively improves the modelling of French texts as shown in Table 3, serving as a new form of cross-lingual language modelling.

We further achieved similar results by incorporating the topic headline in the RIE dataset. The consistently-improved results (in Table 4) demonstrate the robustness of the *output*+mlp approach.

[7] For fair comparison, when computing the perplexity with the 5-gram LM, we exclude all test words marked as $\langle$unk$\rangle$ (i.e., with low counts or OOVs) from consideration.

[8] This concurs with the finding in (Yao et al., 2015), who showed that DGLSTM produced the state-of-the-art results over Penn Treebank dataset.

Method	test (en)	test (fr)
5-gram LM	55.7	38.5
LSTM	40.3	28.5
DGLSTM	36.4	25.4
output+mlp+h	**33.3**	**24.0**

Table 4: Perplexity scores based on the sampled RIE dataset. +h: topic headline.

4 Conclusion

We have proposed an effective approach to boost the performance of RNNLM using auxiliary side information (e.g. keywords, title, description, topic headline) of a textual utterance. We provided an empirical analysis of various ways of injecting such information into a distributed representation, which is then incorporated into either the input, hidden, or output layer of RNNLM architecture. Our experimental results reveal consistent improvements are achieved over strong baselines for different datasets and genres in two languages. Our future work will investigate the model performance on a closely-related task, i.e., neural machine translation (Sutskever et al., 2014; Bahdanau et al., 2015). Furthermore, we will explore learning methods to combine utterances with and without the auxiliary side information.

Acknowledgements

The authors would like to thank the reviewers for valuable comments and feedbacks. Cong Duy Vu Hoang was supported by research scholarships from the University of Melbourne, Australia. Dr Trevor Cohn was supported by the ARC (Future Fellowship).

References

D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of International Conference on Learning Representations (ICLR 2015)*, September.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July.

K. M. Hermann and P. Blunsom. 2014a. Multilingual Distributed Representations without Word Alignment. In *Proceedings of International Conference on Learning Representations (ICLR 2014)*, December.

Karl Moritz Hermann and Phil Blunsom. 2014b. Multilingual Models for Compositional Distributed Semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland, June. Association for Computational Linguistics.

Geoffrey E Hinton. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural computation*, 14(8):1771–1800.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2013)*.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. 2014. Multimodal Neural Language Models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603.

T. Mikolov, S. Kombrink, A. Deoras, and J. H. Burget, L.and Cernocky. 2011. RNNLM - Recurrent Neural Network Language Modeling Toolkit. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE Automatic Speech Recognition and Understanding Workshop, December.

Andriy Mnih and Geoffrey Hinton. 2007. Three New Graphical Models for Statistical Language Modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648.

R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. 2013. How to Construct Deep Recurrent Neural Networks. *ArXiv e-prints*, December.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems (NIPS 2014)*, pages 3104–3112.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1 (6):80–83, Dec.

K. Yao, T. Cohn, K. Vylomova, K. Duh, and C. Dyer. 2015. Depth-Gated LSTM. *ArXiv e-prints*, August.