Exponentially Decaying Bag-of-Words Input Features for Feed-Forward Neural Network in Statistical Machine Translation

Jan-Thorsten Peter, Weiyue Wang, Hermann Ney
Human Language Technology and Pattern Recognition, Computer Science Department
RWTH Aachen University, 52056 Aachen, Germany
{peter,wwang,ney}@cs.rwth-aachen.de

Abstract

Recently, neural network models have achieved consistent improvements in statistical machine translation. However, most networks only use one-hot encoded input vectors of words as their input. In this work, we investigated the exponentially decaying bag-of-words input features for feed-forward neural network translation models and proposed to train the decay rates along with other weight parameters. This novel bag-of-words model improved our phrase-based state-of-the-art system, which already includes a neural network translation model, by up to 0.5% BLEU and 0.6% TER on three different translation tasks and even achieved a similar performance to the bidirectional LSTM translation model.

1 Introduction

Neural network models have recently gained much attention in research on statistical machine translation. Several groups have reported strong improvements over state-of-the-art baselines when combining phrase-based translation with feed-forward neural network-based models (FFNN) (Schwenk et al., 2006; Vaswani et al., 2013; Schwenk, 2012; Devlin et al., 2014), as well as with recurrent neural network models (RNN) (Sundermeyer et al., 2014). Even in alternative translation systems they showed remarkable performance (Sutskever et al., 2014; Bahdanau et al., 2015).

The main drawback of a feed-forward neural network model compared to a recurrent neural network model is that it can only have a limited context length on source and target sides. Using the *Bag-of-Words* (BoW) model as additional input of a neural network based language model, (Mikolov et al., 2015) have achieved very similar perplexities on automatic speech recognition tasks in comparison to the long short-term memory (LSTM) neural network, whose structure is much more complex. This suggests that the bag-of-words model can effectively store the longer term contextual information, which could show improvements in statistical machine translation as well. Since the bag-of-words representation can cover as many contextual words without further modifying the network structure, the problem of limited context window size of feed-forward neural networks is reduced. Instead of predefining fixed decay rates for the exponentially decaying bag-of-words models, we propose to learn the decay rates from the training data like other weight parameters in the neural network model.

2 The Bag-of-Words Input Features

The bag-of-words model is a simplifying representation applied in natural language processing. In this model, each sentence is represented as the set of its words disregarding the word order. Bag-of-words models are used as additional input features to feed-forward neural networks in addition to the one-hot encoding. Thus, the probability of the feed-forward neural network translation model with an m-word source window can be written as:

$$p(e_1^I \mid f_1^J) \approx \prod_{i=1}^{I} p(e_i \mid f_{b_i-\Delta_m}^{b_i+\Delta_m}, f_{\text{BoW},i}) \qquad (1)$$

where $\Delta_m = \frac{m-1}{2}$ and b_i is the index of the single aligned source word to the target word e_i. We applied the *affiliation* technique proposed in (Devlin et al., 2014) for obtaining the one-to-one align-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 293–298,
Berlin, Germany, August 7-12, 2016.

ments. The bag-of-words input features $f_{\text{BoW},i}$ can be seen as normalized n-of-N vectors as demonstrated in Figure 1, where n is the number of words inside each bag-of-words.

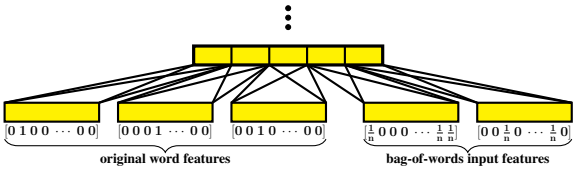


Figure 1: The bag-of-words input features along with the original word features. The input vectors are projected and concatenated at the projection layer. We omit the hidden and output layers for simplification, since they remain unchanged.

2.1 Contents of Bag-of-Words Features

Before utilizing the bag-of-words input features we have to decide which words should be part of it. We tested multiple different variants:

1. Collecting all words of the sentence in one bag-of-words except the currently aligned word.
2. Collecting all preceding words in one bag-of-words and all succeeding words in a second bag-of-words.
3. Collecting all preceding words in one bag-of-words and all succeeding words in a second bag-of-words except those already included in the source window.

All of these variants provide the feed-forward neural network with an unlimited context in both directions. The differences between these setups only varied by 0.2% BLEU and 0.1% TER. We choose to base further experiments on the last variant since it performed best and seemed to be the most logical choice for us.

2.2 Exponentially Decaying Bag-of-Words

Another variant is to weight the words within the bag-of-words model. In the standard bag-of-words representation these weights are equally distributed for all words. This means the bag-of-words input is a vector which marks if a word is given or not and does not encode the word order. To avoid this problem, the *exponential decay* approach proposed in (Clarkson and Robinson, 1997) has been adopted to express the distance of contextual words from the current word. Therefore the bag-of-words vector with decay weights can be defined as following:

$$\tilde{f}_{\text{BoW},i} = \sum_{k \in S_{\text{BoW}}} d^{|i-k|} \tilde{f}_k \qquad (2)$$

where

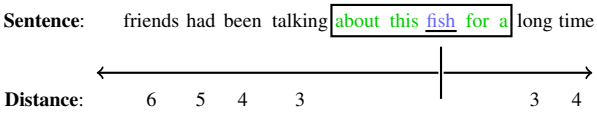
- i, k — Positions of the current word and words within the BoW model respectively.
- $\tilde{f}_{\text{BoW},i}$ — The value vector of the BoW input feature for the i-th word in the sentence.
- $\tilde{f}_k$ — One-hot encoded feature vector of the k-th word in the sentence.
- S_{BoW} — Indices set of the words contained in the BoW. If a word appears more than once in the BoW, the index of the nearest one to the current word will be selected.
- d — Decay rate with float value ranging from zero to one. It specifies how fast weights of contextual words decay along with distances, which can be learned like other weight parameters of the neural network.

Instead of using fixed decay rate as in (Irie et al., 2015), we propose to train the decay rate like other weight parameters in the neural network. The approach presented by (Mikolov et al., 2015) is comparable to the corpus decay rate shown here, except that their work makes use of a diagonal matrix instead of a scalar as decay rate. In our experiments, three different kinds of decay rates are trained and applied:

1. Corpus decay rate: all words in vocabulary share the same decay rate.
2. Individual decay rate for each bag-of-words: each bag-of-words has its own decay rate given the aligned word.
3. Individual decay rate for each word: each word uses its own decay rate.

We use the English sentence **"friends had been talking about this fish for a long time"** as an example to clarify the differences between these variants. A five words contextual window centered at the current aligned word `fish` has been applied: {`about`, `this`, `fish`, `for`, `a`}. The bag-of-words models are used to collect all

other source words outside the context window: {friends, had, been, talking} and {long, time}. Furthermore, there are multiple choices for assigning decay weights to all these words in the bag-of-words feature:



1. Corpus decay rate: d

Weights: d^6 d^5 d^4 d^3 d^3 d^4

2. Bag-of-words individual decay rate: $d = d_{\text{fish}}$

Weights: d^6_{fish} d^5_{fish} d^4_{fish} d^3_{fish} d^3_{fish} d^4_{fish}

3. Word individual decay rate:
$d \in \{d_{\text{friends}}, d_{\text{had}}, d_{\text{been}}, d_{\text{talking}}, d_{\text{long}}, d_{\text{time}}\}$

Weights: d^6_{friends} d^5_{had} d^4_{been} d^3_{talking} d^3_{long} d^4_{time}

3 Experiments

3.1 Setup

Experiments are conducted on the IWSLT 2013 German→English, WMT 2015 German→English and DARPA BOLT Chinese→English translation tasks. GIZA++ (Och and Ney, 2003) is applied for aligning the parallel corpus. The translation quality is evaluated by case-insensitive BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) metric. The scaling factors are tuned with MERT (Och, 2003) with BLEU as optimization criterion on the development sets. The systems are evaluated using *MultEval* (Clark et al., 2011). In the experiments the maximum size of the n-best lists applied for reranking is 500. For the translation experiments, the averaged scores are presented on the development set from three optimization runs.

Experiments are performed using the *Jane* toolkit (Vilar et al., 2010; Wuebker et al., 2012) with a log-linear framework containing following feature functions:

- Phrase translation probabilities both directions
- Word lexicon features in both directions
- Enhanced low frequency counts (Chen et al., 2011)
- 4-gram language model
- 7-gram word class language model (Wuebker et al., 2013)
- Word and phrase penalties
- Hierarchical reordering model (Galley and Manning, 2008)

Additionally, a neural network translation model, similar to (Devlin et al., 2014), with following configurations is applied for reranking the n-best lists:

- Projection layer size 100 for each word
- Two non-linear hidden layers with 1000 and 500 nodes respectively
- Short-list size 10000 along with 1000 word classes at the output layer
- 5 one-hot input vectors of words

Unless otherwise stated, the investigations on bag-of-words input features are based on this neural network model. We also integrated our neural network translation model into the decoder as proposed in (Devlin et al., 2014). The relative improvements provided by integrated decoding and reranking are quite similar, which can also be confirmed by (Alkhouli et al., 2015). We therefore decided to only work in reranking for repeated experimentation.

3.2 Exponentially Decaying Bag-of-Words

As shown in Section 2.2, the exponential decay approach is applied to express the distance of contextual words from the current word. Thereby the information of sequence order can be included into bag-of-words models. We demonstrated three different kinds of decay rates for words in the bag-of-words input feature, namely the corpus general decay rate, the bag-of-words individual decay rate and the word individual decay rate.

Table 1 illustrates the experimental results of the neural network translation model with exponentially decaying bag-of-words input features on IWSLT 2013 German→English, WMT 2015 German→English and BOLT Chinese→English

	IWSLT				WMT		BOLT	
	test		eval11		newstest2013		test	
	BLEU[%]	TER[%]	BLEU[%]	TER[%]	BLEU[%]	TER[%]	BLEU[%]	TER[%]
Baseline + NNTM	31.9	47.5	36.7	43.0	28.8	53.8	17.4	67.1
+ BoW Features	32.0	47.3	36.9	42.9	28.8	53.5*	17.5	67.0
+ Fixed DR (0.9)	32.2*	47.3	37.0*	42.6*†	29.0	53.5*	17.7*	66.8*
+ Corpus DR	32.1	47.3	36.9	42.7*	29.1*†	53.5*	17.7*	66.7*†
+ BoW DR	**32.4***†	**47.0***†	**37.2***†	**42.4***†	**29.2***†	**53.2***†	**17.9***†	**66.6***†
+ Word DR	32.3*†	**47.0***	37.1*	42.7*	29.1*†	53.4*	17.8*†	66.7*†
Baseline + LSTM	32.2*	47.4	37.1*	42.5*†	29.0	53.3*	17.6	66.8*

Table 1: Experimental results of translations using exponentially decaying bag-of-words models with different kinds of decay rates. Improvements by systems marked by $*$ have a 95% statistical significance from the baseline system, whereas † denotes the 95% statistical significant improvements with respect to the `BoW Features` system (without decay weights). We experimented with several values for the fixed decay rate (DR) and 0.9 performed best. The applied RNN model is the LSTM bidirectional translation model proposed in (Sundermeyer et al., 2014).

translation tasks. Here we applied two bag-of-words models to separately contain the preceding and succeeding words outside the context window. We can see that the bag-of-words feature without exponential decay weights only provides small improvements. After appending the decay weights, four different kinds of decay rates provide further improvements to varying degrees. The bag-of-words individual decay rate performs the best, which gives us improvements by up to 0.5% on BLEU and up to 0.6% on TER. On these tasks, these improvements even help the feed-forward neural network achieve a similar performance to the popular long short-term memory recurrent neural network model (Sundermeyer et al., 2014), which contains three LSTM layers with 200 nodes each. The results of the word individual decay rate are worse than that of the bag-of-words decay rate. One reason is that in word individual case, the sequence order can still be missing. We initialize all values for the tunable decay rates with 0.9. In the IWSLT 2013 German→English task, the corpus decay rate is tuned to 0.578. When investigating the values of the trained bag-of-words individual decay rate vector, we noticed that the variance of the value for frequent words is much lower than for rare words. We also observed that most function words, such as prepositions and conjunctions, are assigned low decay rates. We could not find a pattern for the trained value vector of the word individual decay rates.

3.3 Comparison between Bag-of-Words and Large Context Window

The main motivation behind the usage of the bag-of-words input features is to provide the model with additional context information. We compared the bag-of-words input features to different source side windows to refute the argument that simply increasing the size of the window could achieve the same results. Our experiments showed that increasing the source side window beyond 11 gave no more improvements while the model that used the bag-of-words input features is able to achieve the best result (Figure 2). A possible explanation for this could be that the feed-forward neural network learns its input position-dependent. If one source word is moved by one position the feed-forward neural network needs to have seen a word with a similar word vector at this position during training to interpret it correctly. The likelihood of precisely getting the position decreases with a larger distance. The bag-of-words model on the other hand will still get the same input only slightly stronger or weaker on the new distance and decay rate.

4 Conclusion

The aim of this work was to investigate the influence of exponentially decaying bag-of-words input features with trained decay rates on the feed-forward neural network translation model. Applying the standard bag-of-words model as an additional input feature in our feed-forward neural network translation model only yields slight im-

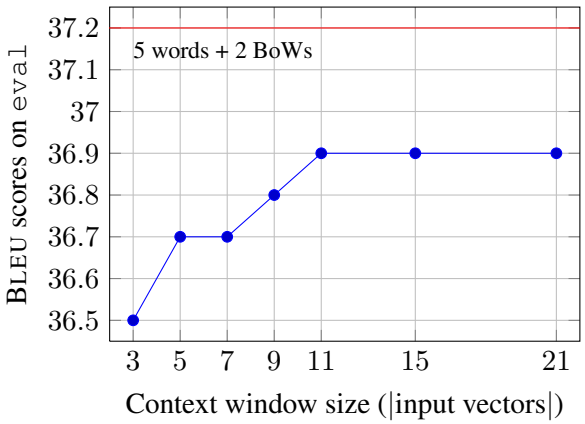


Figure 2: The change of BLEU scores on the eval11 set of the IWSLT 2013 German→English task along with the source context window size. The source windows are always symmetrical with respect to the aligned word. For instance, window size five denotes that two preceding and two succeeding words along with the aligned word are included in the window. The average sentence length of the corpus is about 18 words. The red line is the result of using a model with bag-of-words input features and a bag-of-words individual decay rate.

provements, since the original bag-of-words representation does not include information about the ordering of each word. To avoid this problem, we applied the exponential decay weight to express the distances between words and propose to train the decay rate as other weight parameters of the network. Three different kinds of decay rates are proposed, the bag-of-words individual decay rate performs best and provides improvements by averagely 0.5% BLEU on three different translation tasks, which is even able to outperform a bidirectional LSTM translation model on the given tasks. By contrast, applying additional one-hot encoded input vectors or enlarging the network structure can not achieve such good performances as bag-of-words features.

Acknowledgments

This paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement nº 645452 (QT21).

References

Tamer Alkhouli, Felix Rietig, and Hermann Ney. 2015. Investigations on phrase-based decoding with recurrent neural network language and translation models. In *EMNLP 2015 Tenth Workshop on Statistical Machine Translation (WMT 2015)*, pages 294–303, Lisbon, Portugal, September.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, May.

Boxing Chen, Roland Kuhn, George Foster, and Howard Johnson. 2011. Unpacking and Transforming Feature Functions: New Ways to Smooth Phrase Tables. In *Proceedings of MT Summit XIII*, pages 269–275, Xiamen, China, September.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*, pages 176–181, Portland, OR, USA, June.

P. Clarkson and A. Robinson. 1997. Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 799–802, Washington, DC, USA, April.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380, Baltimore, MD, USA, June.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, HI, USA, October.

Kazuki Irie, Ralf Schlter, and Hermann Ney. 2015. Bag-of-Words Input for Long History Representation in Neural Network-based Language Models for Speech Recognition. In *Proceedings of the 16th Annual Conference of International Speech Communication Association*, pages 2371–2375, Dresden, Germany, September.

Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. 2015. Learning longer memory in recurrent neural networks. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, May.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29:19–51, March.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA, July.

Holger Schwenk, Daniel Déchelotte, and Jean-Luc Gauvain. 2006. Continuous space language models for statistical machine translation. In *Proceedings of the 44th Annual Meeting of the International Committee on Computational Linguistics and the Association for Computational Linguistics*, pages 723–730, Sydney, Australia, July.

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1071–1080, Mumbai, India, December.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA, USA, August.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 14–25, Doha, Qatar, October.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, WA, USA, October.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 262–270, Uppsala, Sweden, July.

Joern Wuebker, Matthias Huck, Stephan Peitz, Malte Nuhn, Markus Freitag, Jan-Thorsten Peter, Saab Mansour, and Hermann Ney. 2012. Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation. In *International Conference on Computational Linguistics*, pages 483–491, Mumbai, India, December.

Joern Wuebker, Stephan Peitz, Felix Rietig, and Hermann Ney. 2013. Improving Statistical Machine Translation with Word Class Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1377–1381, Seattle, WA, USA, October.