Similarity-Based Reconstruction Loss for Meaning Representation

Olga Kovaleva, Anna Rumshisky, Alexey Romanov
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
{okovaleva,arum,aromanov}@cs.uml.edu

Abstract

This paper addresses the problem of representation learning. Using an autoencoder framework, we propose and evaluate several loss functions that can be used as an alternative to the commonly used cross-entropy reconstruction loss. The proposed loss functions use similarities between words in the embedding space, and can be used to train any neural model for text generation. We show that the introduced loss functions amplify semantic diversity of reconstructed sentences, while preserving the original meaning of the input. We test the derived autoencoder-generated representations on paraphrase detection and language inference tasks and demonstrate performance improvement compared to the traditional cross-entropy loss.

1 Introduction

Natural language processing (NLP) tasks that use an encoder-decoder architecture tend to rely on the cross-entropy reconstruction loss to generate the target output. A great majority of deep learning models used at present for state-of-the-art machine translation, question answering, summarization, and dialogue generation employ this type of architecture.

The standard cross-entropy loss penalizes the model whenever it fails to produce the exact word from the ground truth data used for training. However, in many NLP tasks that deal with generating text from semantic representation, recovering the exact word is not necessarily optimal, and often generating a near-synonym or just a semantically close word is nearly as good or even better from the point of view of model performance. Consider a situation when a decoder model generates a word by sampling from a softmax over the vocabulary-sized final layer to produce an output. Since cross-entropy loss forces a model to generate the exact words corresponding to those in the input text, the model will be penalized when semantically close but distinct outputs are generated. This is clearly undesirable in many cases when the exact output is not required.

In this paper, we introduce and experiment with a series of distance-based reconstruction losses. Using an auto-encoder derived representation of sentence meaning, we test their impact on model performance in several tasks that require building a semantic representation, including paraphrase detection and entailment / inference. We show that the loss functions that take into account distributional similarity between the word embeddings of the generated output and the ground truth tokens lead to a substantial improvement in performance on such tasks in an unsupervised setting.

2 Related Work

The encoder-decoder setting was first used in deep learning by Sutskever et al. (2014) and has been successfully adapted to a problem of representation learning since then. To date, numerous approaches based on the encoder-decoder idea have been suggested for unsupervised feature extraction from textual data.

Cer et al. (2018) modify the Transformer architecture (Vaswani et al., 2017) originally suggested for machine translation to produce sentence embeddings that target transfer learning to other NLP tasks. Arora et al. (2016) claim that sentence representation as simple weighted averaging of word vectors beats more sophisticated recurrent network-based models. McCann et al. (2017) show that adding machine translation-learned vectors to models designed for other NLP tasks improves their performance. Nangia et al. (2017) in RepEval-2017 report that in-sentence attention and biLSTM-based models extract represen-

tation of meaning from text reasonably well. Logeswaran and Lee (2018) and Kiros et al. (2015) change the problem of learning sentence representations to a classification task for predicting context sentences. Subramanian et al. (2018) demonstrate that sharing the same sentence encoder across different tasks leads to performance improvements.

All the listed works, however, propose methods that either develop task-specific architectures, or use large corpora of labeled data to learn embeddings at a sentence level. Unlike the mentioned papers, the simple modification we propose does not require data annotation and can be used with any state-of-the-art neural models for text generation. Surprisingly, we have not found other work that uses the proposed idea despite its simplicity.

3 Experiments

The objective of a classic autoencoder is to minimize the difference between the given input $\mathbf{X}$ and the reconstructed output $\hat{\mathbf{X}}$.

$$\mathcal{L}(\mathbf{X}, g(f(\mathbf{X})) \tag{1}$$

where f is the encoder function and g is the decoder function. We propose and compare several modifications of distance-based losses, that apply different penalties to the model depending on the similarity of the produced words to the targets in the embedding space.

- *Weighted similarity loss*

$$\mathcal{L} = -\sum_{i=1}^{V} \text{sim}(y_t, y_i) p_i \tag{2}$$

  where p_i is the softmax probability over vocabulary size, $-1 \leq \text{sim} \leq 1$ is the similarity between the tokens embeddings vectors, y_t and y_i are the ground-truth token and the predicted token, respectively, and V is the total vocabulary size. Intuitively, this loss encourages the model to produce high probabilities for words that are close to the target word. In the present experiments, we use cosine as the similarity measure.

- *Weighted cross-entropy loss*

$$\mathcal{L} = -\sum_{i=1}^{V} \text{sim}(y_t, y_i) \log p_i \tag{3}$$

  Here the optimization function can be seen as the "weighted" cross-entropy, meaning that every ground-truth token is represented with similarities to other words in the vocabulary rather than with a traditional one-hot-encoding scheme. The schematic illustration of the true label encoding for the weighted similarity and weighted cross-entropy loss functions is shown in Figure 1 (right).

- *Soft label loss*

$$\mathcal{L} = -\sum_{i=1}^{V} y_i^* log p_i \tag{4}$$

  This cost function is similar to the previous one in terms of true label y_i representation: we encode ground-truth tokens as their similarities across the vocabulary, but we consider only the top N closest words in the vocabulary and normalize the similarities so that they add up to one $\sum_{i=1}^{V} y_i^* = 1$. Essentially, the loss function can be interpreted as cross-entropy with soft targets. We vary N from 3 to 10 in our experiments. We also exclude common English stop-words from soft target encoding, i.e. we apply a regular cross entropy loss for reconstructing of these words. The schematic illustration is given in Figure 1 (center).

$$y_i^* = \begin{cases} \frac{\text{sim}(y_t, y_i)}{\sum_{j=1}^{N} \text{sim}(y_t, y_j)}, & y_i \in \text{top N} \\ 0, & y_i \notin \text{top N} \end{cases} \tag{5}$$

We use pre-trained fastText (Bojanowski et al., 2016) word vectors to compute similarities between words.

Note that the more recently proposed ELMo embeddings (Peters et al., 2018), for example, can not be used in our case, since they are context-dependent, which means that similarities between individual words can not be pre-computed.

To find out how the proposed loss functions affect the quality of the derived representations, we trained several autoencoder models using the regular cross-entropy, as well as the three variants of the similarity-based reconstruction loss described above.

In these experiments, we use the Yelp restaurant reviews dataset (Shen et al., 2017). This dataset was originally introduced for a sentiment classification task and consists of 600K sentences.

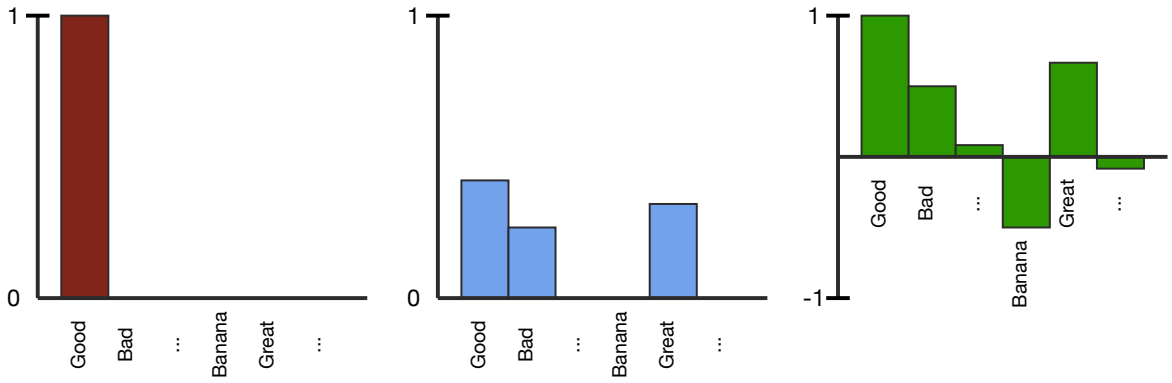

Figure 1: Schematic illustration of true-label encoding using the standard cross-entropy loss (left), soft label loss for $N = 3$ (center) and weighted similarity/weighted cross-entropy loss (right). All the three examples "good", "great" and "bad" are close in the embedding space, since they appear in similar contexts. Note that all the soft labels add up to 1, while weighted similarity labels for the third loss can vary in the range from -1 to 1.

Model	MSRP		SNLI	SICK-E
	F1	Acc	Acc	Acc
Cross-entropy (vanilla AE)	79.0	66.9	44.8	56.8
Soft label, $N = 3$	77.6	67.1	57.8	71.8
Soft label, $N = 5$	79.1	67.3	57.2	71.6
Soft label, $N = 10$	77.9	66.5	57.9	**72.4**
Weighted similarity	77.5	65.6	**69.1**	56.6
Weighted cross-entropy	**79.4**	**68.2**	57.2	70.2

Table 1: Results of transfer learning tasks performance of the proposed autoencoder models. All the models are trained on the Yelp reviews dataset with the use of fastText pre-trained word embeddings.

Our autoencoder model is implemented using the PyTorch deep learning framework (Paszke et al., 2017). In our architecture, both the encoder and the decoder are implemented as single layer LSTMs, each with the hidden size of 256 units. We divide our dataset into train/dev/test splits in 70/10/20 ratio. The resulting vocabulary size of the training dataset is 9.5K tokens. For our training, we use the Adam optimizer (Kingma and Ba, 2014) with the learning rate that varies depending on the tested loss between 0.001 and 0.0001.

We test our learned representations using the SentEval toolkit (Conneau et al., 2017). SentEval is an open-source Python library for evaluating sentence embeddings on a diverse set of language tasks. This toolkit provides a cluster of downstream tasks taken from various competitions such as SemEval as well as a set of probing tasks. In current paper, we focus on the paraphrase detection task using the Microsoft Research Paraphrase Corpus (MSRP) (Dolan et al., 2004), as well as the inference/entailment tasks using the Stanford Natural Language Inference corpus (SNLI) (Bowman et al., 2015) and the SICK-Entailment dataset from SemEval-2014 (Marelli et al., 2014). We selected these tasks because they seem to be likely to benefit from capturing word-level semantic similarity. Table 1 shows the scores averaged over three (3) runs.

4 Discussion

We find that almost all of the proposed loss functions outperform the vanilla autoencoder trained with cross-entropy on all three tasks (see Table 1). The only exception is the weighted similarity loss function. Compared to the logarithm-based losses, this loss applies softer penalties when the ground-truth tokens are predicted to have lower probabilities. We conclude that the non-linearity introduced by a logarithm function contributes to more efficient training.

Among the models we tested, the best scores were achieved by the weighted cross-entropy loss for MSRP (68.2%), the weighted similarity loss for SNLI (69.1%) and by the soft label loss for

Configuration	Autoencoder outputs	
	Input sentence	Reconstructed sentence
Cross-entropy	you can trust this business	you can trust this business
Soft label, $N = 3$	the taste was so good	the flavor was so good
Soft label, $N = 5$	her tone was incredibly rude	her attitude was incredibly unprofessional
Soft label, $N = 10$	a very nice spot for a quiet lunch	a very nice slot for a tranquil lunchtime
Weighted similarity	once again the staff were wonderful	that so that service and great
Weighted cross-entropy	great breakfast option	great food place

Table 2: Sample autoencoder reconstruction outputs for the tested loss configurations.

SICK-E (72.4%). We observe that for the paraphrase task, all the soft label losses behaved similarly, while for the inference/entailment, increasing the number of neighbors improved performance.

In order to better understand how different modifications of the soft label loss affected model performance on transfer tasks, we conducted some additional experiments. Specifically, we investigated the effects of (a) varying the number N of word neighbors used to compute the loss function, and (b) removing the normalization factor by getting rid of the denominator in Eq. 5 (i.e. soft label similarities no longer sum up to 1). Note that when $N = 1$, the soft label loss becomes identical to cross-entropy. When the normalization factor is removed, having $N = V$ makes the soft label loss identical to the weighted similarity loss.

We found that the normalization factor slightly reduced the accuracy for all of the three tasks (see Figure 2). Interestingly, we have not established a universal tendency for the optimal choice of N: for the language inference tasks, the best accuracy was achieved at N close to 10, while for the paraphrasing task the suitable choice for N was in the range of 3-5.

The performance figures obtained for each loss are well illustrated by the quality of the reconstructed examples in Table 2. The standard cross-entropy, as expected, aims at the accurate word-by-word reconstruction of the input sentence. The autoencoder with our least successful weigthed similarity loss function manages to learn most frequent corpus-specific words (e.g. "great service"), but the overall meaning is not conveyed well. he rest of the models succeed in reconstructing synonyms at the word-level. This results in a slightly different expression style (e.g. "her tone was incredibly rude" becomes "her attitude was incredibly unprofessional"), but the overall meaning is reconstructed correctly.

Obviously, the quality of the generated representations depends to a large extent on the selection of pre-trained word embeddings. The related drawback that we observe in our choice of the fastText vectors is that the target ground-truth tokens can be replaced with word inflections as well as with antonyms, which in certain cases can change the meaning of the sentence to the opposite.

For a subset of configurations, we conducted exploratory testing on additional tasks, including different subsets of STS and SICK-Relatedness data. For nearly all tasks tested, we recorded better performance compared to cross-entropy, with the minimum relative gain being 1%. The only performance reduction was in plagiarism detection, which may be expected to favor exact replication.

Although the scores we obtained are below the state-of-the-art for the considered tasks, our goal was to demonstrate that in a traditional encoder-decoder setting, which is extensively used for a number of NLP problems, the proposed loss functions beat the conventional cross-entropy. The major advantage of our proposal is that it is very simple and highly generalizable, i.e. without a sophisticated model architecture, our model is able to produce diversified outputs and can be easily integrated in any existing encoder-decoder architectures.

5 Conclusion

In this paper, we introduced the loss functions that leverage word-level distributional similarity between the generated output and the ground truth. Compared to the representations learned by a vanilla autoencoder, the proposed reconstruction loss variants show substantially improved performance on several semantic representation tasks.

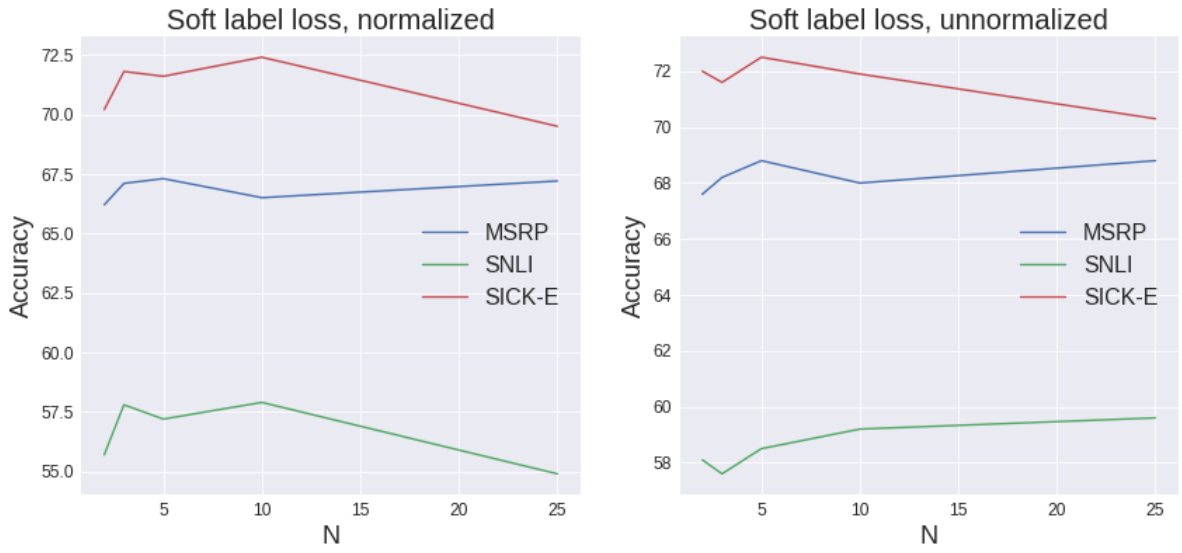

Figure 2: The effect of the soft label loss modifications on task performance. N is the number of closest words neighbors used to compute the loss function.

Further, relative to the conventional cross-entropy, the tested loss variants produce more diverse output while preserving the underlying semantics.

We focused on the autoencoder architecture which requires no pre-annotated data to generate the representations. The major benefit of our proposal is that the proposed loss functions can be plugged directly into any NLP model that generates text word-by-word and that may benefit from more diverse output. The potential applications of our proposal therefore include any of the common NLP problems where language diversity is desirable, including conversational agents, paraphrase generation and text summarization.

The next step for this work is to evaluate the performance of the proposed loss functions in state-of-the-art models for the NLP tasks that leverage sentence-level semantic representation, such as the ones we explored in the present study. Further experiments with the proposed loss functions are also needed to evaluate the effects of different word embedding models on the quality of derived representations.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli,

et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*, pages 216–223.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.