Table 3: Summary statistics for the datasets after tokenization. c denotes the number of target classes.

Data	c	Train	Dev	Test
SNLI	3	549367	9842	9842
MR	2	8529	1067	1066
SST1	5	8544	1101	2210
SST2	2	6920	872	1821
TREC	6	5452	500	500

A More Experiment Details

A.1 Dataset

Here, we detail the using dataset, and the detailed training/dev/test splits are shown on Table 3.

- SNLI (Bowman et al., 2015): a collection of human-written English sentence pairs manually labeled for balanced classification with the labels: entailment, contradiction, and neutral. This is the natural language inference dataset, which is also solved via classification.
- MR v1.0[6]: Movie reviews with one sentence per review labeled positive or negative for sentiment classification.
- SST1[7]: an extension of MR but with fine-grained labels: very positive, positive, neutral, negative, very negative.
- SST2[8]: same as SST1 but with neutral reviews removed and only using positive and negative labels.
- TREC[9]: question samples that classify each question into one of 6 question types: about person, location, numeric information, etc.

A.2 Training Details

(1) Adam optimizer for parameter updating with learning rate of 1e-4; trainable embeddings with size 300.
(2) A MLP with 1 hidden layer as the classifier. For a fair comparison, the hidden unit size is set to 300 for LSTM, CNN, Transformer and Capsule. For our model, it is set to 64 when we use sparse representation to do the prediction and still 300 when we use back transformation representations as the prediction features.[10]
(3) SNLI is the task of identifying the relationships between two given sentences. For each model, we first use it to encode the two sentences into the resulting representations respectively, and then concatenate the two sentence representations for the final prediction.
(4) We report the average accuracy over 10 runs of the experiment on the test data. For each run, the maximum accuracy before early stopping is selected as the result of the current run.

B More analysis about ST

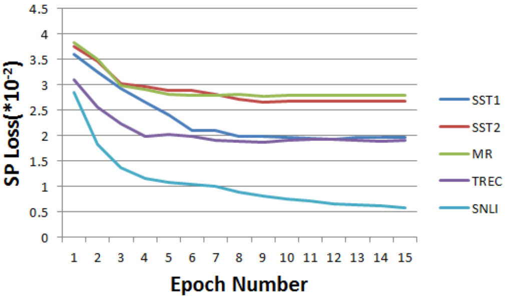

Figure 2: Sparsity evaluation of sparse word representations (the legend is explained below)

Sparsity Analysis: Figure 2 shows the sparsity of the word sparse representations of the 5 datasets. Sparsity is evaluated using the following Sparse Evaluation (SE) function. We proposed this method because previous methods were not designed for sparse representations with both positive and negative values:

$$SE(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}(\sin(\pi \mathbf{y}_i))^2 \qquad (9)$$

As function $(sin(\pi y))^2$ has only three minimum points, -1, 0, 1, it is suitable for measuring the concentration degree of the components of sparse representations. Figure 2 shows a clear decline of SPLoss, which indicates a high concentration degree. Table 4 also gives the statistics about the distributions of the sparse representations. We can see that 'zero' $(V < 0.05)$ takes a large portion of the sparse representations, which is desirable. We can conclude that the learned sparse representations are indeed sparse.

Accuracy of Transformation: We asked a question about the ability of ST to construct LSTM when we introduced the RL^o. Here, we analyze the transformation accuracy of the proposed method

[6] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[7] http://nlp.stanford.edu/sentiment/

[8] http://nlp.stanford.edu/sentiment/

[9] https://cogcomp.seas.upenn.edu/Data/QA/QC/

[10] In detail, the number of parameter of the classifiers for baselines and our model using back transformation representations is 300*300=90,000; while the number for our model using sparse representation is 1000*64=64,000.

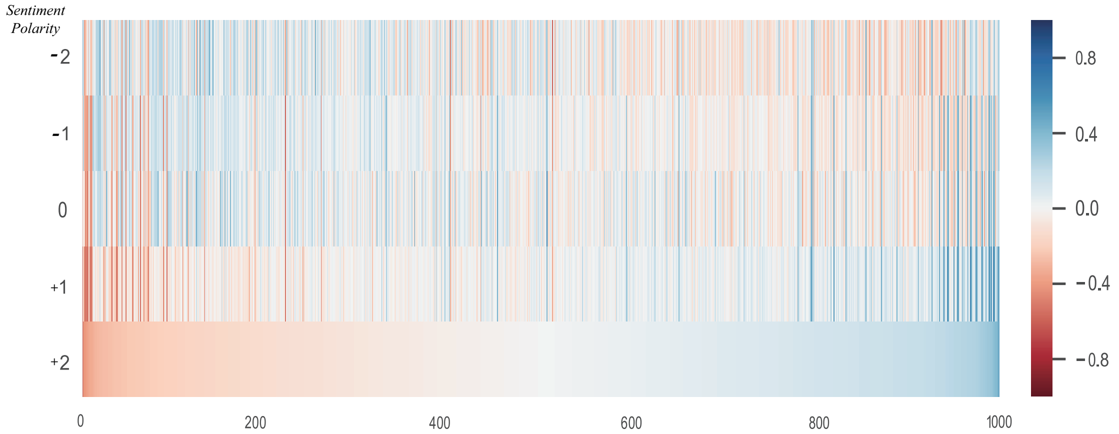

Figure 3: Visualization of learned sparse representations.

Table 4: Distribution of values in the sparse representations (%). $V > 0.6$ ($V < 0.05$) shows the frequency of the values greater (less) than 0.6 (0.05)

Metrics	SNLI	MR	SST1	SST2	TREC
$V > 0.6$	0.14	1.38	1.31	1.71	1.38
$V < 0.05$	99.68	97.39	97.21	96.36	97.16

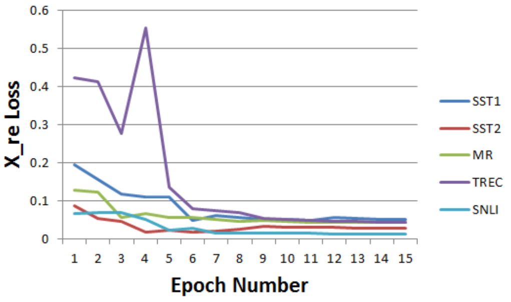

Figure 4: Evaluation of the construction of X.

and give a positive answer to that question. From Table 1, we can see that ST[X'] achieves very similar results to those of LSTM. From the results, we can draw the conclusion that the dense representation generated by ST through backward transformation can achieve very similar results to those of LSTM. Further, we propose a measure to gauge the construction accuracy, named Construction Accuracy Metric (CAM), to evaluate the accuracy of transformation. CAM is formulated as the following function (results are shown in Figure 4):

$$CAM(\mathcal{C}) = \frac{1}{J|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\sum_{j=1}^{J}\frac{|X_{ij} - X'_{ij}|_2^2}{0.5 * |X_{ij}|_2^2 + 0.5 * |X'_{ij}|_2^2} \tag{10}$$

where X_{ij} is the original dense representation of a sub-sentence (generated by LSTM) and X'_{ij} is the backward transformation result of its sparse representation; $\mathcal{C}$ denotes the test set, and J is the length of the sentence. Clearly, this function can evaluate the similarity between X and X' as CAM will raise with the increasing of distance between X and X'. Figure 4 shows that the difference between X and X' is only about 5%. Therefore, we can conclude that our model can construct the outputs of LSTM well.

Interpretability Analysis: Interpretability is one of the most desirable properties of sparse representations. Figure 3 shows the average sparse representation of five classes (tested on the test set of SST1) with different sentiment polarities (-2, -1, 0, 1, 2). Positive numbers refer to positive sentiment, and negative numbers refer to negative sentiment. In order to clearly visualize the differences in the learned representations over the five classes, we sort the bases based on the ascending order of the sparse representation values of +2 (very positive) class.

From Figure 3, we can see that there is a clear color difference for sentiment polarity class +2 and class -2. We can also see a similar phenomenon for sentiment polarity class +1 and class -1 but less pronounced as the their polarities are more similar. These observations demonstrate that the same bases obtain opposite values for classes of opposite sentiments. The bases generating distinct responses for classes with different sentiment polarities can be regarded as primary sentiment bases as they clearly indicate the semantic differences of the classes. In other words, the primary sentiment bases can be explained as sentiment bases. For example, the bases give positive response to positive classes but negative responses to negative classes are the positive sentiment bases, which directly indicate the sentiment polarities.

Comparing with positive and negative classes, neutral class shows relative mixed responses. That means neutral class has similar semantemes to those of both positive and negative classes. This demonstrates that the neutral class is more difficult to identify.

C Future works

Based on this study, many other interesting directions can be pursued in the future, e.g.,:

(1) As we discussed in the paper, the proposed method can construct the output of LSTM well. One future work is to apply ST to language modeling. In this case, the results can be used in many down stream tasks such as machine translation and dialogue systems.

(2) With the help of ST, we can investigate the style transfer on similar tasks in the sparse space by direct semantic reversing. Also, we can use ST to filter out noises or undesirable information.

(3) Based on sparse representations, we can also explore semantic pattern recognition and transformation.