

ARTICLE

Sentence embeddings in NLI with iterative refinement encoders

Aarne Talman*, Anssi Yli-Jyrä and Jörg Tiedemann

Department of Digital Humanities, University of Helsinki, Finland

*Corresponding author. Email: aarne.talman@helsinki.fi

Abstract

Sentence-level representations are necessary for various natural language processing tasks. Recurrent neural networks have proven to be very effective in learning distributed representations and can be trained efficiently on natural language inference tasks. We build on top of one such model and propose a hierarchy of bidirectional LSTM and max pooling layers that implements an iterative refinement strategy and yields state of the art results on the SciTail dataset as well as strong results for Stanford Natural Language Inference and Multi-Genre Natural Language Inference. We can show that the sentence embeddings learned in this way can be utilized in a wide variety of transfer learning tasks, outperforming InferSent on 7 out of 10 and SkipThought on 8 out of 9 SentEval sentence embedding evaluation tasks. Furthermore, our model beats the InferSent model in 8 out of 10 recently published SentEval probing tasks designed to evaluate sentence embeddings' ability to capture some of the important linguistic properties of sentences.

Keywords: natural language inference; sentence representations; representation learning

1. Introduction

Neural networks have been shown to provide a powerful tool for building representations of natural languages on multiple levels of linguistic abstraction. Perhaps the most widely used representations in natural language processing are word embeddings (Mikolov, Sutskever, Chen, *et al.* 2013; Pennington, Socher, and Manning 2014). Recently, there has been a growing interest in models for sentence-level representations using a range of different neural network architectures. Such *sentence embeddings* have been generated using unsupervised learning approaches (Kiros, Zhu, Salakhutdinov, *et al.* 2015; Hill, Cho, and Korhonen 2016) and supervised learning (Bowman, Gauthier, Rastogi, *et al.* 2016; Conneau, Kiela, Schwenk, *et al.* 2017).

Supervision typically comes in the form of an underlying semantic task with labeled data to train the model. The most prominent task for that purpose is natural language inference (NLI) that tries to model the inferential relationship between two or more given sentences. In particular, given two sentences — the premise p and the hypothesis h — the task is to determine whether h is entailed by p , whether the sentences are in contradiction with each other or whether there is no inferential relationship between the sentences (neutral). There are two main neural approaches to NLI. *Sentence encoding-based models* focus on building separate embeddings for the premises and the hypothesis and then combine those using a classifier (Bowman, Angeli, Potts, and Manning 2015; Bowman *et al.* 2016; Conneau *et al.* 2017). Other approaches do not treat the two sentences separately but utilize, for example, cross-sentence attention (Chen, Zhu, Ling, *et al.* 2017a; Tay *et al.* 2018).

With the goal of obtaining general-purpose sentence representations in mind, we opt for the sentence encoding approach. Motivated by the success of the InferSent architecture (Conneau *et al.* 2017), we extend their architecture with a hierarchy-like structure of bidirectional LSTM

(BiLSTM) layers with max pooling. All in all, our model improves the previous state of the art for SciTail (Khot, Sabharwal, and Clark 2018) and achieves strong results for the Stanford Natural Language Inference (SNLI) and Multi-Genre Natural Language Inference corpus (MultiNLI; Williams, Nangia, and Bowman 2018).

In order to demonstrate the semantic abstractions achieved by our approach, we also apply our model to a number of transfer learning tasks using the SentEval testing library (Conneau *et al.* 2017) and show that it outperforms the InferSent model on 7 out of 10 and SkipThought (Kiros *et al.* 2015) on 8 out of 9 tasks, comparing to the scores reported by Conneau *et al.* (2017). Moreover, our model outperforms the InferSent model in 8 out of 10 recently published SentEval probing tasks designed to evaluate sentence embeddings' ability to capture some of the important linguistic properties of sentences (Conneau, Kruszewski, Lample, *et al.* 2018). This highlights the generalization capability of the proposed model, confirming that its architecture is able to learn sentence representations with strong performance across a wide variety of different natural language processing (NLP) tasks.

2. Related Work

There is a wide variety of approaches to sentence-level representations that can be used in NLI. Bowman *et al.* (2015, 2016) explore recurrent neural network (RNN) and long short-term memory (LSTM) architectures, Mou, Men, Li, *et al.* (2016) convolutional neural networks and Vendrov, Kiros, Fidler, *et al.* (2016) GRUs, to name a few. The basic idea behind these approaches is to encode the premise and hypothesis sentences separately and then combine those using a neural network classifier.

Conneau *et al.* (2017) explore multiple different sentence embedding architectures ranging from LSTM, BiLSTM, and intra-attention to convolution neural networks and the performance of these architectures on NLI tasks. They show that, out of these models, BiLSTM with max pooling achieves the strongest results not only in NLI but also in many other NLP tasks requiring sentence-level meaning representations. They also show that their model trained on NLI data achieves strong performance on various transfer learning tasks.

Although sentence embedding approaches have proven their effectiveness in NLI, there are multiple studies showing that treating the hypothesis and premise sentences together and focusing on the relationship between those sentences yields better results (Tay *et al.* 2018; Chen *et al.* 2017a). These methods are focused on the inference relations rather than the internal semantics of the sentences. Therefore, they do not offer similar insights about the sentence-level semantics, as individual sentence embeddings do, and they cannot straightforwardly be used outside of the NLI context.

3. Model Architecture

Our proposed architecture follows a sentence embedding-based approach for NLI introduced by Bowman *et al.* (2015). The model illustrated in Figure 1 contains sentence embeddings for the two input sentences, where the output of the sentence embeddings are combined using a heuristic introduced by Mou *et al.* (2016), putting together the concatenation (u, v) , absolute element-wise difference $|u - v|$, and element-wise product $u * v$. The combined vector is then passed on to a three-layered multi-layer perceptron (MLP) with a three-way softmax classifier. The first two layers of the MLP both utilize dropout and a ReLU activation function.

We use a variant of ReLU called Leaky ReLU (Maas, Hannun, and Ng 2013), defined by:

$$\text{LeakyReLU}(x) = \max(0, x) + y * \min(0, x),$$

where we set $y = 0.01$ as the negative slope for $x < 0$. This prevents the gradient from dying when $x < 0$.

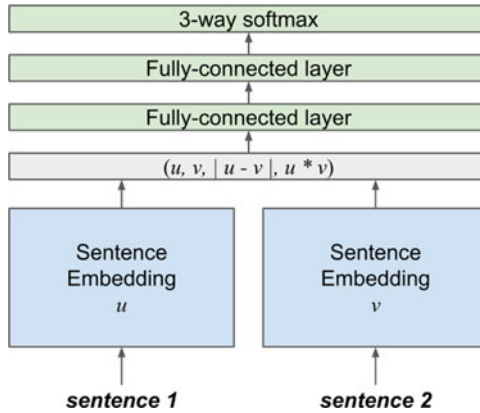


Figure 1. Overall NLI architecture.

For the sentence representations, we first embed the individual words with pre-trained word embeddings. The sequence of the embedded words is then passed on to the sentence encoder which utilizes BiLSTM with max pooling. Given a sequence T of words $(w_1 \dots, w_T)$, the output of the bidirectional LSTM is a set of vectors (h_1, \dots, h_T) , where each $h_t \in (h_1, \dots, h_T)$ is the concatenation

$$h_t = [\vec{h}_t, \overleftarrow{h}_t]$$

of a forward and backward LSTMs

$$\vec{h}_t = \overrightarrow{LSTM}_t(w_1, \dots, w_t)$$

$$\overleftarrow{h}_t = \overleftarrow{LSTM}_t(w_T, \dots, w_t).$$

The max pooling layer produces a vector of the same dimensionality as h_t , returning, for each dimension, its maximum value over the hidden units (h_1, \dots, h_T) .

Motivated by the strong results of the BiLSTM max pooling network by Conneau *et al.* (2017), we experimented with combining BiLSTM max pooling networks in a hierarchy-like structure.^a To improve the BiLSTM layers' ability to remember the input words, we let each layer of the network reread the input embeddings instead of stacking the layers in a strict hierarchical model. In this way, our model acts as an *iterative refinement architecture* that reconsiders the input in each layer while being informed by the previous layer through initialisation. This creates a hierarchy of refinement layers and each of them contributes to the NLI classification by max pooling the hidden states. In the following, we refer to that architecture with the abbreviation HBMP. Max pooling is defined in the standard way of taking the highest value over each dimension of the hidden states and the final sentence embedding is the concatenation of those vectors coming from each BiLSTM layer. The overall architecture is illustrated in Figure 2.

To summarize the differences between our model and traditional stacked BiLSTM architectures, we can list the following three main aspects:

1. Each layer in our model is a separate BiLSTM initialized with the hidden and cell states of the previous layer.
2. Each layer in our model receives the same word embeddings as its input.
3. The final sentence representation is the concatenation of the max pooled output of each layer in the encoder network.

^aConneau *et al.* (2017) explore a similar architecture using convolutional neural networks, called Hierarchical ConvNet.

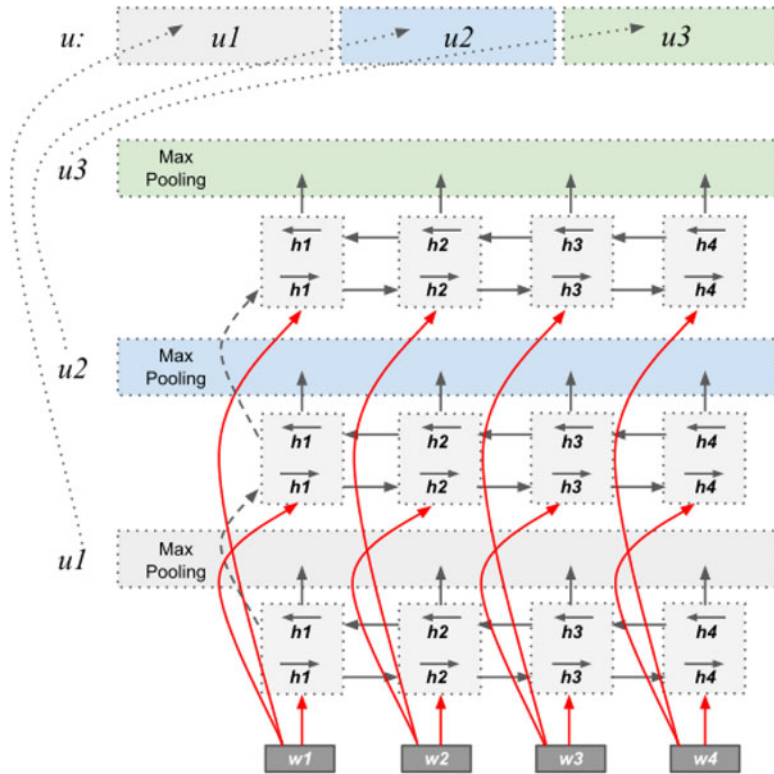


Figure 2. Architecture of the HBMP sentence encoder (where $T = 4$).

In order to study the effect of our architecture, we conduct a comparison of HBMP with the following alternative models:

1. **BiLSTM-Ens:** Ensemble of three BiLSTMs with max pooling, all getting the same embeddings as the input.
2. **BiLSTM-Ens-Train:** Ensemble of three BiLSTMs with max pooling, with the hidden and cell states of each BiLSTM being trainable parameters of the whole network.
3. **BiLSTM-Ens-Tied:** Ensemble of three BiLSTMs with max pooling, where the weights of the BiLSTMs are tied.
4. **BiLSTM-Stack:** A strictly hierarchical model with three BiLSTM layers, where the second and third layers receive the output of the previous layer as their input.

In the first model (BiLSTM-Ens), we contrast our architecture with a similar setup that does not only transfer knowledge between layers but also combines information from three separate BiLSTM layers for the final classification. The second model (BiLSTM-Ens-Train) adds a trainable initialization to each layer to study the impact of the hierarchical initialization that we propose in our architecture. The third model (BiLSTM-Ens-Tied) connects the three layers by tying parameters to each other. Finally, the fourth model (BiLSTM-Stack) implements a standard hierarchical network with stacked layers that do not reread the original input.

We apply the standard SNLI data for the comparison of these different architectures (see Section 5 for more information about the SNLI benchmark). Table 1 lists the results of the experiment.

The results show that HBMP performs better than each of the other models, which supports the use of our setup in favor of alternative architectures. Furthermore, we can see that the different

Table 1. SNLI test accuracies (%) of different architectures

Model	Accuracy	Confidence interval (95%)*
600D HBMP (our model)	86.6	[84.6, 88.7]
600D BiLSTM-Ens	86.3	[84.4, 88.3]
600D BiLSTM-Ens-Train	86.3	[84.3, 88.4]
600D BiLSTM-Ens-Tied	86.1	[83.8, 87.9]
600D BiLSTM-Stack	86.3	[84.2, 88.3]

*Confidence intervals calculated over 1000 random samples of 1000 sentence pairs.

components all contribute to the final score. Ensembling information from three *separate* BiLSTM layers (with independent parameters) improves the performance as we can see in the comparison between BiLSTM-Ens and BiLSTM-Ens-Tied. Trainable initialization does not seem to add to the model's capacity and indicates that the hierarchical initialization that we propose is indeed beneficial. Finally, feeding the same input embeddings to all BiLSTMs of HBMP leads to an improvement over the stacked model that does not reread the input information.

Using these initial findings, we will now look at a more detailed analyses of the performance of HBMP on various datasets and tasks. But before, we first give some more details about the implementation of the model and the training procedures we use. Note that the same specifications also apply to the experiments that we already discussed above.

4. Training Details

The architecture was implemented using PyTorch. We have published our code in GitHub: <https://github.com/Helsinki-NLP/HBMP>.

For all of our models, we used a gradient descent optimization algorithm based on the Adam update rule (Kingma and Ba 2015), which is pre-implemented in PyTorch. We used a learning rate of 5e-4 for all our models. The learning rate was decreased by the factor of 0.2 after each epoch if the model did not improve. We used a batch size of 64. The models were evaluated with the development data after each epoch and training was stopped if the development loss increased for more than three epochs. The model with the highest development accuracy was selected for testing.

We use pre-trained GloVe word embeddings of size 300 dimensions (GloVe 840B 300D; Pennington *et al.* 2014), which were fine-tuned during training. The sentence embeddings have hidden size of 600 for both directions (except for SentEval test, where we test models with 600D and 1200D per direction) and the three-layer MLP have the size of 600 dimensions. We use a dropout of 0.1 between the MLP layers (except just before the final layer). Our models were trained using one NVIDIA Tesla P100 GPU.

5. Evaluation Benchmarks

To further study the performance of HBMP, we train our architecture with three common NLI datasets:

- the SNLI corpus,
- the MultiNLI corpus,
- the Textual Entailment Dataset from Science Question Answering (SciTail).

Note that we treat them as separate tasks and do not mix any of the training, development, and test data in our NLI experiments. We further perform additional linguistic error analyses using

the *MultiNLI Annotation Dataset* and the *Breaking NLI* dataset. Finally, in order to test the ability of the model to learn general-purpose representations, we apply the downstream tasks that are bundled in the *SentEval* package for sentence embedding evaluation. Note that we combine SNLI and MultiNLI data in those experiments in order to be compatible with related work. Below, we provide a few more details about each of the evaluation frameworks.

SNLI: The SNLI corpus (Bowman *et al.* 2015) is a dataset of 570k human-written sentence pairs manually labeled with the gold labels entailment, contradiction, and neutral. The dataset is divided into training (550,152 pairs), development (10,000 pairs), and test sets (10,000 pairs). The source for the premise sentences in SNLI was image captions taken from the Flickr30k corpus (Young, Lai, Hodosh, *et al.* 2014).

MultiNLI: The MultiNLI corpus (Williams *et al.* 2018) is a broad-coverage corpus for NLI consisting of 433k human-written sentence pairs labeled with entailment, contradiction, and neutral. Unlike the SNLI corpus, which draws the premise sentence from image captions, MultiNLI consists of sentence pairs from 10 distinct genres of both written and spoken English. The dataset is divided into training (392,702 pairs), development (20,000 pairs), and test sets (20,000 pairs).

Only five genres are included in the training set. The development and test sets have been divided into matched and mismatched, where the former includes only sentences from the same genres as the training data and the latter includes sentences from the remaining genres not present in the training data.

In addition to the training, development, and test sets, MultiNLI provides a smaller annotation dataset, which contains approximately 1000 sentence pairs annotated with linguistic properties of the sentences and is split between the matched and mismatched datasets.^b This dataset provides a simple way to assess what kind of sentence pairs an NLI system is able to predict correctly and where it makes errors. We use the annotation dataset to perform linguistic error analysis of our model and compare the results to results obtained with *InferSent*. For our experiment with the annotation dataset, we use the annotations for the MultiNLI mismatched dataset.

SciTail: *SciTail* (Khot *et al.* 2018) is an NLI dataset created from multiple-choice science exams consisting of 27k sentence pairs. Each question and the correct answer choice have been converted into an assertive statement to form the hypothesis. The dataset is divided into training (23,596 pairs), development (1,304 pairs), and test sets (2,126 pairs). Unlike the SNLI and MultiNLI datasets, *SciTail* uses only two labels: entailment and neutral.

Breaking NLI: *Breaking NLI* (Glockner, Shwartz, and Goldberg 2018) is a test set (8,193 pairs) which is constructed by taking premises from the SNLI training set and constructing several hypotheses from them by changing at most one word within the premise. It was constructed to highlight how poorly current neural network models for NLI can handle lexical meaning.

SentEval: *SentEval* (Conneau *et al.* 2017; Conneau and Kiela 2018) is a library for evaluating the quality of sentence embeddings.^c It contains 17 downstream tasks as well as 10 probing tasks. The downstream datasets included in the tests were MR movie reviews, CR product reviews, SUBJ subjectivity status, MPQA opinion-polarity, SST binary sentiment analysis, TREC question-type classification, MRPC paraphrase detection, SICK-Relatedness (SICK-R) semantic textual similarity, SICK-Entailment (SICK-E) NLI, and STS14 semantic textual similarity. The probing tasks evaluate how well the sentence encodings are able to capture the following linguistic properties: length prediction, word content analysis, tree depth prediction, top constituents prediction, word order analysis, verb tense prediction, subject number prediction, object number prediction, semantic odd man out, and coordination inversion.

^bThe annotated dataset and description of the annotations are available at http://www.nyu.edu/projects/bowman/multinli/multinli_1.0_annotations.zip

^cThe *SentEval* test suite is available online at <https://github.com/facebookresearch/SentEval>.

Table 2. SNLI test accuracies (%)

Model	Accuracy
BiLSTM Max Pool (InferSent) ^a	84.5
Distance-based Self-Attention ^b	86.3
ReSA ^c	86.3
600D BiLSTM with generalized pooling ^{s,b}	86.6
600D Dynamic Self-Attention Model ^c	86.8
2400D Multiple-Dynamic Self-Attention Model ^c	87.4
600D HBMP (our model)	86.6

Results marked with ^a by Conneau *et al.* (2017), ^b by Chen, Ling, and Zhu (2018), and ^c by Yoon, Lee, and Lee (2018).

For the SentEval tasks, we trained our model on NLI data consisting of the concatenation of the SNLI and MultiNLI training sets consisting of 942,854 sentence pairs in total. This allows us to compare our results to the InferSent results which were obtained using a model trained on the same data (Conneau *et al.* 2017). Conneau *et al.* (2017) have shown that including all the training data from SNLI and MultiNLI improves significantly the model performance on transfer learning tasks, compared to training the model only on SNLI data.

6. Model Performance on the NLI Task

In this section, we discuss the performance of the proposed sentence encoding approach in common NLI benchmarks. From the experiments, we can conclude that the model provides strong results on all of the three NLI datasets. It clearly outperforms the similar but non-hierarchical BiLSTM models reported in the literature and fares well in comparison to other state of the art architectures in the sentence encoding category. In particular, our results are close to the current state of the art on SNLI in this category and strong on both the matched and mismatched test sets of MultiNLI. Finally, on SciTail, we achieve the new state of the art with an accuracy of 86.0%.

Below, we provide additional details on our results for each of the benchmarks. We compare our model only with other state of the art sentence encoding models and exclude cross-sentence attention models, except for SciTail where previous sentence encoding model-based results have not been published.

6.1 SNLI

For the SNLI dataset, our model provides the test accuracy of 86.6% after four epochs of training. The comparison of our results with the previous state of the art and selected other sentence embedding-based results are reported in Table 2.

6.2 MultiNLI

For the MultiNLI matched test set (MultiNLI-m), our model achieves a test accuracy of 73.7% after three epochs of training, which is 0.8% points lower than the state of the art 74.5% by Nie and Bansal (2017). For the mismatched test set (MultiNLI-mm), our model achieves a test accuracy of 73.0% after three epochs of training, which is 0.6% points lower than the state of the art 73.6% by Chen, Zhu, Ling, *et al.* (2017b).

A comparison of our results with the previous state of the art and selected other approaches are reported in Table 3.

Table 3. MultiNLI test accuracies (%)

Model	Accuracy (MultiNLI-m)	Accuracy (MultiNLI-mm)
CBOW ^a	66.2	64.6
BiLSTM ^a	67.5	67.1
BiLSTM + enh embed + max pooling ^b	70.7	70.8
BiLSTM + Inner-attention ^c	72.1	72.1
Deep Gated Attn. BiLSTM encoders ^d	73.5	73.6
Shortcut-Stacked BiLSTM ^e	74.5	73.5
600D HBMP	73.7	73.0

Results marked with ^a are baseline results by Williams *et al.* (2018), ^b by Vu (2017), ^c by Balazs *et al.* (2017), ^d by Chen *et al.* (2017b), and ^e by Nie and Bansal (2017). Our results for the MultiNLI test sets were obtained by submitting the predictions to the respective Kaggle competitions.

Table 4. SciTail test accuracies (%)

Model	Accuracy
DecompAtt ^a	72.3
ESIM ^a	70.6
Ngram ^a	70.6
DGEM w/o edges ^a	70.8
DGEM ^a	77.3
CAFE ^b	83.3
InferSent	85.1
600D HBMP	86.0

Results marked with ^a are baseline results reported by Khot *et al.* (2018) and ^b by Tay *et al.* (2018).

Although we did not achieve state of the art results for the MultiNLI dataset, we believe that a systematic study of different BiLSTM max pooling structures could reveal an architecture providing the needed improvement.

6.3 SciTail

On the SciTail dataset, we compared our model also against non-sentence embedding-based models, as no results have been previously published which are based on independent sentence embeddings. We obtain a score of 86.0% after four epochs of training, which is +2.7% points absolute improvement on the previous published state of the art by Tay *et al.* (2018). Our model also outperforms InferSent which achieves an accuracy of 85.1% in our experiments. The comparison of our results with the previous state of the art results are reported in Table 4.

The results achieved by our proposed model are significantly higher than the previously published results. It has been argued that the lexical similarity of the sentences in SciTail sentence pairs make it a particularly difficult dataset (Khot *et al.* 2018). If this is the case, we hypothesize that our model is indeed better at identifying entailment relations beyond focusing on the lexical similarity of the sentences.

Table 5. Model performance by F-score, comparing HBMP to InferSent (Conneau *et al.* 2017) (our implementation)

	SNLI		MultiNLI-m		MultiNLI-mm		SciTail	
	HBMP	InferSent	HBMP	InferSent	HBMP	InferSent	HBMP	InferSent
Entailment	88.5	86.8	77.2	74.4	77.9	74.9	81.0	81.3
Contradiction	89.1	86.2	75.3	71.8	75.6	71.5	-	-
Neutral	81.9	80.9	68.2	67.1	68.6	65.4	88.9	88.1

Table 6. SNLI confusion matrices for HBMP and InferSent

	Predicted - HBMP				Predicted - InferSent				
	Entail	Contradict	Neutral	Recall (%)	Entail	Contradict	Neutral	Recall (%)	
Gold	Entail	3047	58	263	90.5	2967	95	306	88.1
	Contradict	117	2840	280	87.7	154	2756	327	85.1
	Neutral	357	240	2622	81.5	346	302	2571	79.9
	<i>Precision (%)</i>	86.5	90.5	82.8	85.6	87.4	80.2		

7. Error Analysis of NLI Predictions

To better understand what kind of inferential relationships our model is able to identify, we conducted an error analysis for the three datasets. We report the results below.

Table 5 shows the accuracy of predictions per label (in terms of F-scores) for the HBMP model and compares them to the InferSent model. This analysis shows that our model leads to a significant improvement over the outcome of the non-hierarchical model from previous work in almost all categories on all the three benchmarks. The only exception is the entailment score on SciTail, which is slightly below the performance of InferSent.

To see in more detail how our HBMP model is able to classify sentence pairs with different labels and what kind of errors it makes, we summarize error statistics as confusion matrices for the different datasets. They highlight the HBMP model’s strong performance across all the labels.

On the **SNLI dataset**, our model clearly outperforms InferSent on all labels in terms of precision and recall. Table 6 contains the confusion matrices for that dataset comparing HBMP to InferSent. The precision on contradiction exceeds 90% for our model and reaches high recall values for both entailment and contradiction. The performance is lower for neutral and the confusion of that label with both contradiction and entailment is higher. However, HBMP still outperforms InferSent by a similar margin as for the other two labels.

Unlike for the SNLI and both of the MultiNLI datasets, on the **SciTail dataset**, our model is most accurate on sentence pairs labeled neutral, having an F-score of 88.9% compared to pairs marked with entailment, where the F-score was 81.0%. InferSent has slightly higher accuracy on entailment, whereas HBMP outperforms InferSent on neutral. Table 7 contains the confusion matrices for the SciTail dataset comparing the HBMP to InferSent. This analysis reveals that our model mainly suffers in recall on entailment detection, whereas it performs well for neutral with respect to recall. It is difficult to say what the reason might be for the mismatch between the two systems, but the overall performance of our architecture suggests that it is superior to the InferSent model even though the balance between precision and recall on individual labels is different.

The error analysis of the **MultiNLI dataset** is not standard as it cannot be based on test data. As the labeled test data are not openly available for MultiNLI, we analyzed the error statistics for this dataset based on the development data.

Table 7. SciTail confusion matrices for HBMP and InferSent based on the development set

		Predicted - HBMP			Predicted - InferSent		
		Entail	Neutral	Recall (%)	Entail	Neutral	Recall (%)
Gold	Entail	632	210	75.0	673	169	79.9
	Neutral	88	1196	93.1	140	1144	89.1
<i>Precision (%)</i>		88.0	85.0		82.8	87.1	

Table 8. MultiNLI-matched confusion matrices for HBMP and InferSent based on the development set

		Predicted - HBMP				Predicted - InferSent			
		Entail	Contradict	Neutral	Recall (%)	Entail	Contradict	Neutral	Recall (%)
Gold	Entail	2781	196	486	80.3	2614	278	587	75.1
	Contradict	372	2354	514	72.7	449	2241	523	69.7
	Neutral	528	443	2158	69.0	477	507	2139	68.5
<i>Precision (%)</i>		75.6	78.7	68.3		73.8	74.1	65.8	

Table 9. MultiNLI-mismatched confusion matrices for HBMP and InferSent

		Predicted - HBMP				Predicted - InferSent			
		Entail	Contradict	Neutral	Recall (%)	Entail	Contradict	Neutral	Recall (%)
Gold	Entail	2841	163	459	82.0	2731	246	486	78.9
	Contradict	438	2279	523	70.3	491	2226	523	68.7
	Neutral	613	387	3111	68.0	611	510	2008	64.2
<i>Precision (%)</i>		73.0	81.0	68.4		71.2	74.6	66.6	

For the matched dataset (MultiNLI-m), our model had a development accuracy of 74.1%. For MultiNLI-m our model has the best accuracy on sentence pairs labeled with entailment, having an F-score of 77.2%. The model is also almost as accurate in predicting contradictions, with an F-score of 75.3%. Similar to SNLI, our model is less effective on sentence pairs labeled with neutral, having an F-score of 68.2% but, again, the HBMP model outperforms the InferSent on all the labels. Table 8 contains the confusion matrices for the MultiNLI matched dataset comparing the HBMP to InferSent. Our model improves upon InferSent in all values of precision and recall, in some cases by a wide margin.

For the MultiNLI mismatched dataset (MultiNLI-mm), our model had a development accuracy of 73.7%. For MultiNLI-mm our model has very similar performance as with the MultiNLI-m dataset, having the best accuracy on sentence pairs labeled with entailment, having an F-score of 77.9%. The model is also almost as accurate in predicting contradictions, with an F-score of 75.6%. Our model is less effective on sentence pairs labeled with neutral, having an F-score of 68.6%. Table 9 contains the confusion matrices for the MultiNLI Mismatched dataset comparing the HBMP to InferSent and the picture is similar to the result of the matched dataset. Substantial improvements can be seen again, in particular, in the precision of contradiction detection.

Table 10. MultiNLI-mm linguistic error analysis (accuracy %), comparing our HBMP results to the InferSent Conneau *et al.* (2017) results (our implementation). Number of sentence pairs with the linguistic label in brackets after the label name

	Entailment		Contradiction		Neutral	
	HBMP	InferSent	HBMP	InferSent	HBMP	InferSent
Active/passive (10)	100.0	100.0	100.0	100.0	-	-
Anto (16)	-	-	76.9	69.2	85.7	71.4
Belief (44)	88.2	82.4	66.7	61.1	73.9	78.3
Conditional (16)	81.8	81.8	37.5	62.5	57.1	71.4
Coref (22)	75.0	75.0	71.4	64.3	81.8	81.8
Long sentence (77)	80.6	77.4	58.3	61.1	73.8	71.4
Modal (98)	80.9	78.7	68.6	68.6	81.8	70.5
Negation (78)	76.0	64.0	81.8	76.4	58.3	45.8
Paraphrase (33)	89.2	86.5	-	-	-	-
Quantifier (104)	75.0	72.5	73.1	73.1	75.0	77.1
Quantity/time (15)	33.3	50.0	41.7	41.7	33.3	33.3
Tense difference (14)	100.0	75.0	0.0	0.0	83.3	75.0
Word overlap (26)	90.5	95.2	41.7	41.7	50.0	0.0
Total	80.6	83.0	66.3	68.5	73.8	72.7

8. Evaluation of Linguistic Abstractions

The most interesting part of the sentence encoder approach to NLI is the ability of the system to learn generic sentence embeddings that capture abstractions, which can be useful for other downstream tasks as well. In order to understand the capabilities of our model, we first look at the type of linguistic reasoning that the NLI system is able to learn using the MultiNLI annotation set and the Breaking NLI test set. Thereafter, we evaluate downstream tasks using the SentEval library to study the use of our NLI-based sentence embeddings in transfer learning.

8.1 Linguistic error analysis of NLI classifications

The MultiNLI annotation set makes it possible to conduct a detailed analysis of different linguistic phenomena when predicting inferential relationships. We use this to compare our model to InferSent with respect to the type of linguistic properties that are present in the given sentence pairs. Table 10 contains the comparison for the MultiNLI-mm dataset. The analysis shows that our HBMP model outperforms InferSent with antonyms, coreference links, modality, negation, paraphrases, and tense differences. It also produces improved scores for most of the other categories in entailment detection. InferSent gains especially with conditionals in contradiction and in the word overlap category for entailments. This seems to suggest that InferSent relies a lot on matching words to find entailment and specific constructions indicating contradictions. HBMP does not seem to use word overlap as an indication for entailment that much and is better on detecting neutral sentences in this category. This outcome may indicate that our model works with stronger lexical abstractions than InferSent. However, due to the small number of examples per annotation category and small differences in the scores in general, it is hard to draw reliable conclusions from this experiment.

Table 11. Breaking NLI scores (accuracy %)

Category	Decomp. Attn*	ESIM*	WordNet Baseline*	KIM*	Infer-Sent	600D HBMP
Antonyms	41.6	70.4	95.5	86.5	51.6	54.7
Antonyms(wordnet)	55.1	74.6	94.5	78.8	63.7	69.1
Cardinals	53.5	75.5	98.6	93.4	49.4	58.8
Colors	85.0	96.1	98.7	98.3	90.6	90.4
Countries	15.2	25.4	100.0	70.8	77.2	81.2
Drinks	52.9	63.7	94.8	96.6	85.1	81.3
Instruments	96.9	90.8	67.7	96.9	98.5	96.9
Materials	65.2	89.7	75.3	98.7	81.6	82.6
Nationalities	37.5	35.9	78.5	73.5	47.3	49.8
Ordinals	2.1	21.0	40.7	56.6	7.4	4.5
Planets	31.7	3.3	100.0	5.0	75.0	45.0
Rooms	59.2	69.4	89.9	77.6	76.3	72.1
Synonyms	97.5	99.7	70.5	92.1	99.6	84.5
Vegetables	43.1	31.2	86.2	79.8	39.5	40.4
Total	51.9	65.6	85.8	83.5	65.6	65.1

Results marked with * as reported by Glockner *et al.* (2018). InferSent results obtained with our implementation using the training set up described in (Conneau *et al.* 2017). Scores highlighted with bold are top scores when comparing the InferSent and our HBMP model.

8.2 Tests with the breaking NLI dataset

In the second experiment, we conducted testing of the proposed sentence embedding architecture using the **Breaking NLI test set** recently published by Glockner *et al.* (2018). The test set is designed to highlight the lack of lexical reasoning capability of NLI systems.

For the Breaking NLI experiment, we trained our HBMP model and the InferSent model using the SNLI training data. We compare our results with the results published by Glockner *et al.* (2018) and to results obtained with InferSent sentence encoder (our implementation).

The results show that our HBMP model outperforms the InferSent model in 7 out of 14 categories, receiving an overall score of 65.1% (InferSent: 65.6%). Our model is especially strong with handling antonyms, which shows a good level of semantic abstraction on the lexical level. InferSent fares well in narrow categories like drinks, instruments, and planets, which may indicate a problem of overfitting to prominent examples in the training data. The strong result on the synonyms class may also come from a significant representation of related examples in training. However, more detailed investigations are necessary to verify this hypothesis.

Our model also compares well against the other models, outperforming Decomposable Attention model (51.90%) (Parikh, Täckström, Das, *et al.* 2016) and Residual Encoders (62.20%) (Nie and Bansal 2017) in the overall score. As these models are not based purely on sentence embeddings, the obtained result highlights that sentence embedding approaches can be competitive when handling inferences requiring lexical information. The results of the comparison are summarized in Table 11.

8.3 Transfer learning

In this section, we focus on transfer learning experiments that apply sentence embeddings trained on NLI to other downstream tasks. In order to better understand how well the sentence encoding

Table 12. Transfer learning test results for the HBMP model on a number of SentEval downstream sentence embedding evaluation tasks

Task	InferSent	SkipThought	600D HBMP	1200D HBMP
MR	81.1	79.4	81.5	81.7
CR	86.3	83.1	86.4	87.0
SUBJ	92.4	93.7	92.7	93.7
MPQA	90.2	89.3	89.8	90.3
SST	84.6	82.9	83.6	84.0
TREC	88.2	88.4	86.4	88.8
MRPC	76.2/83.1	-	74.6/82.0	76.7/83.4
SICK-R	0.884	0.858	0.876	0.876
SICK-E	86.3	79.5	85.3	84.7
STS14	0.70/0.67	0.44/0.45	0.70/0.66	0.71/0.68

InferSent and SkipThought results as reported by Conneau *et al.* (2017). To remain consistent with other work using SentEval, we report the accuracies as they are provided by the SentEval library.

Table 13. SentEval probing task results (accuracy %)

Task	InferSent	600D HBMP	1200D HBMP
SentLen	71.7	75.9	75.0
WC	87.3	84.1	85.3
TreeDepth	41.6	42.9	43.8
TopConst	70.5	76.6	77.2
BShift	65.1	64.3	65.6
Tense	86.7	86.2	88.0
SubjNum	80.7	83.7	87.0
ObjNum	80.3	79.3	81.8
SOMO	62.1	58.9	59.0
CoordInv	66.8	68.5	70.8

InferSent results are BiLSTM Max (NLI) results as reported by Conneau *et al.* (2018).

model generalizes to different tasks, we conducted various tests implemented in the SentEval sentence embedding evaluation library (Conneau *et al.* 2017) and compared our results to the results published for InferSent and SkipThought (Kiros *et al.* 2015).

We used the SentEval library with the default settings recommended on their website, with a logistic regression classifier, Adam optimizer with learning rate of 0.001, batch size of 64, and epoch size of 4. Table 12 lists the transfer learning results for our models with 600D and 1200D hidden dimensionality and compares it to the InferSent and SkipThought scores reported by Conneau *et al.* (2017). Our 1200D model outperforms the InferSent model on 7 out of 10 tasks. The model achieves higher score on 8 out of 9 tasks reported for SkipThought, having equal score on the SUBJ dataset. No MRPC results have been reported for SkipThought.

To study in more detail the linguistic properties of our proposed model, we also ran the recently published SentEval probing tasks (Conneau *et al.* 2018). Our 1200D model outperforms the InferSent model in 8 out of 10 probing tasks. The results are listed in Table 13.

Looking at both the downstream and the probing tasks, we can observe strong results of our model compared to the InferSent model that already demonstrated good general abstractions on the sentence level according to the original publication by Conneau *et al.* (2017). Hence, HBMP does not only provide competitive NLI scores but also produces improved sentence embeddings that are useful for other tasks.

9. Conclusion

In this paper, we have introduced an iterative refinement architecture (HBMP) based on BiLSTM layers with max pooling that achieves a new state of the art for SciTail and strong results in the SNLI and MultiNLI sentence encoding category. We carefully analyzed the performance of our model with respect to the label categories and the errors it produces in the various NLI benchmarks. We demonstrate that our model outperforms InferSent in nearly all cases with substantially reduced confusion between classes of inferential relationships. The linguistic analysis on MultiNLI also reveals that our approach is robust across the various categories and outperforms InferSent on, for example, antonyms and negations that require a good level of semantic abstraction.

Furthermore, we tested our model using the SentEval sentence embedding evaluation library, showing that it achieves great generalization capability. The model outperforms InferSent on 7 out of 10 downstream and 8 out of 10 probing tasks and SkipThought on 8 out of 9 downstream tasks. Overall, our model performs well across all the conducted experiments, which highlights its applicability for various NLP tasks and further demonstrates the general abstractions that it is able to pick up from the NLI training data.

Although the neural network approaches to NLI have been hugely successful, there has also been a number of concerns raised about the quality of current NLI datasets. Gururangan, Swayamdipta, Levy, *et al.* (2018) and Poliak, Naradowsky, Haldar, *et al.* (2018) show that datasets like SNLI and MultiNLI contain annotation artifacts which help neural network models in classification, allowing decisions only based on the hypothesis sentences as their input. On a theoretical and methodological level, there is an on-going discussion on the nature of various NLI datasets, as well as the definition of what counts as NLI and what does not. For example, Chatzikyriakidis, Cooper, Dobnik, *et al.* (2017) present an overview of the most standard datasets for NLI and show that the definitions of inference in each of them are actually quite different. Talman and Chatzikyriakidis (2019) further highlight this by testing different state of the art neural network models by training them on one dataset and then testing on another, leading to a significant drop in performance for all models.

In addition to the concerns related to the quality of NLI datasets, the success of the proposed architecture raises a number of other interesting questions. First of all, it would be important to understand what kind of semantic information the different layers are able to capture and how they differ from each other. Secondly, we would like to ask whether other architecture configurations could lead to even stronger results in NLI and other downstream tasks. A third question is concerned with other languages and cross-lingual settings. Does the result carry over to multilingual setups and applications? The final question is whether NLI-based sentence embeddings could successfully be combined with other supervised and also unsupervised ways of learning sentence-level representations. We will look at all those questions in our future work.

Author ORCIDs.  Aarne Talman, [0000-0002-3573-5993](https://orcid.org/0000-0002-3573-5993)

Acknowledgments. The work in this paper was supported by the Academy of Finland through project 314062 from the ICT 2023 call on Computation, Machine Learning and Artificial Intelligence, and through projects 270354/273457/313478. This project has also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 771113). We would also like to acknowledge NVIDIA and their GPU grant.



References

- Balazs J., Marrese-Taylor E., Loyola P. and Matsuo Y.** (2017). Refining raw sentence representations for textual entailment recognition via attention. In *Workshop on Evaluating Vector Space Representations for NLP*. ACL. pp. 51–55.
- Bowman S.R., Angeli G., Potts C. and Manning C.D.** (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing EMNLP*. Association for Computational Linguistics. pp. 632–642.
- Bowman S.R., Gauthier J., Rastogi A., Gupta R., Manning C.D. and Potts C.** (2016). A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. pp. 1466–1477
- Chatzikyriakidis S., Cooper R., Dobnik S. and Larsson S.** (2017). An overview of natural language inference data collection: The way forward? In *Proceedings of the Computing Natural Language Inference Workshop*.
- Chen Q., Ling Z.-H. and Zhu X.** (2018). Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics. pp. 1815–1826.
- Chen Q., Zhu X., Ling Z.-H., Wei S., Jiang H. and Inkpen D.** (2017a). Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. pp. 1657–1668.
- Chen Q., Zhu X., Ling Z.-H., Wei S., Jiang H. and Inkpen D.** (2017b). Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics. pp. 36–40. ACL.
- Conneau A. and Kiela D.** (2018). SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the 11th Language Resources and Evaluation Conference*. European Language Resource Association. Miyazaki, Japan: Phoenix Seagaia Conference Center. pp. 1699–1704.
- Conneau A., Kiela D., Schwenk H., Barrault L. and Bordes A.** (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. pp. 670–680.
- Conneau A., Kruszewski G., Lample G., Barrault L. and Baroni M.** (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. pp. 2126–2136.
- Glockner M., Shwartz V. and Goldberg Y.** (2018). Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics. pp. 650–655.
- Gururangan S., Swamydipta S., Levy O., Schwartz R., Bowman S. and Smith N.A.** (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics. pp. 107–112.
- Hill F., Cho K. and Korhonen A.** (2016). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. pp. 11367–1377.
- Khot T., Sabharwal A. and Clark P.** (2018). Scitail: A textual entailment dataset from science question answering. In *AAAI Conference on Artificial Intelligence*.
- Kingma D.P. and Ba J.** (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
- Kiros R., Zhu Y., Salakhutdinov R., Zemel R.S., Urtasun R., Torralba A. and Fidler S.** (2015). Skip-thought vectors. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc. pp. 3294–3302.
- Maas A.L., Hannun A.Y. and Ng A.Y.** (2013). Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*.
- Mikolov T., Sutskever I., Chen K., Corrado G. and Dean J.** (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, USA: Curran Associates, Inc. pp. 3111–3119.
- Mou L., Men R., Li G., Xu Y., Zhang L., Yan R. and Jin Z.** (2016). Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics. pp. 130–136.
- Nie Y. and Bansal M.** (2017). Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics. pp. 41–45.
- Parikh A.P., Täckström O., Das D. and Uszkoreit J.** (2016). A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. pp. 2249–2255.
- Pennington J., Socher R. and Manning C.D.** (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. pp. 1532–1543.

- Poliak A., Naradowsky J., Haldar A., Rudinger R. and Van Durme B.** (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics. pp. 180–191.
- Talman A. and Chatzikyriakidis S.** (2019). Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. [arXiv:1810.09774](https://arxiv.org/abs/1810.09774).
- Tay Y., Tuan L.A. and Hui S.C.** (2018). Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. pp. 1565–1575.
- Vendrov I., Kiros R., Fidler S. and Urtasun R.** (2016). Order-embeddings of images and language. In *6th International Conference on Learning Representations*.
- Vu H.** (2017). Lct-malta's submission to repeval 2017 shared task. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics. pp. 56–60.
- Williams A., Nangia N. and Bowman S.R.** (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics. pp. 1112–1122.
- Yoon D., Lee D. and Lee S.** (2018). Dynamic Self-Attention : Computing Attention over Words Dynamically for Sentence Embedding. [arXiv:1808.07383](https://arxiv.org/abs/1808.07383).
- Young P., Lai A., Hodosh M. and Hockenmaier J.** (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Transactions of the Association for Computational Linguistics (TACL) 2*, pp. 67–78.