Supervised Learning of Universal Sentence Representations from Natural Language Inference Data

발제자: 16기 노시영

Introduction

- Word embedding에 대한 실용성은 어느정도 받아들여져 있는 상황이다.
- 다만, 한 문장의 전체적인 의미, 즉 단어와 단어간의 상호작용을 하나의 벡터로 표현하기에는 한계가 있는 상태이다.
- 문장을 표현해 줄 수 있는 범용적인 과제가 남아있다(Universal representations of sentences)
- 연구를 통해 supervised learning이 unsupervised learning보다 결과가 좋게 나온다는 것이 입증되었다

Related Work

- Computer Vision에서 supervised features을 활용하여 transfer learning을 실현시키는데 성공적이다.

*transfer learning: 기존의 만들어진 모델을 사용하여 새로운 모델을 만들시 학습을 빠르게 하며, 예측을 더 높이는 방법입니다.

- 기존 NLP모델들은 Unsupervised sentence representation learning을 활용했다.
- SkipThought 모델은 Unsupervised learning을 활용하여 지금까지 가장 성공적인 결과를 내놓았다. 이를 비교하기 위해서 supervised training을 활용했다.
- SNLI corpus에 문장 encoder를 학습시켜서 SICK corpus에 적용하는 연구도 했으나, 더욱 단순한 방법으로 unsupervised training한 것보다 결과가 좋진 않았다.

Approach

- SNLI task을 활용하여 universal sentence encoding 모델들을 학습시킬 수 있는 방법에 대한 설명
- Sentence encoder로 활용된 모델들에 대한 설명

(LSTM, GRU, BiLSTM with mean/max pooling, Self-attentive network)

*SNLI dataset (Stanford Natural Language Inference)

- 570k 개의 문장으로 구성된 dataset으로서 사람이 손수 labelling을 함. 크게 세가지 문장의 항목: entailment(포함관계), neutral, contradictory

Ex) “Two women are embracing while holding to go packages”

“Two women are holding packages”=>entailment

“A man is typing on a machine used for stenography”

“The man isn’t operating a stenograph”=>contradiction

문장의 의미론적 이해를 위한 유용한 데이터셋

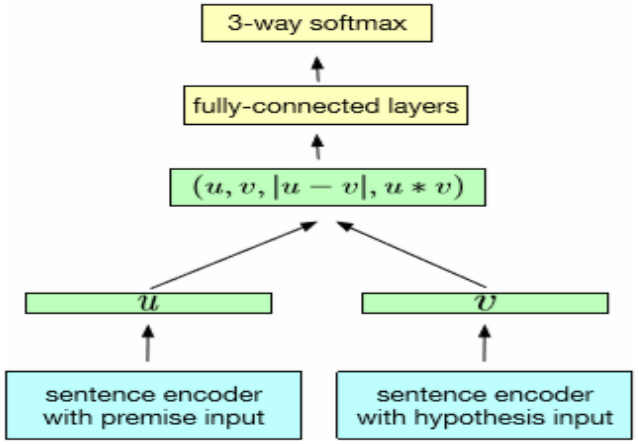


Figure 1: **Generic NLI training scheme.**

LSTM & GRU

h_t=$\overrightarrow{LSTM}$ or h_t=$\overrightarrow{GRU}$(w_1, w_2,..., w_t,)

T개 단어의 문장에서 t개의 hidden representations (h_1, h_2,..., h_T)을 생성한다.

BiLSTM with max/mean pooling

T개 단어의 문장에서 t개의 hidden representations (h_1, h_2,..., h_T)을 생성한다. 하지만 문장의 처음부터 끝까지 한번, 끝에서 처음까지 한번 계산하여 이를 concatenate하여 hidden representation을 생성한다.

그후, 각 hidden unit의 차원에서 최댓값을 취합하거나(max pooling)

평균(mean pooling)을 내서 고정된 벡터를 만들어준다

$$
\begin{aligned}
\overrightarrow{h_t} &= \overrightarrow{\mathrm{LSTM}_t}(w_1, \ldots, w_T) \\
\overleftarrow{h_t} &= \overleftarrow{\mathrm{LSTM}_t}(w_1, \ldots, w_T) \\
h_t &= [\overrightarrow{h_t}, \overleftarrow{h_t}]
\end{aligned}
$$

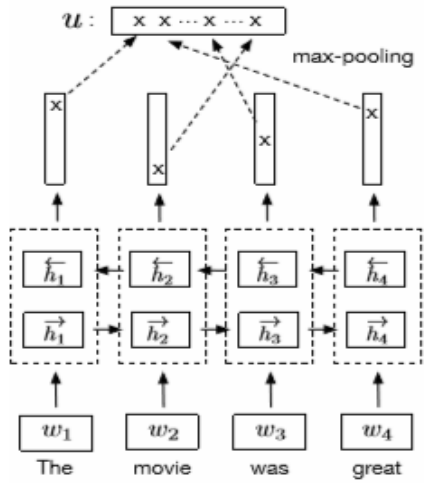


Figure 2: **Bi-LSTM max-pooling network.**

Self-attentive Network

Self-attentive sentence encode는 attention mechanism을 BiLSTM의 hidden states에 적용하여 입력 문장을 벡터 u의 형태로 만들어준다. Attention mechanism은 다음과 같다.

*h_i=BiLSTM에서 얻은 각 hidden state

$$\begin{aligned} \bar{h}_i &= \tanh(W h_i + b_w) \\ \alpha_i &= \frac{e^{\bar{h}_i^T u_w}}{\sum_i e^{\bar{h}_i^T u_w}} \\ u &= \sum_t \alpha_i h_i \end{aligned}$$

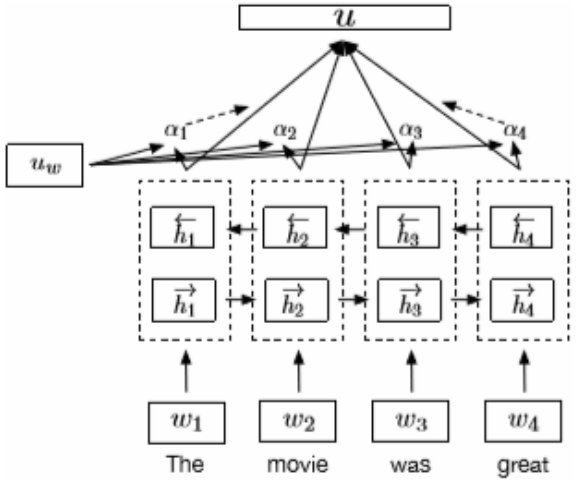
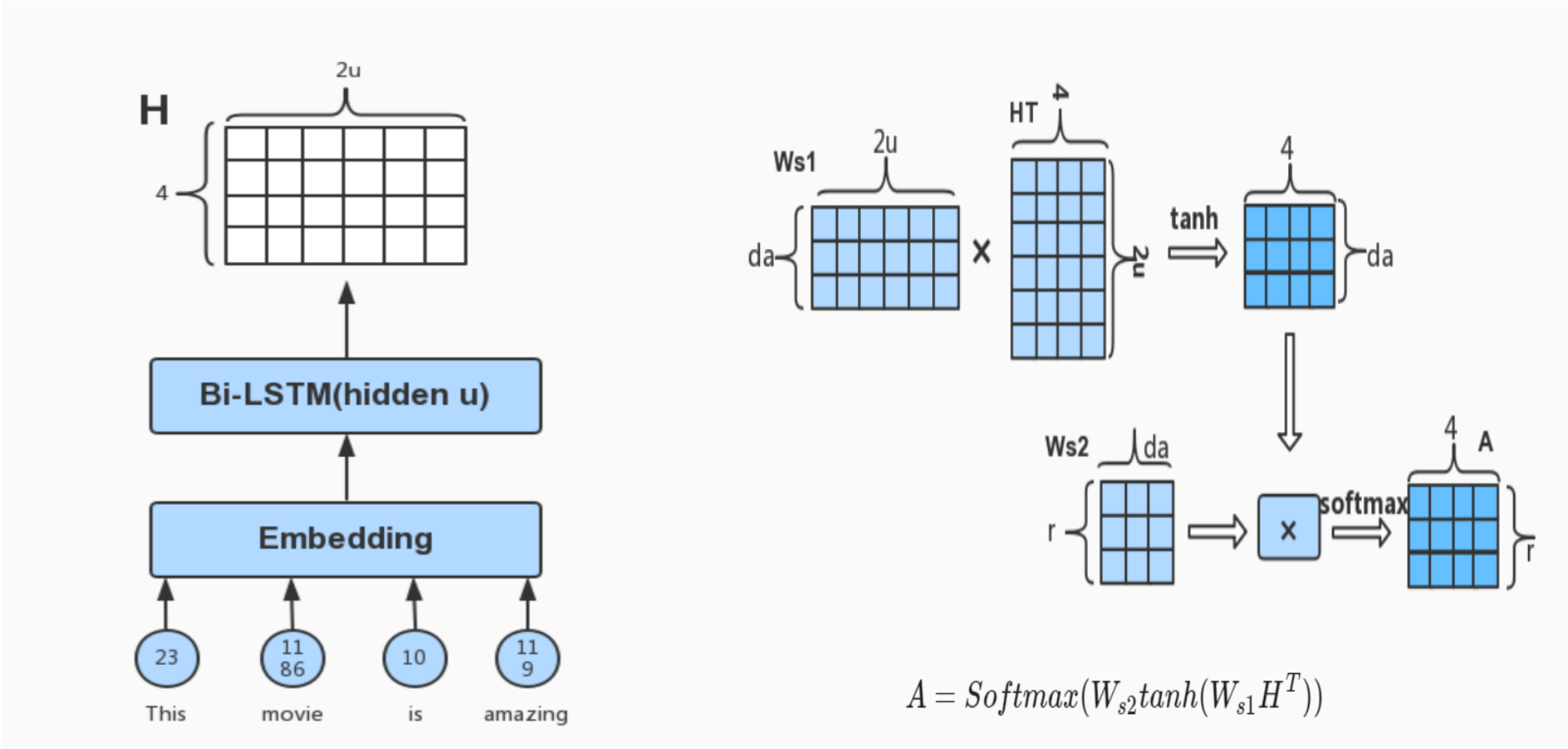


Figure 3: **Inner Attention network architecture.**

Self-attentive Network

M=AH



Heirarchical ConvNet(CNN)

AdaSent에서 아이디어 착안

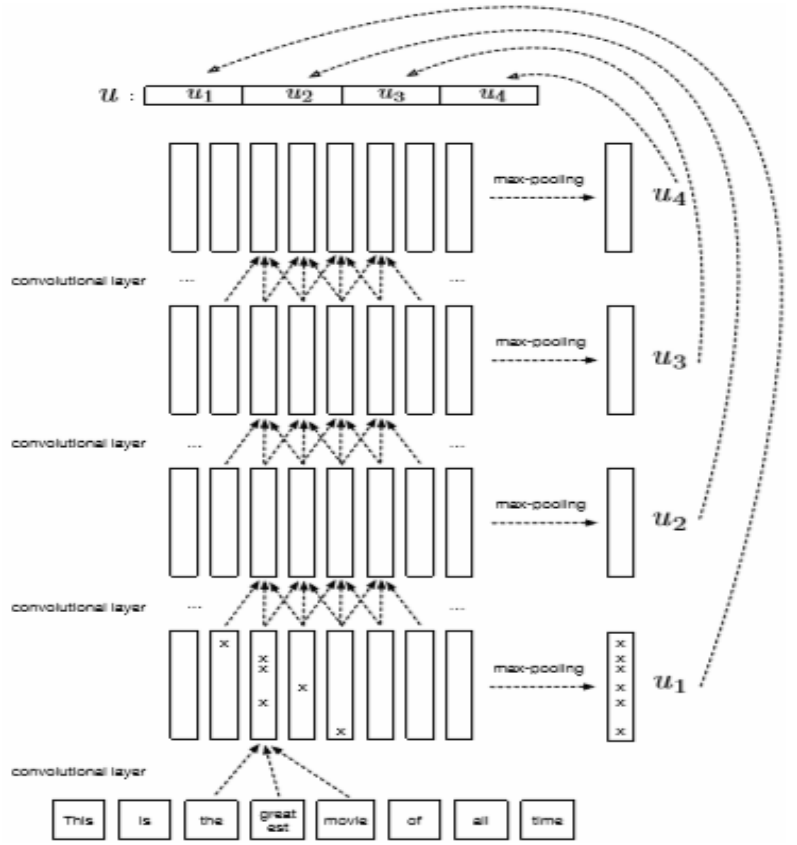
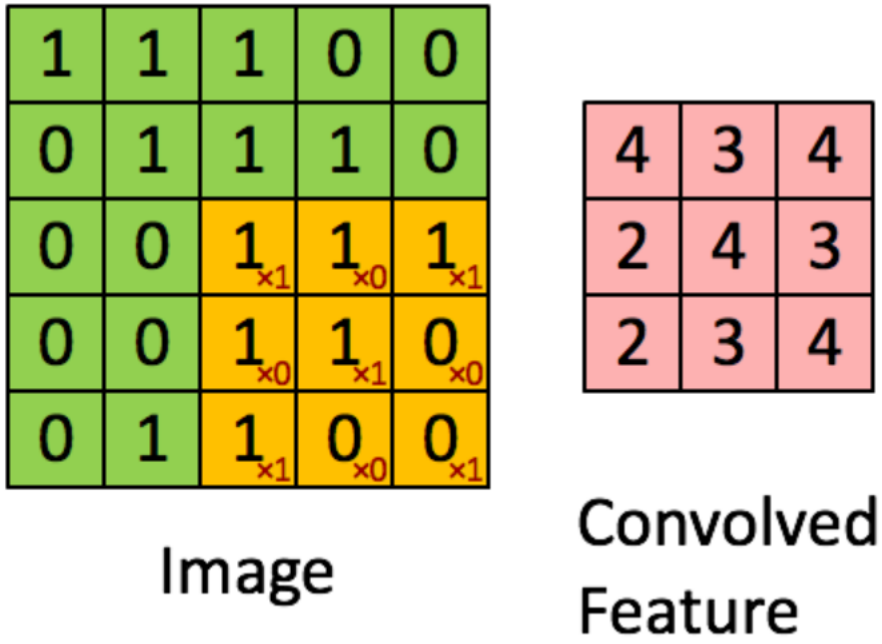


Figure 4: **Hierarchical ConvNet architecture.**

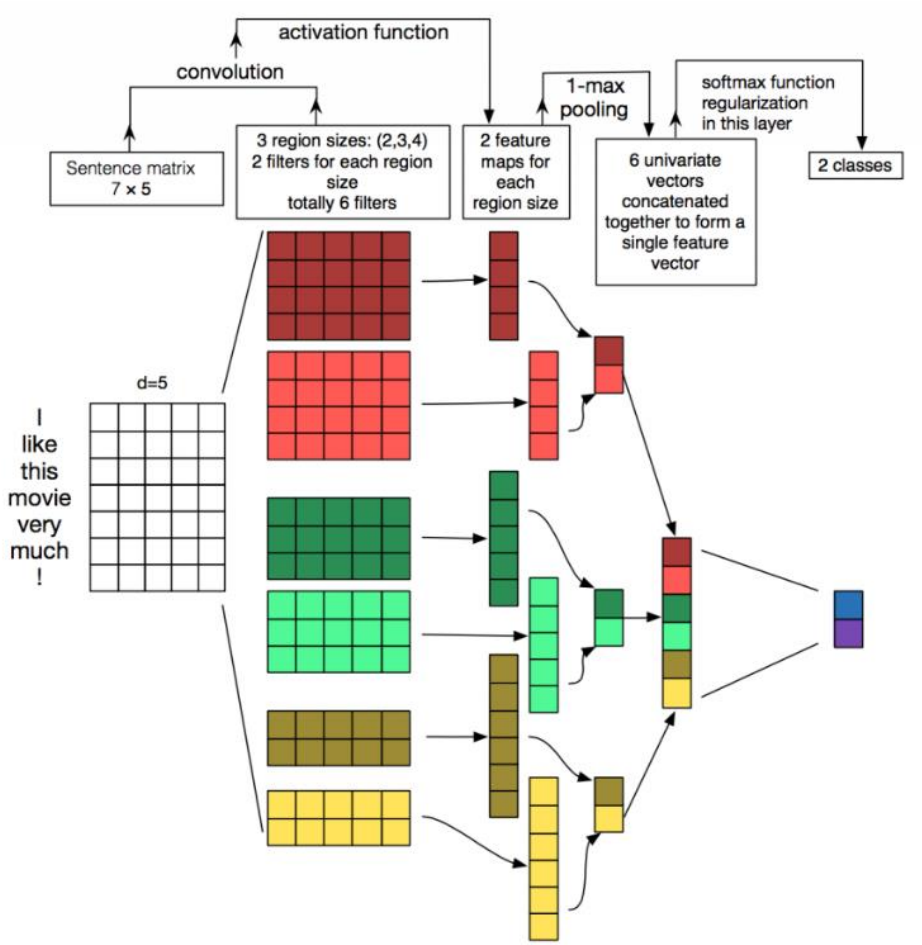
Heirarchical ConvNet(CNN)

Filter의 개념 (특정 단어들을 중요하게 생각한다)



Heirarchical ConvNet(CNN)

1. 특정 단어들의 중요도를 다르게하는 필터들을 slide하면서 벡터를 만든다
2. 그 벡터들을 크기가 똑같은 것들 끼리 묶어써 max-pooling해준다 한후 concatenate 해준다
3. 나온 모든 벡터들을 하나의 벡터로 합친다
4. 그 벡터를 softmax function regularization해준다



Evaluation of Sentence Represenations

자체적인 평가 툴 SentEval을 활용하여 기존 task들을 자동화하여 평가 할 수 있도록 했다. 이 툴은 Adam을 활용하여 batch size 가 64인 logistic regression classifier을 적용한다.

Task들은 다음과 같다.

- Binary and multi-class classification
- (binary 및 multi-class 분류)
- Entailment and semantic relatedness

(포함관계 및 의미론적 유사성)

- STS14 - Semantic Textual Similarity

(STS-14데이터셋을 활용한 의미론적 텍스트 유사성)

- Paraphrase detection
- Caption-Image retrieval

(상대적으로 중요하게 다루지 않은 개념)

*semantic relatedness

Similar함과는 다소 다른 개념. 예를 들어 spoon과 ice cream은 relate되어 있지만, similar 한 것은 아니다. 마찬가지로 I eat soup with a spoon 하고 I eat ice cream with a spoon에서 ice cream과 spoon은 relat하다. 왜냐면 두 단어 다 spoon으로 떠 먹는 단어들이다

다음 표는 classification task이 이루어지는 방법을 보여주는 표인데 C는 몇가지 class로 분류할 것인지 (예를 들면 영화의 sentiment이 긍정인지 부정인지), N은 샘플 수이다.

name	N	task	C	examples
MR	11k	sentiment (movies)	2	"Too slow for a younger crowd , too shallow for an older one." (neg)
CR	4k	product reviews	2	"We tried it out christmas night and it worked great ." (pos)
SUBJ	10k	subjectivity/objectivity	2	"A movie that doesn't aim too high , but doesn't need to." (subj)
MPQA	11k	opinion polarity	2	"don't want"; "would like to tell"; (neg, pos)
TREC	6k	question-type	6	"What are the twin cities ?" (LOC:city)
SST	70k	sentiment (movies)	2	"Audrey Tautou has a knack for picking roles that magnify her [..]" (pos)

Table 1: **Classification tasks**. C is the number of class and N is the number of samples.

Evaluation of Sentence Represenations

NLI(Natural Language Inference) 및
STS(Semantic Textual Similarity) Task

NLI은 SNLI dataset와 SICK-E dataset을 활용하여 두 문장의 관계를 포함관계, 모순관계, 아니면 중립관계로 볼 것인지 판별 하는 것이고

STS은 SICK-R와 STS14 dataset을 활용하여 두 문장간의 유사도(0~5사이의 척도에서)를 측정하는 것이다. 모두다 사람의 손으로 labelling 된 것이다.

name	task	N	premise	hypothesis	label
SNLI	NLI	560k	"Two women are embracing while holding to go packages."	"Two woman are holding packages."	entailment
SICK-E	NLI	10k	A man is typing on a machine used for stenography	The man isn't operating a steno-graph	contradiction
SICK-R	STS	10k	"A man is singing a song and play-ing the guitar"	"A man is opening a package that contains headphones"	1.6
STS14	STS	4.5k	"Liquid ammonia leak kills 15 in Shanghai"	"Liquid ammonia leak kills at least 15 in Shanghai"	4.6

Table 2: **Natural Language Inference and Semantic Textual Similarity tasks**. NLI labels are contra-diction, neutral and entailment. STS labels are scores between 0 and 5.

Model	dim	NLI dev	NLI test	Transfer micro	Transfer macro
LSTM	2048	81.9	80.7	79.5	78.6
GRU	4096	82.4	81.8	81.7	80.9
BiGRU-last	4096	81.3	80.9	82.9	81.7
BiLSTM-Mean	4096	79.0	78.2	83.1	81.7
Inner-attention	4096	82.3	82.5	82.1	81.0
HConvNet	4096	83.7	83.4	82.0	80.9
BiLSTM-Max	4096	**85.0**	**84.5**	**85.2**	**83.7**

Table 3: **Performance of sentence encoder architectures** on SNLI and (aggregated) transfer tasks. Dimensions of embeddings were selected according to best aggregated scores (see Figure 5).

Evaluation of Sentence Represenations

특정 모델들은 Embedding Size에 따른 micro-average performance는 embedding의 크기가 커질 수록 performance또한 현저하게 올라가는 경우가 있었다

(BiLSTM, H ConvNet, inner-att)

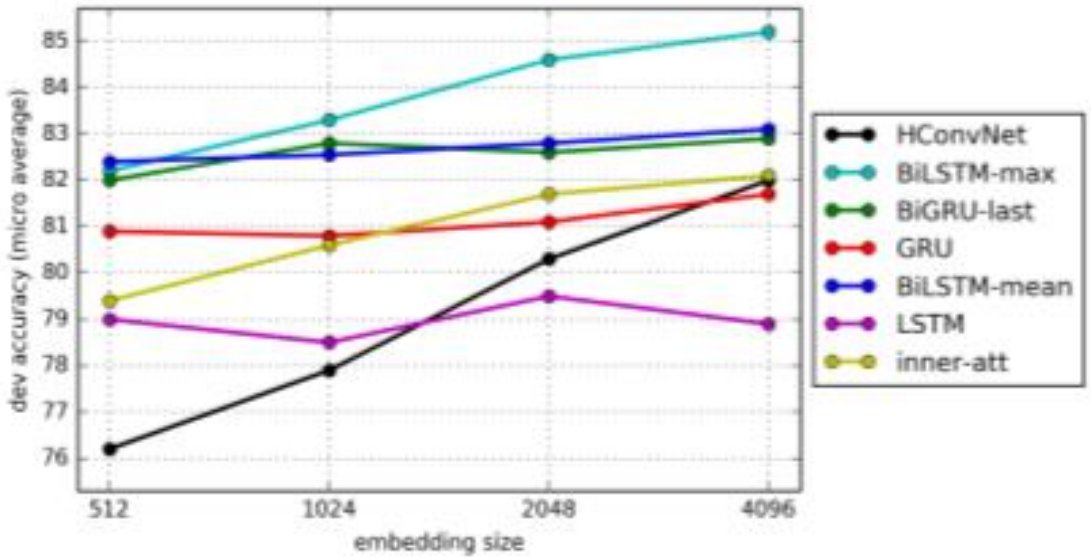


Figure 5: **Transfer performance w.r.t. embedding size** using the micro aggregation method.

Task Results

결과는 크게 세가지 그룹으로 분류하였다.

Group 1: unsupervised unordered sentences을 사용한 모델들

Group 2: unsupervised ordered sentences을 사용한 모델들

Group 3: supervised training을 사용한 모델들

*ordered sentences은?: 'Word Ordering Without Syntax' 논문 참고

현재까지 SkipThought LM 모델이 가장 성능이 좋았으나 학습시간이 길고(1달)

Training set의 규모도 6400만 문장을 활용했다. 그에 반면, SNLI(570k)의 상대적으로 적은 양의 데이터로 supervised training을 하여 일관되게 SkipThought의 성능을 뛰어넘었다.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STS14
Unsupervised representation training (unordered sentences)										
Unigram-TFIDF	73.7	**79.2**	90.3	82.4	-	85.0	73.6/81.7	-	-	.58/.57
ParagraphVec (DBOW)	60.2	66.9	76.3	70.7	-	59.4	72.9/81.1	-	-	.42/.43
SDAE	74.6	78.0	90.8	86.9	-	78.4	**73.7**/80.7	-	-	.37/.38
SIF (GloVe + WR)	-	-	-	-	82.2	-	-	-	**84.6**	**.69**/ -
word2vec BOW†	77.7	79.8	90.9	88.3	79.7	83.6	72.5/81.4	0.803	78.7	.65/.64
fastText BOW†	78.3	81.0	**92.4**	87.8	**81.9**	84.8	**73.9/82.0**	0.815	78.3	.63/.62
GloVe BOW†	**78.7**	78.5	91.6	87.6	79.8	83.6	72.1/80.9	0.800	78.6	.54/.56
GloVe Positional Encoding†	78.3	77.4	91.1	87.1	80.6	83.3	72.5/81.2	0.799	77.9	.51/.54
BiLSTM-Max (untrained)†	77.5	**81.3**	89.6	**88.7**	80.7	**85.8**	73.2/81.6	**0.860**	83.4	.39/.48
Unsupervised representation training (ordered sentences)										
FastSent	70.8	78.4	88.7	80.6	-	76.8	72.2/80.3	-	-	**.63/.64**
FastSent+AE	71.8	76.7	88.8	81.5	-	80.4	71.2/79.1	-	-	.62/.62
SkipThought	76.5	80.1	93.6	87.1	82.0	**92.2**	**73.0/82.0**	**0.858**	82.3	.29/.35
SkipThought-LN	**79.4**	**83.1**	**93.7**	**89.3**	82.9	88.4	-	**0.858**	79.5	.44/.45
Supervised representation training										
CaptionRep (bow)	61.9	69.3	77.4	70.8	-	72.2	73.6/81.9	-	-	.46/.42
DictRep (bow)	76.7	78.7	90.7	87.2	-	81.0	68.4/76.8	-	-	**.67**/**.70**
NMT En-to-Fr	64.7	70.1	84.9	81.5	-	82.8	69.1/77.1	-		.43/.42
Paragram-phrase	-	-	-	-	79.7	-	-	0.849	83.1	**.71**/ -
BiLSTM-Max (on SST)†	(*)	83.7	90.2	89.5	(*)	86.0	72.7/80.9	0.863	83.1	.55/.54
BiLSTM-Max (on SNLI)†	79.9	84.6	92.1	**89.8**	83.3	**88.7**	75.1/82.3	**0.885**	**86.3**	.68/.65
BiLSTM-Max (on AllNLI)†	**81.1**	**86.3**	**92.4**	**90.2**	**84.6**	88.2	**76.2**/**83.1**	**0.884**	**86.3**	**.70**/**.67**
Supervised methods (directly trained for each task – no transfer)										
Naive Bayes - SVM	79.4	81.8	93.2	86.3	83.1	-	-	-	-	-
AdaSent	83.1	86.3	95.5	93.3	-	92.4	-	-	-	-
TF-KLD	-	-	-	-	-	-	80.4/85.9	-	-	-
Illinois-LH	-	-	-	-	-	-	-	-	84.5	-
Dependency Tree-LSTM	-	-	-	-	-	-	-	0.868	-	-

Table 4: **Transfer test results for various architectures trained in different ways.** Underlined are best results for transfer learning approaches, in bold are best results among the models trained in the same way. † indicates methods that we trained, other transfer models have been extracted from (Hill et al., 2016). For best published supervised methods (no transfer), we consider AdaSent (Zhao et al., 2015), TF-KLD (Ji and Eisenstein, 2013), Tree-LSTM (Tai et al., 2015) and Illinois-LH system (Lai and Hockenmaier, 2014). (*) Our model trained on SST obtained 83.4 for MR and 86.0 for SST (MR and SST come from the same source), which we do not put in the tables for fair comparison with transfer methods.