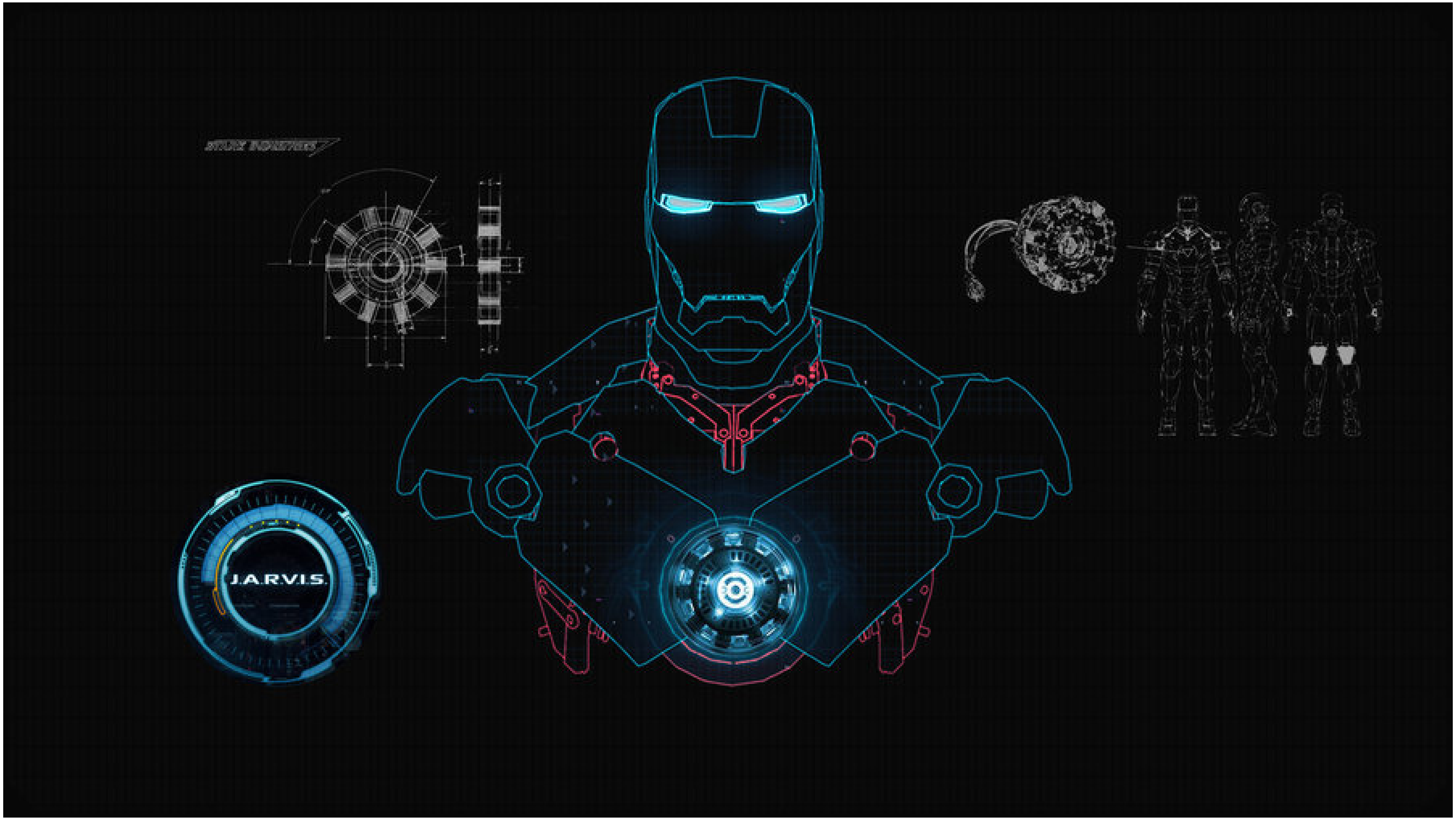
Question Answering System: Just A Rather Deep Intelligent System(JARDIS) - Week2



JARDIS

- Week 2

Content

1. 지난 주 수업 후 미팅에서 무엇을 하였는가
(What we did on the last week’s meeting after class)

2. 지난 주말 무엇을 하였는가
(What we did last weekend)

3. 오늘까지 무엇을 했는가
(What have we done for now)

4. 앞으로 무엇을 할것인가
(The plan)

JARDIS
- Week 2

1. 지난 주 수업 후 미팅에서 무엇을 하였는가
(What we did on the last week’s meeting after class)

- Let’s implement Word2Vec (til the next meeting)

- Word2Vec, GloVe 에 관한 글 읽고 Word Embedding 이 무엇인지 아는 사람이 됨

- Set up a video meeting schedule, Sun Jul 09

- List Randomizer 를 활용한 발표 순서 정하기

JARDIS
- Week 2

2. 지난 주말 무엇을 하였는가
(What we did last weekend)

- Knolpy, Word2Vec, doc2vec implementation

- zoom.us 화상미팅

- Sequence to sequence model 에 대해 알아보자

Introduction to GRU model using keras

Introduction to Sequence to sequence model

Introduction to dynamic memory networks

Introduction to end-to-end memory networks

Proceed with the implementation and experiments

NLP basics

: following the implementation based on the slides in link

Data preprocessing (feak. KoNLPy)

In [7]:

```
def read_data(filename):
    with open(filename, 'rt', encoding='UTF8') as f:
        data = [line.split('\t') for line in f.read().splitlines()]
        data = data[1:] # erase header
    return data
```

In [8]:

```
train_data = read_data('./nsmc/ratings_train.txt')
test_data = read_data('./nsmc/ratings_test.txt')

print(len(train_data))
print(len(train_data[0]))
print(len(test_data))
print(len(test_data[0]))
```

```
150000
3
50000
3
```

POS analyzer

In [8]:

```
from pprint import pprint
'''from konlpy.tag import Twitter

pos_tagger = Twitter()
def tokenize(doc):
    return ['/'.join(t) for t in pos_tagger.pos(doc,norm = True, stem = True)]

train_docs = [(tokenize(row[1]), row[2]) for row in train_data]
test_docs = [(tokenize(row[1]), row[2]) for row in test_data]

print(len(train_docs),len(train_docs[0]))
pprint(train_docs[0])'''
```

Out[8]:

```
"from konlpy.tag import Twitter\n\npos_tagger = Twitter()\ndef tokenize(doc):\n    return ['/'.j
oin(t) for t in pos_tagger.pos(doc,norm = True, stem = True)]\n\ntrain_docs = [(tokenize(row
[1]), row[2]) for row in train_data]\ntest_docs = [(tokenize(row[1]), row[2]) for row in test_da
ta]\n\nprint(len(train_docs),len(train_docs[0]))\npprint(train_docs[0])"
```

In [2]:

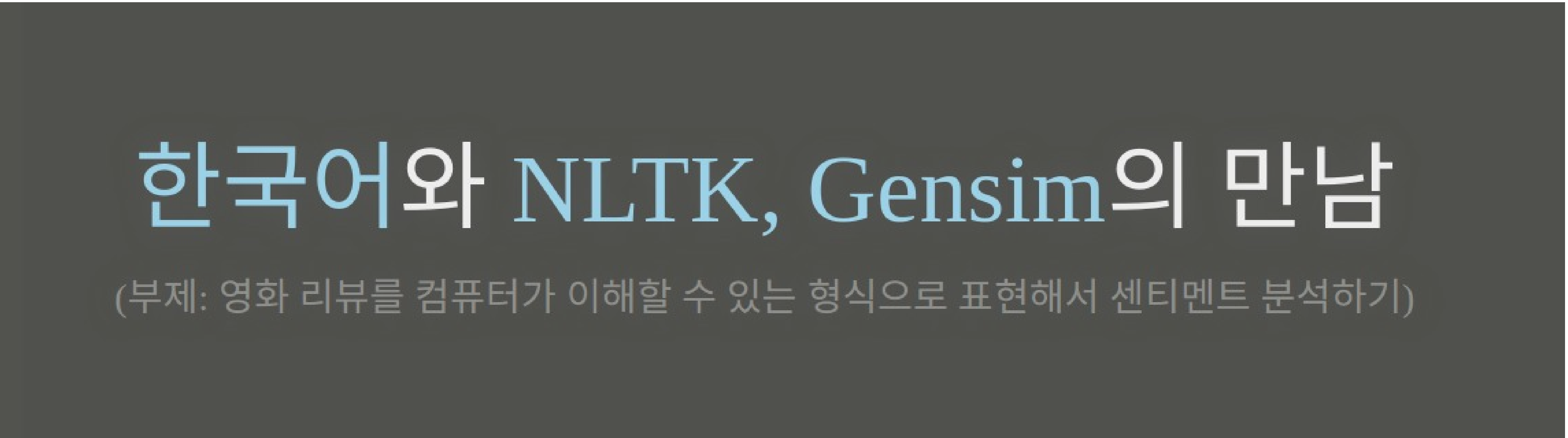
```
import pickle
'''
with open('train_docs.txt','wb') as fp :
```

JARDIS

- Week 2

2. 지난 주말 무엇을 하였는가
(What we did last weekend)

- Konlpy, Word2Vec, doc2vec implementation



한국어와 NLTK, Gensim의 만남

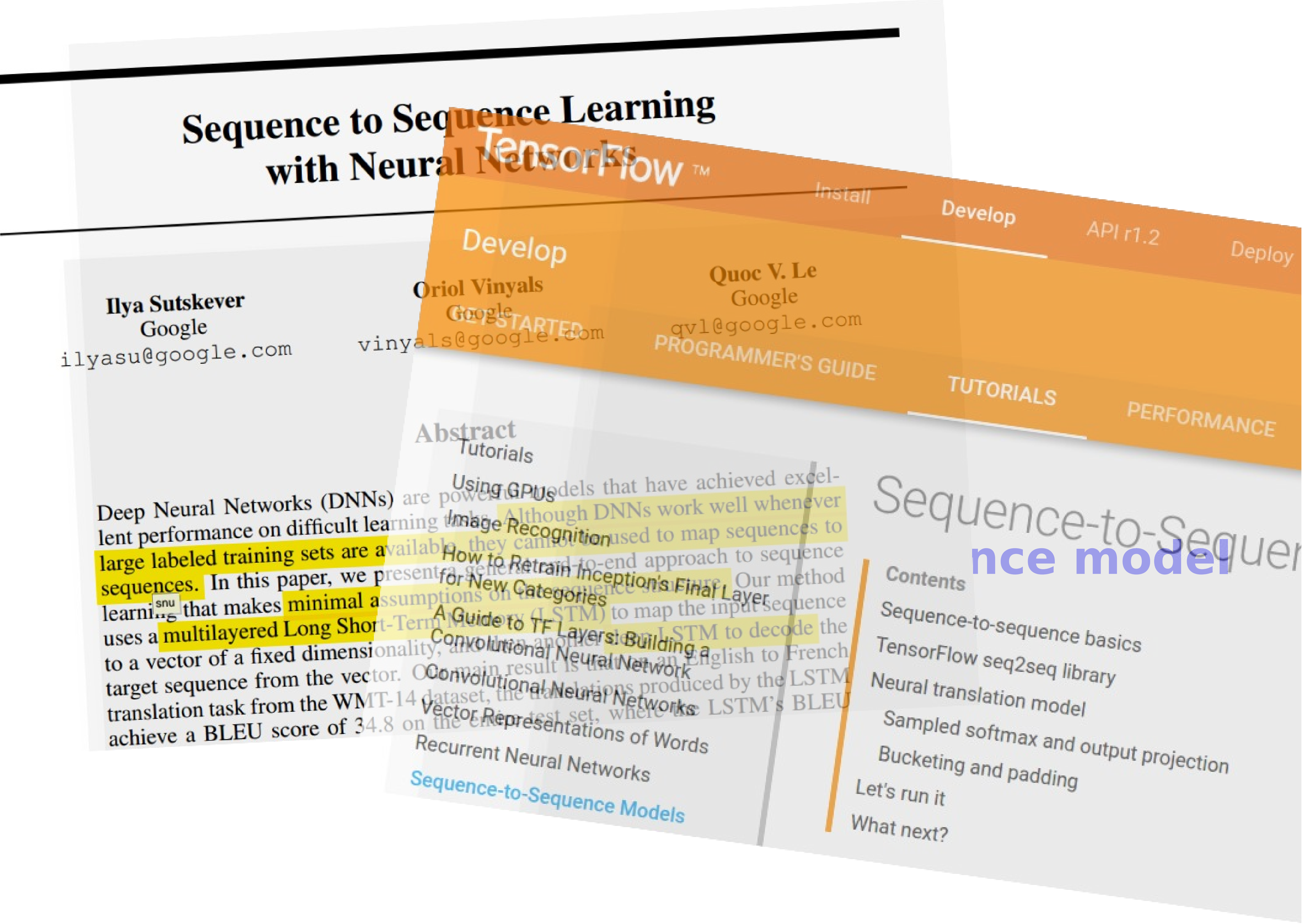
(부제: 영화 리뷰를 컴퓨터가 이해할 수 있는 형식으로 표현해서 센티멘트 분석하기)

JARDIS

- Week 2

3. 오늘까지 무엇을 했는가
(What have we done for now)

Introduction to Sequence to sequence model



Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT-14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU

TensorFlow ™

Install Develop API r1.2 Deploy

Develop

GET STARTED PROGRAMMER'S GUIDE TUTORIALS PERFORMANCE

Sequence-to-Sequen

Contents

nce model

JARDIS

- Week 2

3. 오늘까지 무엇을 했는가
(What have we done for now)

Introduction to Sequence to sequence model

RNN:
LSTM:
Sequence to sequence model:

JARDIS

- Week 2

3. 오늘까지 무엇을 했는가
(What have we done for now)

Introduction to Sequence to sequence model

RNN: 이전의 데이터가 새로운 데이터 처리에 영향

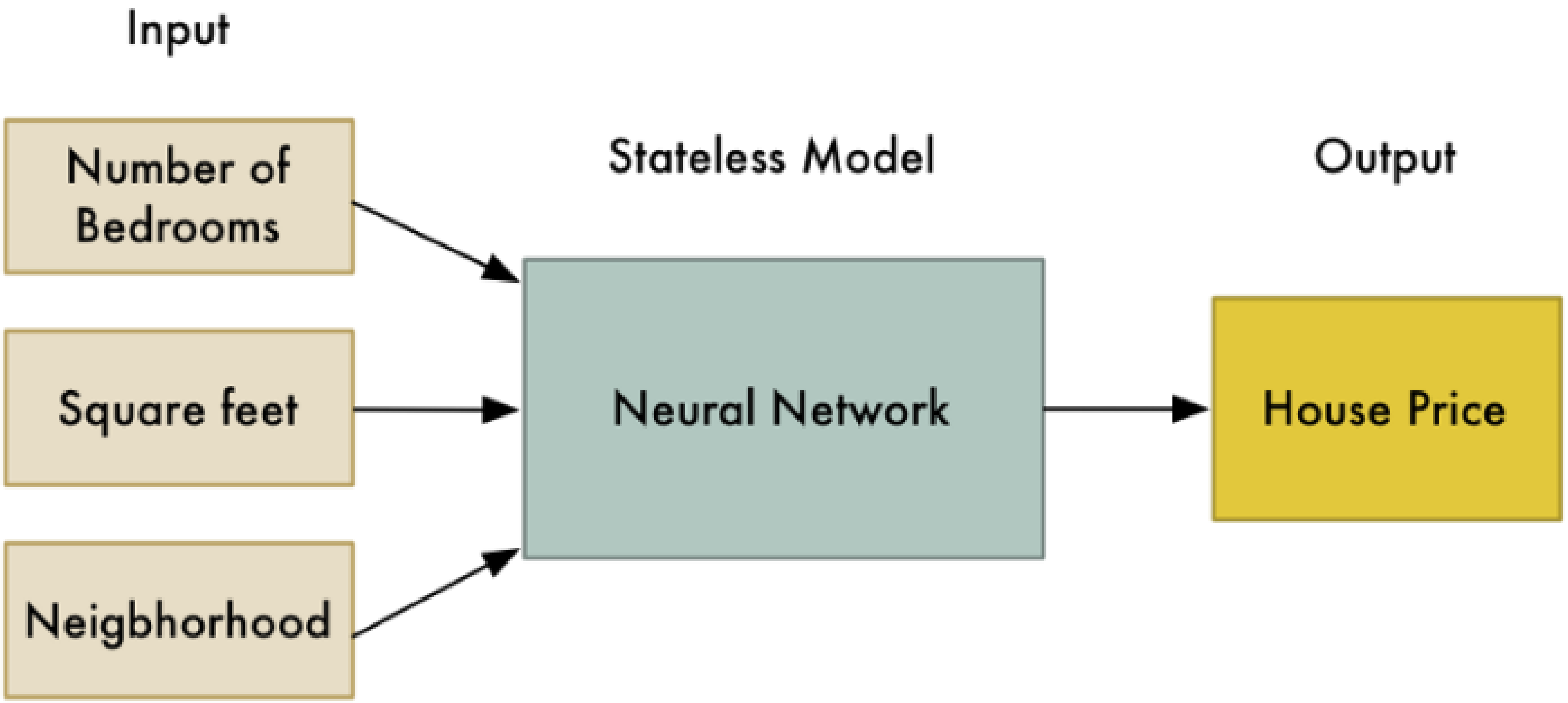
LSTM: RNN 보다 긴 문장을 학습하는데 성능이 좋다고 함

Sequence to sequence model: 좌르륵 연속된 데이터가 들어가고 , 또 좌르륵 연속된 데이터를 내보내는 모델

JARDIS

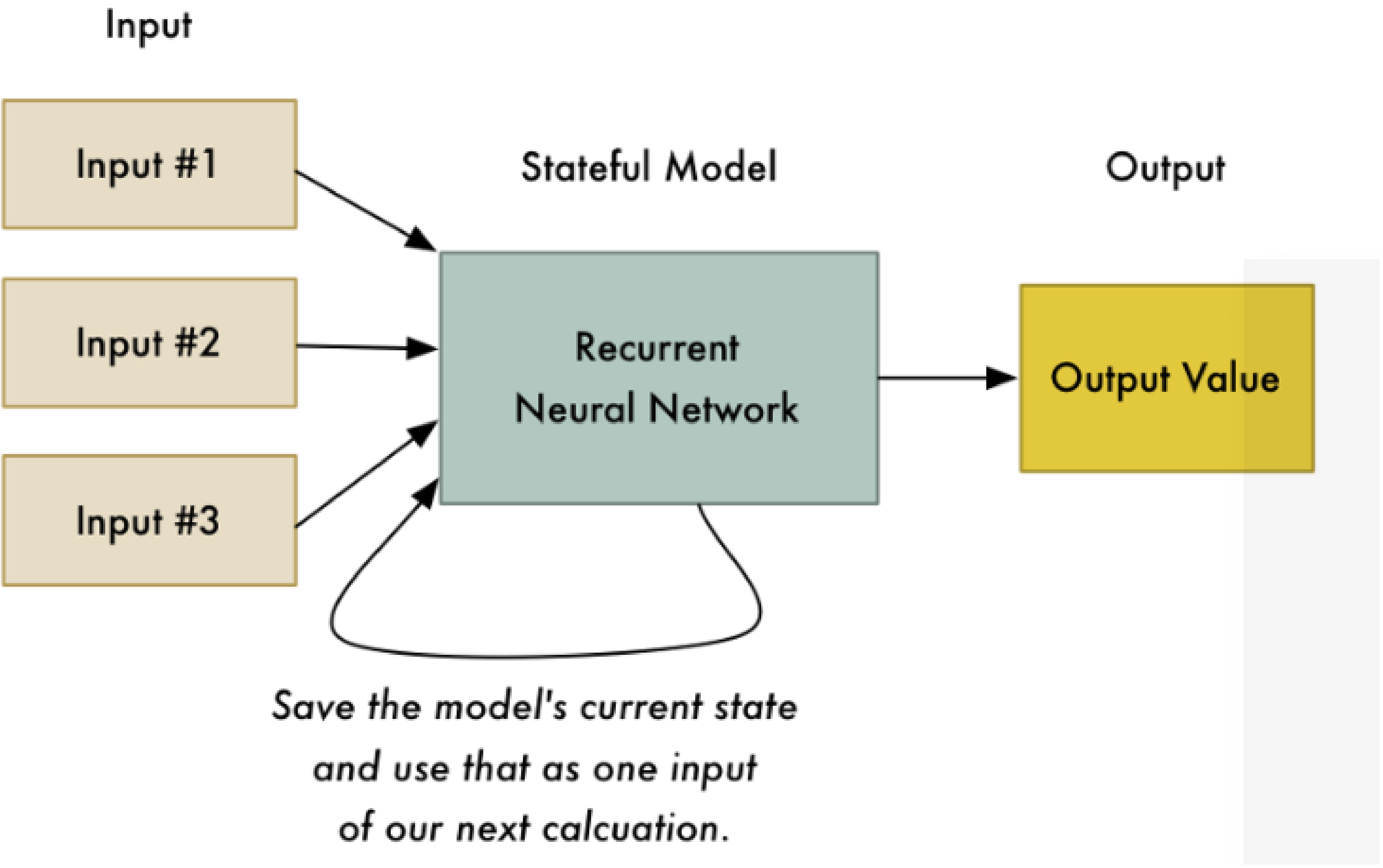
- Week 2

Neural Network



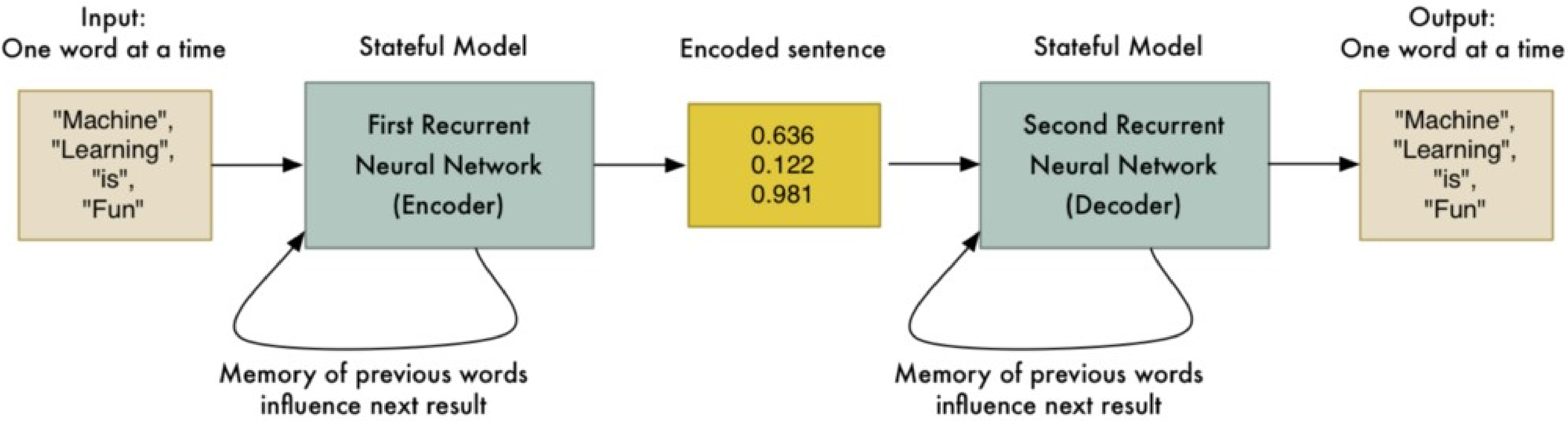
JARDIS

- Week 2



JARDIS

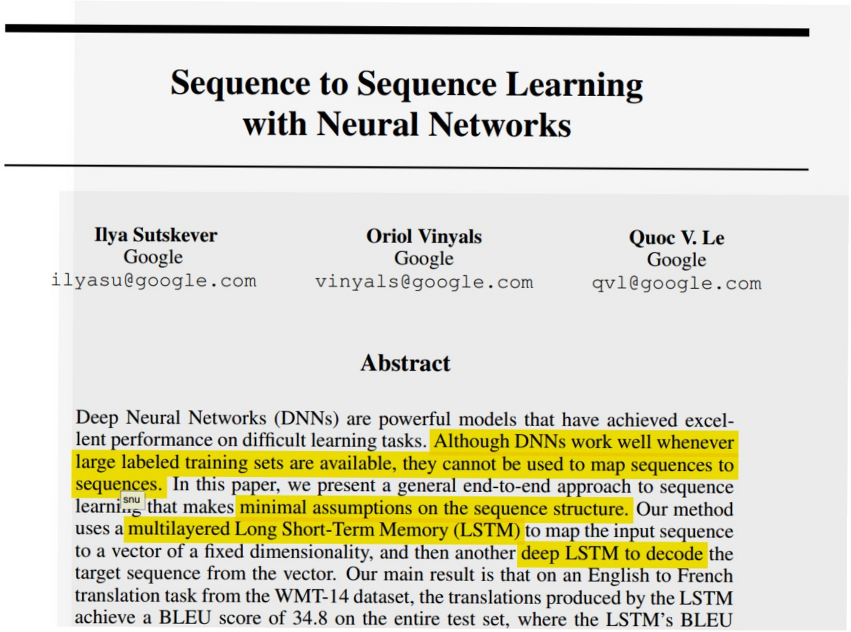
- Week 2



첫번째 RNN: 문장을 나타내는 인코딩을 생성
두번째 RNN: 인코딩 된 것을 받아서 로직을 역으로 돌려서 디코딩

JARDIS

- Week 2



Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT-14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU

Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.

JARDIS

- Week 2

4. 앞으로 무엇을 할것인가
(The plan)

- Seq2Seq 에 대해 공부한 것 처럼 , 다른 모델에 대해서 공부하기
- 한글데이터 Word2Vec 모형을 학습한 것에 다양한 변화 주기
- Word2Vec 모델을 가지고 한글 단어 유사도 테스트를 해 보기
(MSE, Pearson correlation)
- Word2Vec 모델 window size 조절해 가면서 테스트 해 보기

JARDIS

- Week 2

감사합니다 !