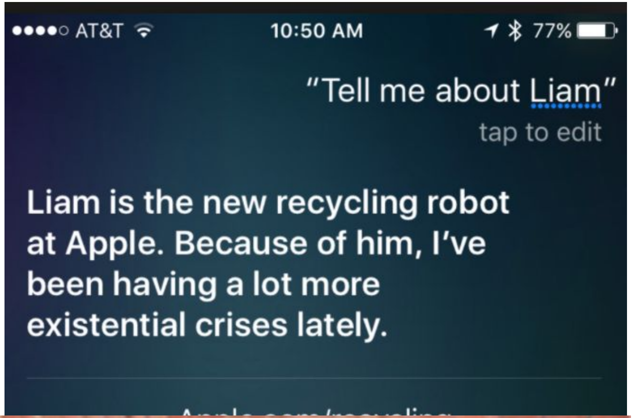
Introduction to NLP

What is natural language processing?

Difficult?
Where do we use natural language processing?

- Question answering
- Machine translation
- A lot More !



http://blog.webcertain.com/machine-translation-technology-the-search-engine-takeover/18/02/2015/

https://sixcolors.com/post/2016/04/siri-tells-you-all-about-liam/

Introduction to NLP

So NLP is something that can help machines achieve these tasks, right?

We can define NLP as:

- A work which enables machines to "understand" human language and further performs useful tasks
- It needs knowledge from CS, AI, Linguistics

Difficult!

Introduction to NLP

Difficulties in NLP:

- We omit a lot of common knowledge, which we assume the reader possesses
- We keep a lot of ambiguities, which we assume the reader knows how to resolve
  - e.g. “The man saw a boy with a telescope.” Who has a telescope? => Ambiguity is a killer

Introduction to NLP

Currently, what are the tools that are commonly used in NLP ?

An interesting demo here: Stanford CoreNLP Demo

- Part-Of-Speech tagging
- Entity Recognition
- Dependency Parsing
- etc

Due to the time limitation, we are gonna talk about some of these tools at the end.

Introduction to NLP

But **why** deep learning for NLP?

Most current NLP tasks work well because of human-designed features.

- Too specific and incomplete
- Require domain-specific knowledge

  => Different domain needs different features

Introduction to NLP

However, deep learning can alleviate these issues

- Features are learned automatically from examples
- The ability to capture the complicated relations

Furthermore

- Gigantic amount of data becomes available today
- Faster CPU/GPU enables us to do deep learning more efficiently

Introduction to NLP

Sounds good, right?

But how do we feed the text data into deep learning models
(e.g. the neural network) ?

This is the most basic and important step. How do we represent a word?

Word Representation

Common/intuitive way to represent a word in computer => using a vector!

A traditional approach: **discrete representation** (one-hot representation)

- Each word is represented using a vector of dimension |V| -- size of vocabulary
- "1" in one spot and "0" in all other spots

Example:
Corpus: "I like deep learning.", "I like neural networks.", "I can do NLP."
=> V = { "I", "like", "deep", "learning", "neural", "networks", "can", "do", "NLP" }

What is the one-hot representation for "like" ? (Using the above order)

=> (0, 1, 0, 0, 0, 0, 0, 0, 0)

Word Representation

Problems with one-hot representation

- Similar words cannot be represented in a similar way
  e.g. We have corpus with only 2 words {“skillful”, “adept”}
  vec(“skillful”) = (1,0), vec(“adept”) = (0,1)
  => The similarity is lost.

- The curse of dimensionality => computational complexity
- The vector is sparse

We need better representation !

Word Representation

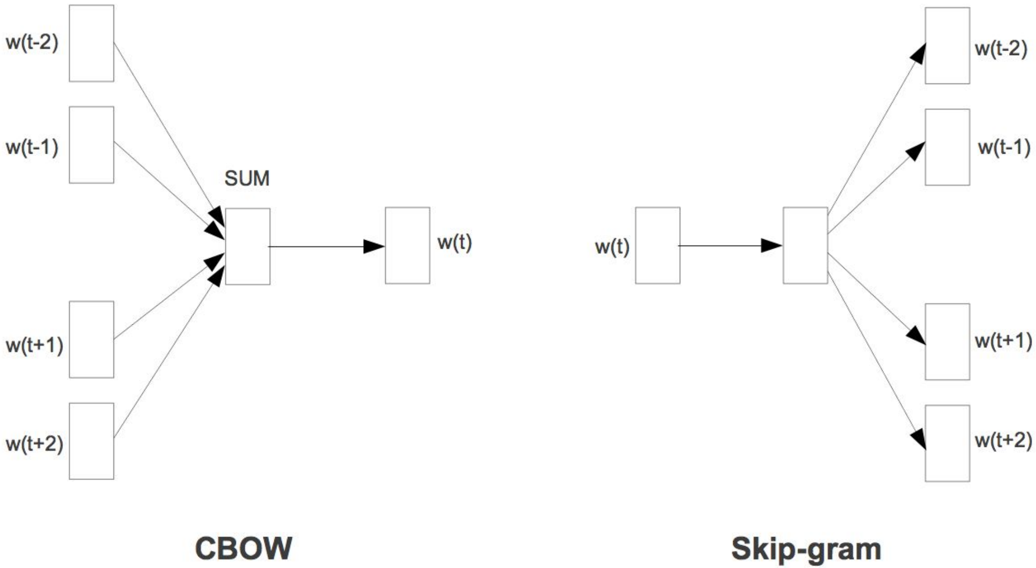
Idea:
We can represent a word by utilizing the information from its other words
=> **Distributional representation**

A Question:
Use **all other words** in the corpus OR just **a window** of words?
Lead to different approaches:
- Full-window approach: e.g. Latent Semantic Analysis (LSA)
- Local-window approach: e.g. Word2Vec

Word Representation

e.g. Word2Vec

- There are 2 variants -- Continuous bag-of-words (CBOW), skip-gram



Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. ICLR, 2013.

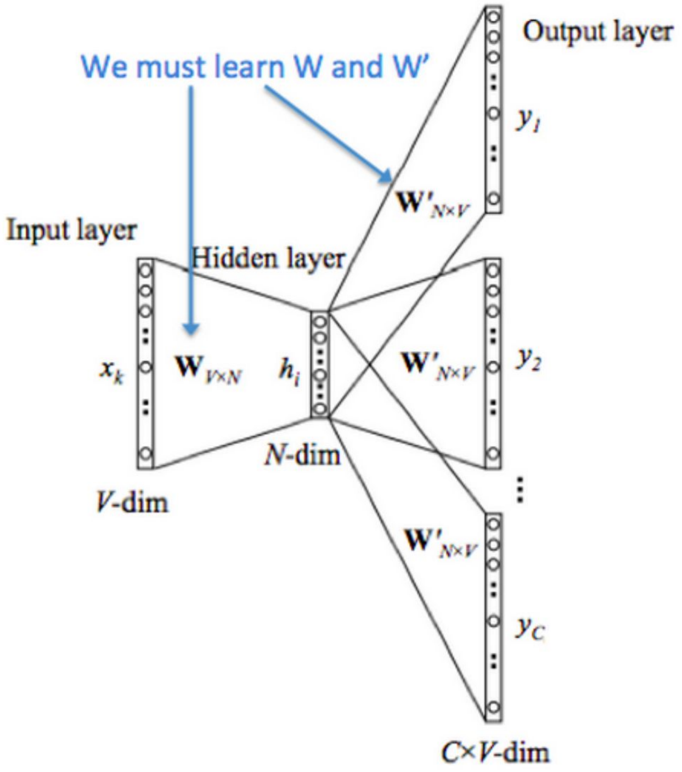
Word Representation

e.g. Word2Vec with skip-gram

- **<u>WI</u>**: input projection matrix of size |V|*N
- **<u>WO</u>**: output projection matrix of size N*|V|

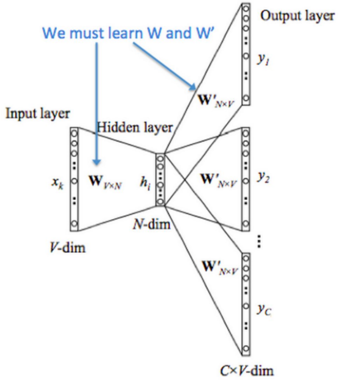
<u>- Objective function:</u>
= the averaged (difference between predicted probabilistic distribution and all neighbors in the window)

<u>An example to explain!</u>



https://web.archive.org/web/20160311161826/http://cs224d.stanford.edu/lecture_notes/LectureNotes1.pdf

Word Representation



e.g. Word2Vec with skip-gram

Example:

Corpus:

“the dog saw a cat”,“the dog chased the cat”, “The cat climbed tree”

Choose **N=3**, then:

|V| = 8, **WI** is of size 8*3, **WO** is of size 3*8

Target

The neighbors of “climbed” are: “cat”, “tree”

One-hot representation:

vec(“climbed”) = [0 0 0 1 0 0 0 0], vec(“cat”) = [0 1 0 0 0 0 0 0 0], vec(“tree”) = [0 0 0 0 0 0 0 1]

Goal...

https://web.archive.org/web/20160311161826/http://cs224d.stanford.edu/lecture_notes/LectureNotes1.pdf

Word Representation

e.g. Word2Vec

Good performance in analogy test both **syntactically** and **semantically**

$$x_{car} - x_{cars} \approx x_{family} - x_{families}$$

$$x_{shirt} - x_{clothing} \approx x_{chair} - x_{furniture}$$

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. ICLR, 2013.

Word Representation

But there are problems...

It only uses the information of a window of size N.

GloVe

Advantages:

- Leverage the global statistical information
- State-of-the-art performance on the analogy test as Word2Vec

More details at:

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. EMNLP, 2014.

Language Models

What are language models?

- Language models compute the probability of occurrence of a number of words in a particular sequence. E.g. $P(w_1, \ldots, w_m)$

Why do we care about language models?

- They are useful for lots of NLP applications like machine translation, text generation and speech recognition, etc.

Language Models

Machine Translation:

- P(strong tea) > P(powerful tea)

Speech Recognition:

- P(speech recognition) > P(speech wreck ignition)

Question Answering / Summarization:

- P(President X attended …) is higher for X = Trump

…

Language Models

Conventional language models apply a fixed window size of previous words to calculate probabilities. (count-based or NN models)

$$P(w_1, ..., w_m) = \prod_{i=1}^{i=m} P(w_i | w_1, ..., w_{i-1}) \approx \prod_{i=1}^{i=m} P(w_i | w_{i-(n-1)}, ..., w_{i-1})$$

Most state-of-the-art models are based on Recurrent Neural Networks (RNN), which are capable of conditioning the model on all previous words in the corpus.

http://cs224d.stanford.edu/lecture_notes/notes4.pdf

RNN in Neural Language Model (NLM)

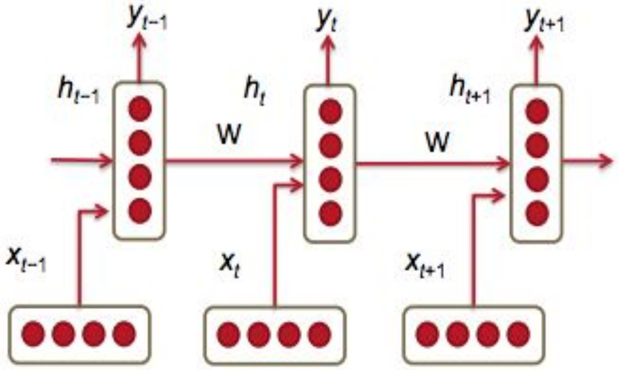
Hidden state: $$h_t = \sigma(W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]})$$

Output: $$\hat{y}_t = softmax(W^{(S)} h_t)$$

Loss function at t: $$J^{(t)}(\theta) = -\sum_{j=1}^{|V|} y_{t,j} \times log(\hat{y}_{t,j})$$

The cross entropy error over a corpus of size T: $$J = -\frac{1}{T} \sum_{t=1}^{T} J^{(t)}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{|V|} y_{t,j} \times log(\hat{y}_{t,j})$$

A measure of confusion: $$Perplexity = 2^J$$



Three-time-step RNN

http://cs224d.stanford.edu/lecture_notes/notes4.pdf

From RNN to CNN

Limitations of current RNN LM that can be alleviated by CNN:

- They are blind to sub-word information. (Morphologically rich languages)
  - Solution: Character-Aware NLM (Kim et al., 2015)
- The computation of features or states for different parts of long sequences cannot occur in parallel
  - Solution: Quasi-RNN (Bradbury et al., 2017)

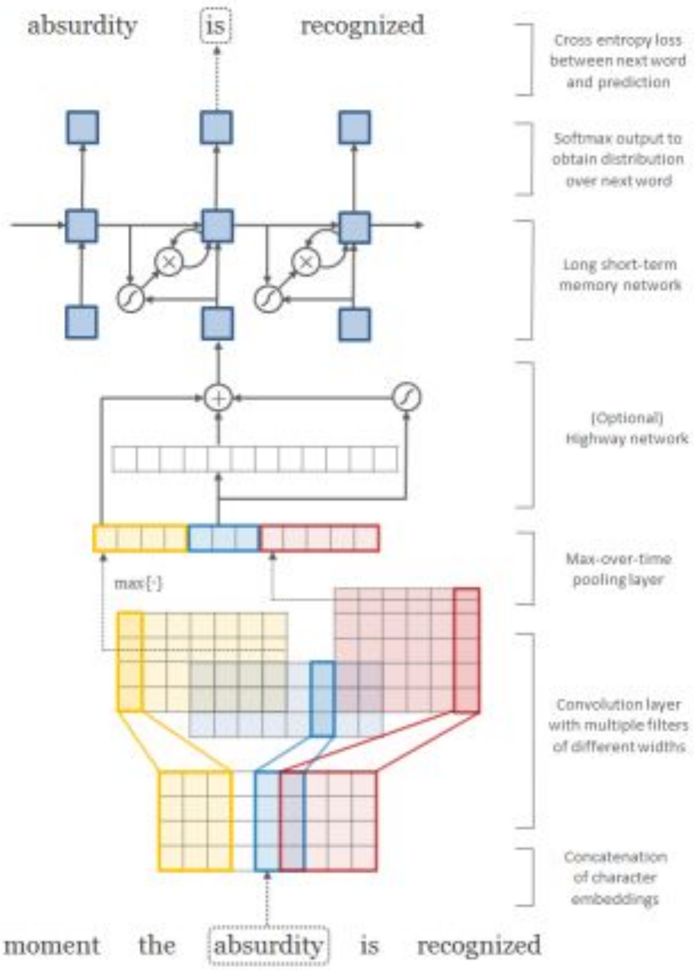
Character-Aware NLM

Highlights of the architecture:

- Instead of using word embeddings as input of RNN, (Kim et al., 2015) proposes to use the output of a character-level CNN as the input of RNN.
- The model has significantly fewer parameters as there is no word embedding involved.
- Highway network layer is added between CNN and RNN to boost performance.
    - Recap of highway network:

$$\mathbf{z} = \mathbf{t} \odot g(\mathbf{W}_H \mathbf{y} + \mathbf{b}_H) + (1 - \mathbf{t}) \odot \mathbf{y}$$

$$\mathbf{t} = \sigma(\mathbf{W}_T \mathbf{y} + \mathbf{b}_T)$$



https://arxiv.org/pdf/1508.06615.pdf

Experiments

	PPL	Size
LSTM-Word-Small	97.6	5 M
LSTM-CharCNN-Small	92.3	5 M
LSTM-Word-Large	85.4	20 M
LSTM-CharCNN-Large	78.9	19 M
Sum-Prod Net† (Cheng et al. 2014)	100.0	5 M
LSTM-Medium† (Zaremba et al. 2014)	82.7	20 M
LSTM-Large† (Zaremba et al. 2014)	78.4	52 M

Perplexity on Penn TreeBank (English)

		Cs	De	Es	Fr	Ru
B&B	KN-4	545	366	241	274	396
	MLBL	465	296	200	225	304
Small	Word	503	305	212	229	352
	Morph	414	278	197	216	290
	Char	397	250	174	203	284
Large	Word	493	286	200	222	357
	Morph	398	263	177	196	271
	Char	**375**	**238**	**163**	**184**	**269**

Perplexity on 2013 ACL Workshop on MT dataset

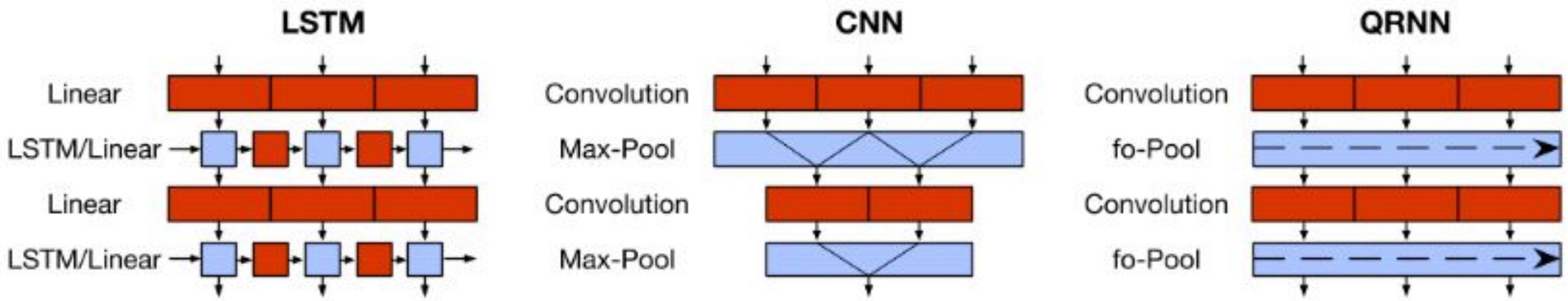
	Small	Large
No Highway Layers	100.3	84.6
One Highway Layer	92.3	79.7
Two Highway Layers	90.1	78.9
Multilayer Perceptron	111.2	92.6

Perplexity of models with different middle layers

https://arxiv.org/pdf/1508.06615.pdf

Quasi-RNN

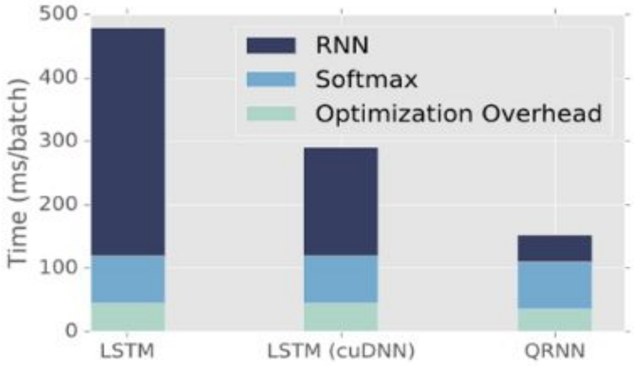
An approach to neural sequence modeling that alternates CNN, which apply in parallel across timesteps and a minimalist recurrent pooling function that applies in parallel across channels (Bradbury et al., 2017)



https://arxiv.org/pdf/1611.01576.pdf

Experiments

Model	Parameters	Validation	Test
LSTM (medium) (Zaremba et al., 2014)	20M	86.2	82.7
Variational LSTM (medium) (Gal & Ghahramani, 2016)	20M	81.9	79.7
LSTM with CharCNN embeddings (Kim et al., 2016)	19M	–	78.9
Zoneout + Variational LSTM (medium) (Merity et al., 2016)	20M	84.4	80.6
Our models			
LSTM (medium)	20M	85.7	82.0
QRNN (medium)	18M	82.9	79.9
QRNN + zoneout ($p = 0.1$) (medium)	18M	82.1	78.3



https://arxiv.org/pdf/1611.01576.pdf

Coreference Resolution

- What is Coreference Resolution ?
  - Identify all noun phrases(mentions) that refer

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.

- Applications
  - Full text understanding
  - Machine translation
  - Text summarization
  - information extraction and question answering

Coreference Resolution

- What is Coreference Resolution ?
  - Identify all noun phrases(mentions) that refer
  - Coreference resolution is a document-level structured prediction task

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.

- Applications
  - Full text understanding
  - Machine translation
  - Text summarization
  - information extraction and question answering

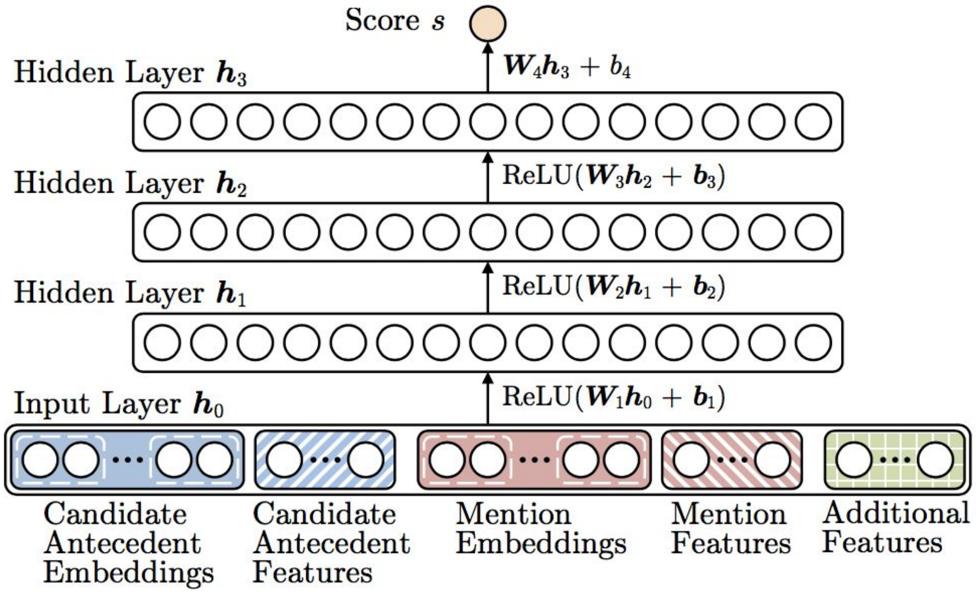
Coreference Models

- Mention Pair models
  - Treat coreference chains as a collection of pairwise links
  - Make independent pairwise decisions
  - Reconcile them in some deterministic way (e.g. transitivity)
- Mention-Ranking Models
  - Dominant approach to coreference resolution in recent years
  - Assign each mention its highest scoring candidate antecedent according to the model
  - Infer global structure by making a sequence of local decisions
- Entity-Mention models
  - A cleaner, but less studied approach
  - Explicitly cluster mentions of the same discourse entity

http://web.stanford.edu/class/cs224n/lectures/cs224n-2017-lecture15.pdf

Neural Mention-Pair Model

- Standard feed-forward neural network
  - From (Clark and Manning, 2016); similar to Wiseman et al. (2015)
  - Input layer: word embeddings and a few categorical features



http://cs.stanford.edu/people/kevclark//resources/clark-manning-emnlp2016-deep.pdf

Neural Mention-Pair Model

- Experiment
  - Dataset: English and Chinese Portions of the CoNLL 2012 Shared Task dataset

Model	English	Chinese
Chen & Ng (2012) [CoNLL 2012 Chinese winner]	54.52	**57.63**
Fernandes (2012) [CoNLL 2012 English winner]	**60.65**	51.46
Björkelund & Kuhn. (2014) Best previous Chinese system]	61.63	**60.06**
Wiseman et al. (2016) [Best previous English system]	**64.21**	—
Clark & Manning (ACL 2016)	65.29	63.66

Example Wins

Anaphor	Antecedent
the country's leftist rebels	the guerillas
the company	the New York firm
216 sailors from the ``USS cole''	the crew
the gun	the rifle

http://cs.stanford.edu/people/kevclark//resources/clark-manning-emnlp2016-deep.pdf