Language Models

Sandeep Soni

02/22/2024

Classification

x	y
The movie was fantastic	😀
The movie could have been better but wasn't bad	😐
That movie was a waste of time and money	😡

- Predict y from x
- $\hat{y} = \hat{h}(x)$
- Model $\hat{h}$ as P(Y|X)
- X is high-dimensional
- P(😀|"the", "movie", "was", "fantastic")

Classifiers

- Naive Bayes
  - Maximize P(X,Y) and then use Bayes theorem
  - Conditional independence assumptions
    $P(\mathrm{X}_i|\mathrm{Y}) = P(\mathrm{X}_i|\mathrm{Y}, \mathrm{X}_{i-1})$
  - $P(\mathrm{X}_i|\mathrm{Y})$ is estimated simply by counting
  - Smoothing is required for estimating probabilities

Generative

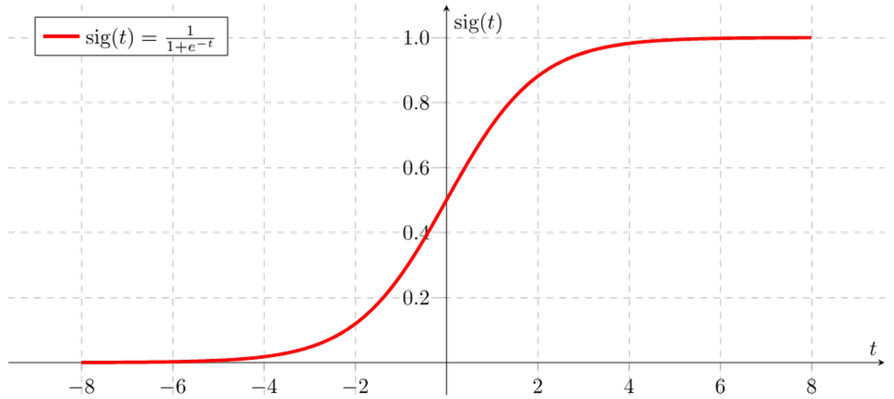
Classifiers

- Logistic Regression
  - Maximize P(Y|X)
  - $P(Y|X) = \sigma(\mathbf{w}^\top \mathbf{x})$
  - Weights learned using iterative algorithms
  - Regularization controlled with hyperparameters

Discriminative

Classifiers

- Logistic Regression
  - Maximize P(Y|X)
  - $P(Y|X) = \sigma(\mathbf{w}^{\top}\mathbf{x})$
  - Weights learned using iterative algorithms
  - Regularization controlled with hyperparameters



Discriminative

Classifiers

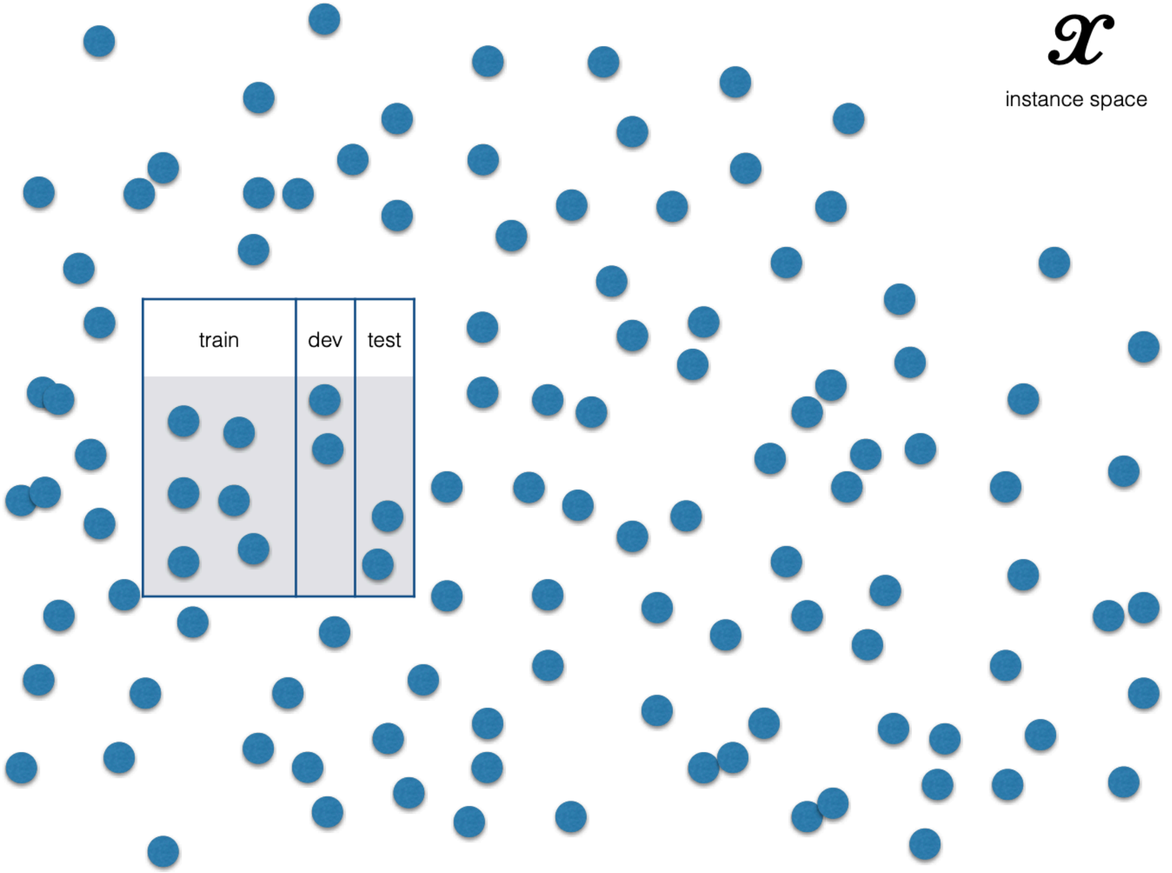
- Linear models: Naive Bayes, Logistic Regression
  - Perceptrons, support vector machines
- Non-linear models: Decision trees, kernel based methods, neural networks

Supervised learning

Train: Estimate weights, set hyperparameters

Dev: Evaluate using some performance metrics

Test: Do final evaluation



Slide from David Bamman

Evaluation

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

Evaluation

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

	y=spam	y=ham
yhat = spam	4	3
yhat=ham	1	2

Evaluation

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

	y=spam	y=ham
yhat = spam	4	3
yhat=ham	1	2

Evaluation

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

	y=spam	y=ham
yhat = spam	4	3
yhat=ham	1	2

$$N = \text{True positives} + \text{True negatives} + \text{False positives} + \text{False negatives}$$

Evaluation Metrics

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation Metrics

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation Metrics

$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{N}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation Metrics

$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{N}$$

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation Metrics

$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{N}$$

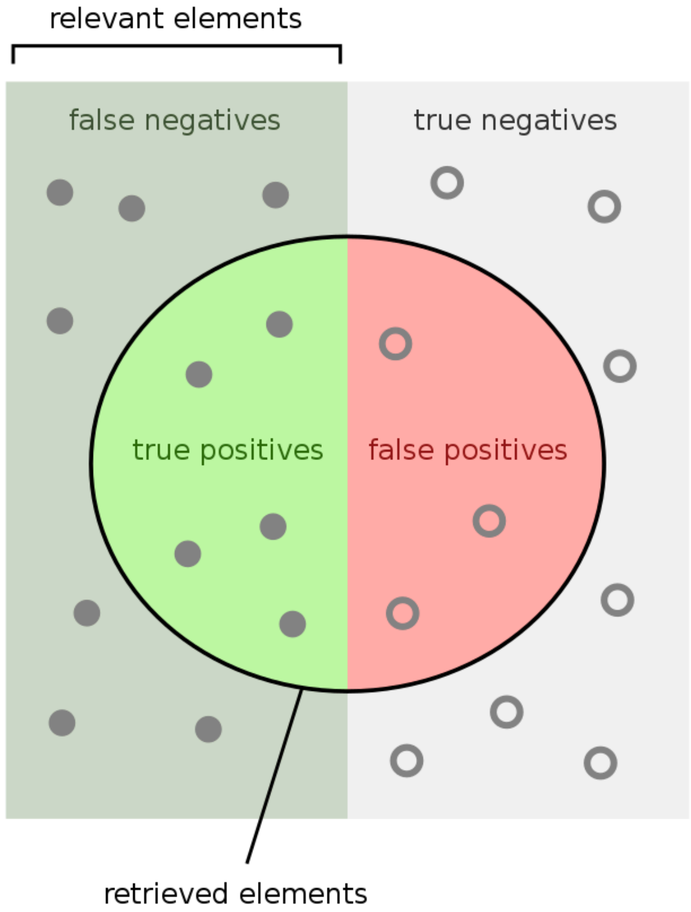
$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$\text{F}_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Precision and Recall



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{retrieved elements}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Source: Wikipedia

In class

- Text classification demo

Classification

$$P(y \mid x) = \frac{P(y)P(x \mid y)}{P(x)}$$

Bayes Theorem

$$P(y \mid x) = \frac{P(y)P(x \mid y)}{\sum_{y'} P(x, y')}$$

$$P(y \mid x) = \frac{P(y)P(x \mid y)}{\sum_{y'} P(y')P(x \mid y')}$$

Marginalization

What is the probability of a given piece of text i.e. P(x)?

Why should we model P(x)?

- NLP perspective: many tasks are sequence-to-sequence tasks
  - Machine translation: text in source language -> text in target language
  - Summarization: long text -> short text

Why should we model P(x)?

- A system should be “fluent” in generating output sequences
- How can we quantify fluency?

El cafe negro me gusta mucho

Machine translation system I

The coffee black me pleases much

El cafe negro me gusta mucho

Machine translation system II

I love dark coffee

Machine translation system I

The coffee black me pleases much

P(“The coffee black me pleases much”)

<

Machine translation system II

I love dark coffee

P(“I love dark coffee”)

“When I look at an article in Russian, I say: `This is really written in English but it has been coded in some strange symbols. I will now proceed to decode.”

—Warren Weaver (1955)

Noisy channel model

$$P_{\text{engish}\leftarrow\text{spanish}}(\mathbf{w}^e \mid \mathbf{w}^s) \propto P_{\text{engish,spanish}}(\mathbf{w}^e, \mathbf{w}^s)$$

$$= \boxed{P_{\text{spanish}\leftarrow\text{english}}(\mathbf{w}^{(s)} \mid \mathbf{w}^{(e)})} \times \boxed{P_{\text{english}}(\mathbf{w}^{(e)})}$$

Translation model

Language model

We want the noisy channel model to have high fidelity to the source text and high fluency in the generated output

Language model

- $\mathscr{V}$ is a vocabulary of symbols that need to be generated
- $\mathcal{S}$ is a infinite set of sequences of symbols taken from $\mathscr{V}$; each sequence ends with a special token STOP
- A sequence $x \in \mathcal{S}$

Language model

- $P(x) = P(x_1, x_2, \ldots, x_n)$
- P("I will be back") = P("I", "will", "be", "back", STOP)
- $0 \leq P(x) \leq 1$
- $\sum_{x' \in \mathcal{S}} P(x') = 1$

Language model

- Language modeling is a task of estimating a probabilistic model over words i.e. P(x)
- This is not easy!
- What is the P(“to be or not to be”)

Chain Rule

$$P(x_1, x_2, \ldots, x_n) = P(x_1)$$
$$\times P(x_2 \mid x_1)$$
$$\times P(x_3 \mid x_1, x_2)$$
$$\ldots$$
$$\times P(x_n \mid x_1, \ldots, x_{n-1})$$

$$P(\texttt{to be or not to be}) = P(\texttt{to})$$
$$\times P(\texttt{be} \mid \texttt{to})$$
$$\times P(\texttt{or} \mid \texttt{to}, \texttt{be})$$
$$\ldots$$
$$\times P(\texttt{be} \mid \texttt{to}, \texttt{be}, \ldots, \texttt{to})$$

Chain Rule

$$P(\text{to be or not to be}) = P(\text{to})$$

$$\times P(\text{be}|\text{to})$$

$$\times P(\text{or}|\text{to},\text{be})$$

$$\ldots$$

$$\times P(\text{be}|\text{to},\text{be},\ldots,\text{to})$$

P(“be” | “to be or not to”) is hard to estimate unless we make some assumptions

Markov assumptions

$$P(x_i \mid x_{i-1}, \ldots, x_1) \approx P(x_i)$$ zero-order

$$P(x_i \mid x_{i-1}, \ldots, x_1) \approx P(x_i \mid x_{i-1})$$ first-order

$$P(x_i \mid x_{i-1}, \ldots, x_1) \approx P(x_i \mid x_{i-2}, x_{i-1})$$ second-order

N-gram language models

$$P(x) = \prod_i P(x_i) \times P(\text{STOP})$$

unigram

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

$$P(x) = \prod_i P(x_i \mid x_{i-2}, x_{i-1}) \times P(\text{STOP} \mid x_{n-1}, x_n)$$

trigram

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

START _____

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

START I

START I _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

START I

START I like

START I like _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

START I

START I like

START I like black

START I like black _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

START I

START I like

START I like black

START I like black coffee

START I like black coffee ____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

START I

START I like

START I like black

START I like black coffee

START I like black coffee STOP

ESTIMATION

Maximum likelihood estimation

$$P(x) = \prod_i P(x_i) \times P(\text{STOP})$$

unigram

$$P(x_i) = \frac{c(x_i)}{N}$$

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

$$P(x_i \mid x_{i-1}) = \frac{c(x_i, x_{i-1})}{c(x_{i-1})}$$

$$P(x) = \prod_i P(x_i \mid x_{i-2}, x_{i-1}) \times P(\text{STOP} \mid x_{n-1}, x_n)$$

trigram

$$P(x_i \mid x_{i-2}, x_{i-1}) = \frac{c(x_i, x_{i-1}, x_{i-2})}{c(x_{i-1}, x_{i-2})}$$

Data sparsity

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Figure 4.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray.

SLP 4.3

If probabilities are estimated from a fixed finite sample, we'll encounter situations where we won't see some terms, bigrams, etc?

Data sparsity

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Figure 4.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray.

SLP 4.3

How to calculate P(x)? How to estimate P(x_i)?

Smoothing

Maximum likelihood estimate

$$P(x_i) = \frac{c(x_i)}{N}$$

Smoothed estimate

$$P(x_i) = \frac{c(x_i) + \alpha}{N + V\alpha}$$

Various different ways to do smoothing; other solutions include interpolation

Evaluation

- How do we know if our language model is good?
- Ideally, we should evaluate a language model extrinsically (e.g., how much a language model improves the performance of a machine translation model)
- More often we evaluate a language model intrinsically

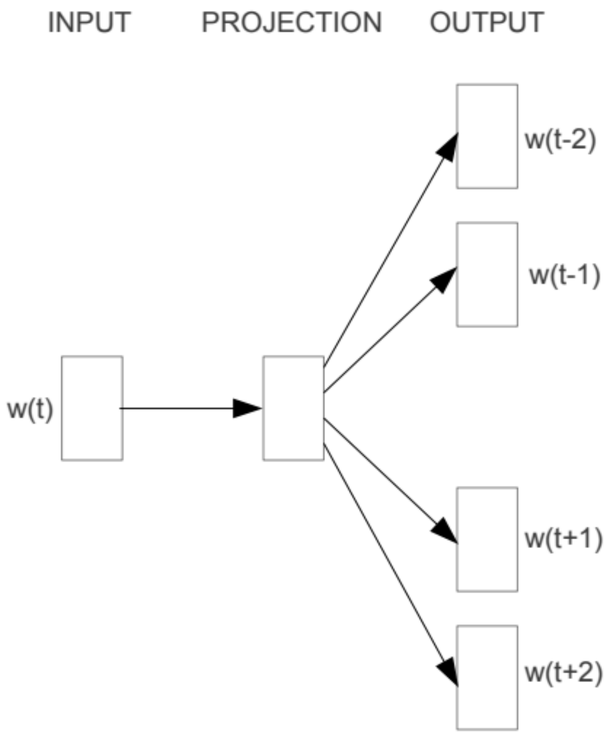
Evaluation

- A good language model should assign high probability to unseen sequences that are plausible
- A good language model should be less perplexed or surprised
- Perplexity = inverse probability of test data, averaged over words

Perplexity

- Perplexity = $2^{-\frac{l(w)}{M}}$, where M is the total number of unseen tokens
- $l(w) = \sum_{m=1}^{M} \log P(w_m \mid w_{m-1}, \ldots, w_1)$
- Based on the language model of your choice, you'll calculate l(w)
- Smaller perplexity is better!

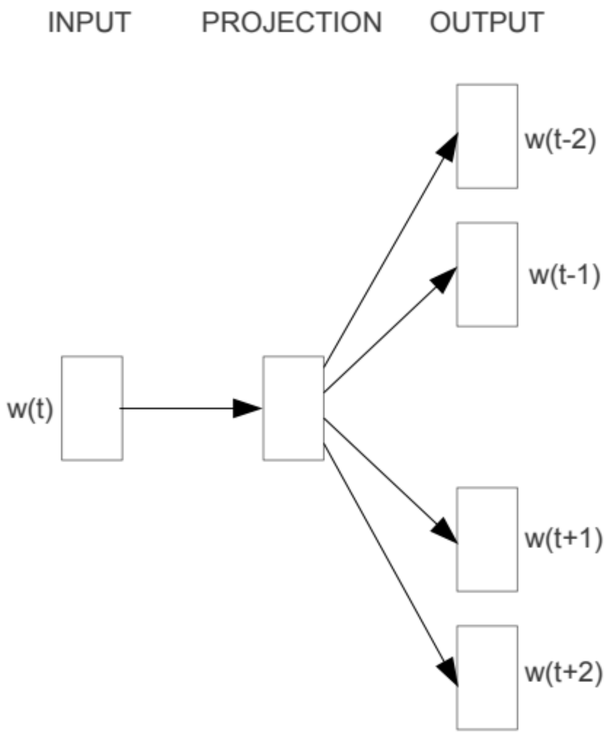
Skipgram



Skip-gram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong performance in the second half

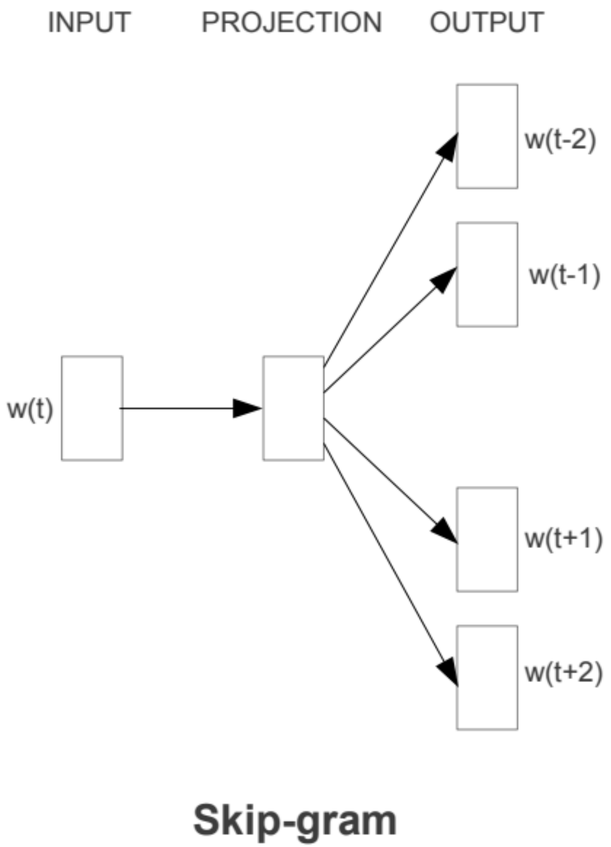
Skipgram



Skip-gram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong _____ in the second half

Skipgram



Skip-gram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong _____ in the second half

Word2vec learned word vectors using the skipgram language model

In class

- lm exploration