

Machine Learning (CS 181):
13. Topic Lecture
Neural Networks for Language

David Parkes and Sasha Rush

Contents

- 1 Language Models
- 2 Supervised Learning
- 3 Neural Networks For Language
- 4 Recurrent Neural Networks
- 5 Sequence-to-Sequence
- 6 Work at Harvard



*It is a capital mistake to theorize before one has data.
Insensibly one begins to twist facts to suit theories, instead of
theories to suit facts. -Sherlock Holmes, A Scandal in Bohemia*



*It is a capital mistake to theorize before one has data.
Insensibly one begins to twist facts to suit theories, instead of
theories to suit facts. -Sherlock Holmes, A Scandal in Bohemia*

It is a capital mistake to theorize before one has ----- ...

108 938 285 28 184 29 593 219 58 772 ----- ...

Language Modeling Task

Given a sequence of text give a probability distribution over the next word.

The Shannon game. Estimate the probability of the next letter/word given the previous.

*THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
READING LAMP ON THE DESK SHED GLOW ON
POLISHED ---*

■ Shannon (1948)

We may consider a discrete source, therefore, to be represented by a stochastic process. Conversely, any stochastic process which produces a discrete sequence of symbols chosen from a finite set may be considered a discrete source. This will include such cases as:

- 1. Natural written languages such as English, German, Chinese. ...*

4. *Third-order approximation*

*IN NO 1ST LAT WHEY CRATICT FROURE BIRS GROCID
PONDENOME OF DEMONSTURES OF THE REPTAGIN IS
REGOACTIONA OF CRE*

5. *First-Order Word Approximation. Rather than continue with tetragram, ... , ll-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.*

*REPRESENTING AND SPEEDILY IS AN GOOD APT OR
COME CAN DIFFERENT NATURAL HERE HE THE A IN
CAME THE TO OF TO EXPERT GRAY COME TO
FURNISHES THE LINE MESSAGE HAD BE THESE.*

6. Second-Order Word Approximation. The word transition probabilities are correct but no further structure is included.

*THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH
'RITER THAT THE CHARACTER OF THIS POINT IS
THEREFORE ANOTHER METHOD FOR THE LETTERS
THAT THE TIME OF WHO EVER TOLD THE PROBLEM
FOR AN UNEXPECTED*

The resemblance to ordinary English text increases quite noticeably at each of the above steps.

Language Modeling Applications

- Speech Recognition
- Machine Translation
- Summarization
- Dialogue
- Soft Keyboards
- Word Correction
- Text Simplification
- ...



Contents

- 1 Language Models
- 2 Supervised Learning
- 3 Neural Networks For Language
- 4 Recurrent Neural Networks
- 5 Sequence-to-Sequence
- 6 Work at Harvard

*THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
READING LAMP ON THE DESK SHED GLOW ON
POLISHED ___*

- Sample pairs are (\mathbf{x}, \mathbf{y}) .
- Input is sentence up until the blank, output is next word prediction.
- Challenging multi-class prediction problem, feature representation matters.
- Begin by considering count-based categorical prediction models:

$$p(\mathbf{y}|\mathbf{x})$$

*THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
READING LAMP ON THE DESK SHED GLOW ON
POLISHED ---*

- Sample pairs are (\mathbf{x}, \mathbf{y}) .
- Input is sentence up until the blank, output is next word prediction.
- Challenging multi-class prediction problem, feature representation matters.
- Begin by considering count-based categorical prediction models:

$$p(\mathbf{y}|\mathbf{x})$$

Word Representation

- We say the vocabulary of the language is $|\mathcal{V}|$
- In practice $|\mathcal{V}|$ is 10,000 - 100,000 unique words
- Each word has an associated numerical index.
- We represent each word as a one-hot vector C_k where $k \in \{1, \dots, |\mathcal{V}|\}$

$$[0, 0, 0, 0, 0, 1, \dots, 0]$$

- For this problem, both input and output will use one-hot C_k vectors.

Input Representation

- Recall *bag-of-words* model, with \mathbf{x} counts of words
- Why doesn't this work here?

- Alternative model, *bigram*, i.e. two words

~~THE ROOM WAS NOT VERY LIGHT A SMALL~~
~~OBLONG READING LAMP ON THE DESK SHED GLOW~~
~~ON POLISHED ---~~

- \mathbf{x} is one-hot vector $\mathbf{x} \in \{0, 1\}^{|\mathcal{V}|}$ representing last word.
- What assumptions is this making?

Input Representation

- Recall *bag-of-words* model, with \mathbf{x} counts of words
- Why doesn't this work here?
- Alternative model, *bigram*, i.e. two words

~~THE ROOM WAS NOT VERY LIGHT A SMALL~~
~~OBLONG READING LAMP ON THE DESK SHED GLOW~~
~~ON POLISHED ---~~

- \mathbf{x} is one-hot vector $\mathbf{x} \in \{0, 1\}^{|\mathcal{V}|}$ representing last word.
- What assumptions is this making?

Output Representation

- Output is the next word which we predict based on the last word.
- Let $\mathcal{Y} = \{0, 1\}^{|\mathcal{V}|}$ be the set of all possible words as one-hot vectors.
- Notation $\mathbf{y} = C_k$ means correct output is word k .

- What is the probability $p(\mathbf{y}|\mathbf{x})$ under a bigram distribution?
- Assume we have parameters for all possible inputs $\{\pi_j\}$ and that C_j is the last word seen.
- Model as a categorical distribution:

$$p(\mathbf{y} = C_k \mid \mathbf{x} = C_j; \{\pi_j\}) = \pi_{jk}$$

where for all j , $\pi_{jk} \geq 0$ and $\sum_{k=1}^c \pi_{jk} = 1$

- What is the probability $p(\mathbf{y}|\mathbf{x})$ under a bigram distribution?
- Assume we have parameters for all possible inputs $\{\pi_j\}$ and that C_j is the last word seen.
- Model as a categorical distribution:

$$p(\mathbf{y} = C_k \mid \mathbf{x} = C_j; \{\pi_j\}) = \pi_{jk}$$

where for all j , $\pi_{jk} \geq 0$ and $\sum_{k=1}^c \pi_{jk} = 1$

- What is the probability $p(\mathbf{y}|\mathbf{x})$ under a bigram distribution?
- Assume we have parameters for all possible inputs $\{\boldsymbol{\pi}_j\}$ and that C_j is the last word seen.
- Model as a categorical distribution:

$$p(\mathbf{y} = C_k \mid \mathbf{x} = C_j; \{\boldsymbol{\pi}_j\}) = \pi_{jk}$$

where for all j , $\pi_{jk} \geq 0$ and $\sum_{k=1}^c \pi_{jk} = 1$

Bigram Model

Bigram model represents the probability of the next word y given the last word x

$$p(y = C_k \mid x = C_j; \{\pi_j\}) = \pi_{jk}$$

$x \backslash y$	the	dog	cat	horse	...
the	0	0.2	0.2	0.1	
dog	0	0	0	0	
cat	0	0.05	0	0	
horse	0	0	0	0	
\vdots					

- Can extend \mathbf{x} to include multiple previous words.

- For instance a *trigram* model uses,

ON POLISHED ___

- For this model $\mathbf{x} \in \{0, 1\}^{2 \times |\mathcal{V}|}$.
- Note: this is different than bag-of-words, vector gets larger $[C_{j'}; C_j]$.

$$p(\mathbf{y} = C_k \mid \mathbf{x} = [C_{j'}; C_j]; \{\boldsymbol{\pi}_{j'j}\}) = \pi_{j'jk}$$

- Different weight vector for all pairs, $\boldsymbol{\pi}_{j'j}$

N-Gram Model

Bigram model represents the probability of the next word y given the last word x

$$p(y = C_k \mid \mathbf{x} = [C_{j'}; C_j]; \{\pi_{j'j}\}) = \pi_{j'jk}$$

$\mathbf{x} \backslash \mathbf{y}$	the	dog	cat	horse	...
on the	0	0.2	0.2	0.1	
by the	...				
with the	...				
near the	...				
⋮					

Learning N-Gram Models

Ingredients:

- 1 Corpus (e.g. the entire web)

Steps:

- (1) Collect words, (2) Count up n-grams, (3) Divide*

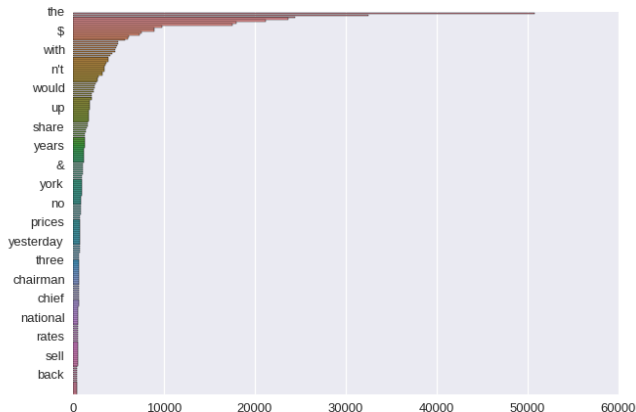
$$p(\mathbf{y}|\mathbf{x}) = \frac{\#(\mathbf{y}, \mathbf{x})}{\#(\mathbf{x})}$$

Number of token	1,024,908,267,229
Number of sentences	95,119,665,584
Size compressed (counts only)	24 GB

Number of unigrams	13,588,391
Number of bigrams	314,843,401
Number of trigrams	977,069,902
Number of fourgrams	1,313,818,354
Number of fivegrams	1,176,470,663

Zipf' Law (1935,1949)

The frequency of any word is inversely proportional to its rank in the frequency table.



Contents

- 1 Language Models
- 2 Supervised Learning
- 3 Neural Networks For Language**
- 4 Recurrent Neural Networks
- 5 Sequence-to-Sequence
- 6 Work at Harvard

Intuition: N-Gram Issues

In training we might see,

the arizona corporations commission **authorized**

But at test we see,

the colorado businesses organization ---

- Does this training example help here?
 - Not really. No count overlap.
- Intuition: hope to learn that similar words act similarly.

Intuition: N-Gram Issues

In training we might see,

the arizona corporations commission **authorized**

But at test we see,

the colorado businesses organization ---

- Does this training example help here?
 - Not really. No count overlap.
- Intuition: hope to learn that similar words act similarly.

Intuition: N-Gram Issues

In training we might see,

the arizona corporations commission **authorized**

But at test we see,

the colorado businesses organization ---

- Does this training example help here?
 - Not really. No count overlap.
- Intuition: hope to learn that similar words act similarly.

Consider now using neural networks for this prediction.

Two important ideas that we have seen so far.

1. Softmax for multiclass prediction of next word (out of \mathcal{V})
2. Adaptive basis to learn representation of past words

Predicting the Next Word

- Recall: softmax approach for multi-class classification.

$$p(\mathbf{y} = C_k | \mathbf{x}; \{\mathbf{w}_\ell\}, \mathbf{W}^1) = \frac{\exp(\mathbf{w}_k^\top \phi(\mathbf{x}; \mathbf{W}^1))}{\sum_\ell \exp(\mathbf{w}_\ell^\top \phi(\mathbf{x}; \mathbf{W}^1))}$$

- The score term, gives the score for each output word C_k conditioned on some representation of the input.

$$\mathbf{w}_k^\top \phi(\mathbf{x}; \mathbf{W}^1)$$

- Probability of word is softmax over these scores.
- Why might this be hard to compute?

Neural Networks

- Recall: Approach of neural networks has been to learn adaptive basis,

$$\phi(\mathbf{x}; \mathbf{w}) = \sigma(\mathbf{W}^1 \mathbf{x} + \mathbf{w}_0^1)$$

- What would this look like for a bigram model?

$$\mathbf{x} = C_j$$

$$\mathbf{W}^1 \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{W}_{*,j}^1$$

- Returns just a column vector of the matrix.

Interpretation of Network

- Notation input \mathbb{R}^m and basis \mathbb{R}^d .
- Unlike other problems here m is large $|\mathcal{V}|$ (10,000-100,000)
- But d can be much smaller. Why is that?

$$\mathbf{W}^1 \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{W}_{\star,j}^1$$

- The output of this layer $\mathbf{W}_{\star,j}^1 \in \mathbb{R}^d$ is called a *word vector*.

Sparse versus Distributed Representation

- Internally we have now seen two representations of words
- \mathbf{x} Sparse: Very high dimensional, easy to get back the original word, but feature weights are not shared between words. w_j is weight for one word.

$$[0, 0, 0, 1, \dots 0]$$

- $\phi(\mathbf{x})$ Distributed: Low dimensional but dense, each dimension is a feature of the word, w_j is weight for basis function.

$$[0.23, 0.32, 0.109, -0.1231, \dots, 0.402]$$

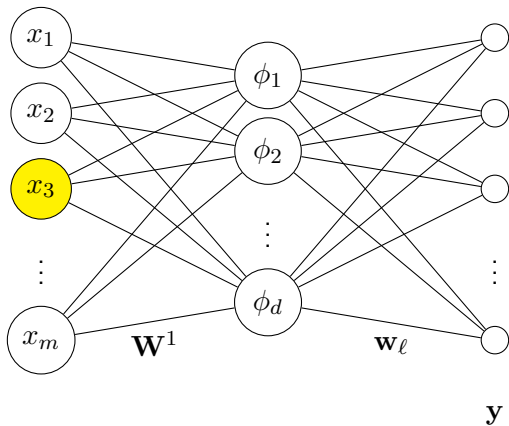
Sparse versus Distributed Representation

- Internally we have now seen two representations of words
- \mathbf{x} Sparse: Very high dimensional, easy to get back the original word, but feature weights are not shared between words. \mathbf{w}_j is weight for one word.

$$[0, 0, 0, 1, \dots 0]$$

- $\phi(\mathbf{x})$ Distributed: Low dimensional but dense, each dimension is a feature of the word, \mathbf{w}_j is weight for basis function.

$$[0.23, 0.32, 0.109, -0.1231, \dots, 0.402]$$



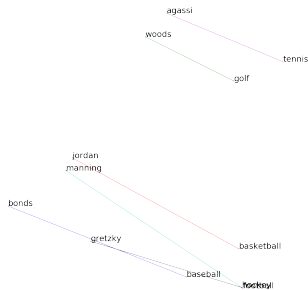
Depiction of the neural network. First layer is one-hot with $m = |\mathcal{V}|$. Hidden layer has $d \ll m$. Last layer is the score given to each possible output word. Original representation is sparse, and basis layer gives a *dense* representation of \mathbf{x} .

Word Vectors In Practice

- Word2Vec: famous pre-trained word vectors trained on Google News.
- Famous for learning a “linear” representation of words in \mathbb{R}^d

$$\phi(\text{king}) - \phi(\text{man}) + \phi(\text{woman}) \approx \phi(\text{queen})$$

- Vectors extracted from \mathbf{W}^1 after training on language modeling.



Linear substructure of word vectors

Example

Contents

- 1 Language Models
- 2 Supervised Learning
- 3 Neural Networks For Language
- 4 Recurrent Neural Networks
- 5 Sequence-to-Sequence
- 6 Work at Harvard

*THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
READING LAMP ON THE DESK SHED GLOW ON
POLISHED ---*

Recall model is,

$$p(\mathbf{y}|\mathbf{x})$$

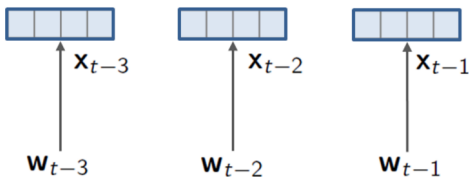
- So far we have assumed input \mathbf{x} is fixed size n-grams.
- But with deep neural network basis we do not need to do this.
- *Recurrent neural networks* use all the context words.

Recurrent Neural Network Language Model

Word Vector	sparse input	\Rightarrow	dense basis
RNNs	sequences of basis	\Rightarrow	dense basis
Softmax	dense basis	\Rightarrow	discrete predictions

Word Vectors

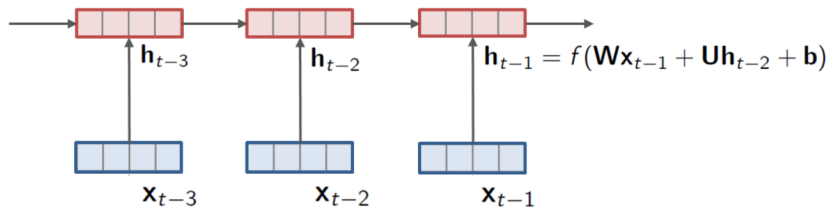
sparse input \Rightarrow dense basis



- First layer of the network maps from one-hot words to word vectors (using layer shown above).

RNNs/LSTMs

sequences of basis functions \Rightarrow dense basis

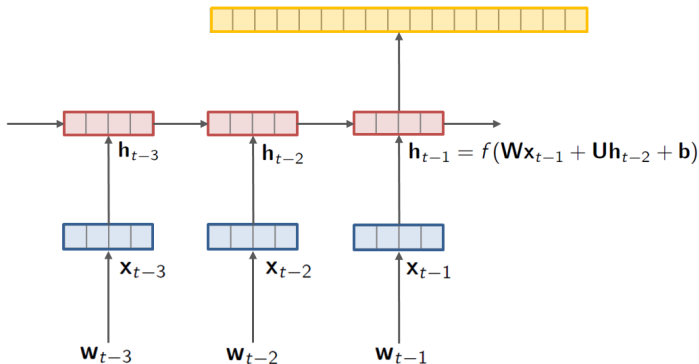


- Each word vector is connected together in a chain over time.

-

$$\phi(\mathbf{x}, \mathbf{h}) = \sigma(\mathbf{W}^1 \mathbf{x} + \mathbf{W}'^1 \mathbf{h} + \mathbf{w}_0)$$

LM/Softmax

dense features \Rightarrow discrete predictions

$$p(y = C_k | \mathbf{x}; \{\mathbf{w}_\ell\}, \mathbf{W}^1) = \frac{\exp(\mathbf{w}_k^\top \phi(\mathbf{x}; \mathbf{W}^1))}{\sum_\ell \exp(\mathbf{w}_\ell^\top \phi(\mathbf{x}; \mathbf{W}^1))}$$

- Here $\phi(\mathbf{x})$ encapsulates the red and blue parts of the network.

Contents

- 1 Language Models
- 2 Supervised Learning
- 3 Neural Networks For Language
- 4 Recurrent Neural Networks
- 5 Sequence-to-Sequence**
- 6 Work at Harvard

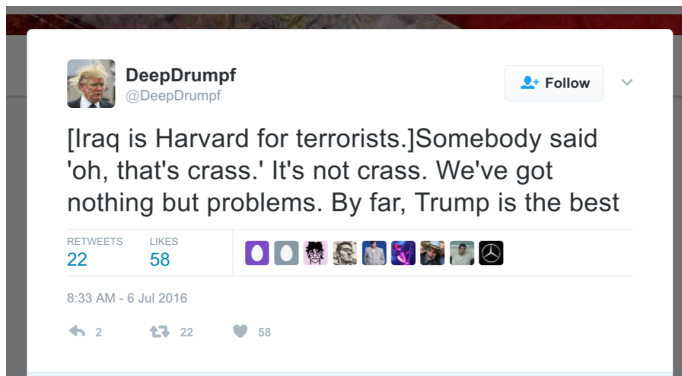
RNN models have lead to a **major** increase in the accuracy of language models.

Why did this matter?

Application of RNNs

RNN models have lead to a **major** increase in the accuracy of language models.

Why did this matter?



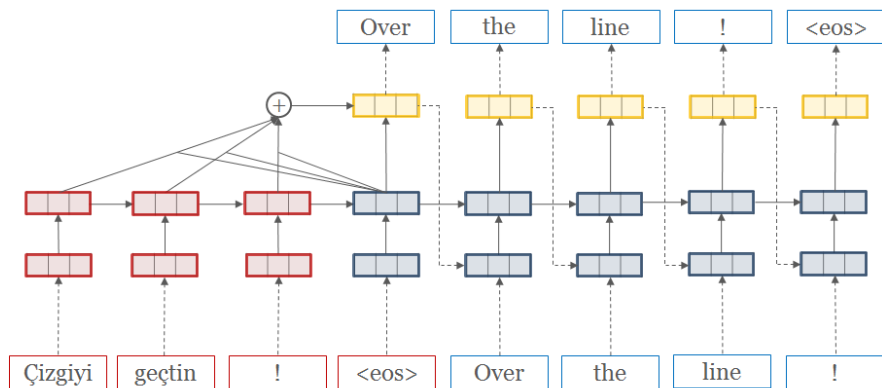
Sequence-to-Sequence

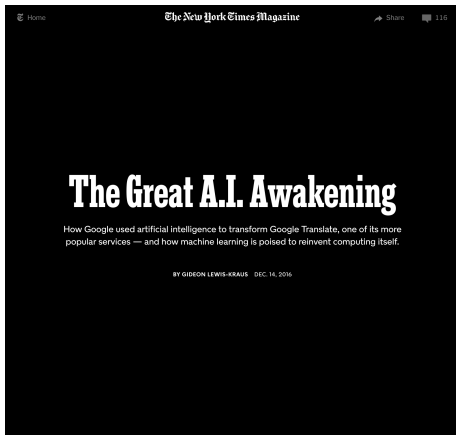
- Instead of just doing language model, also have basis functions for another input.
- For example, French-to-English translation:
 - Given a French sentence and previous English words, predict next English word

$$p(\mathbf{y}|\mathbf{x})$$

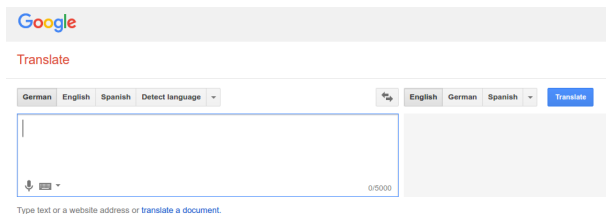
Where \mathbf{y} is next English word.

Sequence-to-Sequence





Google Translate



The screenshot shows the Google Translate web interface. At the top is the Google logo. Below it is the word "Translate" in red. The interface features two language selection dropdown menus: one on the left for the source language (currently set to "German") and one on the right for the target language (currently set to "English"). A "Detect language" button is also present. A large text input box is on the left, and a corresponding output box is on the right. A "Translate" button is located to the right of the output box. At the bottom of the input box, there is a microphone icon and a character count "0/5000".

Type text or a website address or [translate a document](#).

- Completely replaced their machine translation systems.
- Accuracy significantly increased
- Reduced code size tremendously (rumor around 1K).

Applications

- Machine Translation
- Question Answering
- Conversation
- Parsing
- Speech
- Caption Generation
- Video-Generation
- NER/POS-Tagging
- Summarization

Contents

- 1 Language Models
- 2 Supervised Learning
- 3 Neural Networks For Language
- 4 Recurrent Neural Networks
- 5 Sequence-to-Sequence
- 6 Work at Harvard

- Our open-source system <http://opennmt.net>

Text Translation

This demo platform allows you to experience Pure Neural™ machine translation based on the last Research community's findings and SYSTRAN's R&D. You can translate up to 2000 characters of text in the languages proposed below. Check out the [information page](#) to learn more.

English

×

↔

German

▼

Filter ⓘ

Select a profile

Translation on the internet

Übersetzung im Internet

Showing results for: Translation c

translation [ˈtrænzˈleɪʃən]

Übersetzung [ˈʊbɐˈzɛʃʊŋ]

(↔ interpretation)

english translation

certified translation

French translation

machine translation

on [ɒn] adv

darüber [ˈdaraʊ̯bɐ]

(↔ over)

spät [ʃpɛt]

(↔ late, subsequently)

daran [ˈdara]

(↔ most)

danach [ˈdanach]

[Demo]

Source (First Sentence)

Russian Defense Minister Ivanov called Sunday for the creation of a joint front for combating global terrorism.

Target (Title)

Russia calls for joint front against terrorism.

- Used by Washington Post to suggest headlines

Source (First Sentence)

Russian Defense Minister Ivanov called Sunday for the creation of a joint front for combating global terrorism.

Target (Title)

Russia calls for joint front against terrorism.

- Used by Washington Post to suggest headlines

Source (Original Sentence)

There is no a doubt, tracking systems has brought many benefits in this information age .

Target (Corrected Sentence)

There is no doubt, tracking systems have brought many benefits in this information age .

- 1st on BEA'11 grammar correction task

Applications: Im2Markup (Deng et al, 2017)

The diagram illustrates the LaTeX rendering process for the equation $r = \frac{\sqrt{Q_3}}{l} \sin\left(\frac{l}{\sqrt{Q_3}u}\right)$. Dashed lines connect the LaTeX code to the corresponding parts of the rendered equation:

- `r` points to the variable r .
- `=` points to the equals sign.
- `\frac` points to the fraction bar.
- `\sqrt` points to the square root symbol.
- `Q_3` points to the term Q_3 under the square root.
- `l` points to the denominator l .
- `\sin` points to the sine function.
- `\left` points to the left parenthesis.
- `\frac` points to the fraction bar inside the sine function.
- `l` points to the numerator l inside the sine function.
- `\sqrt` points to the square root symbol inside the sine function.
- `Q_3` points to the term Q_3 under the square root inside the sine function.
- `u` points to the denominator u inside the sine function.
- `\right` points to the right parenthesis.

[Latex Example]

[Project]