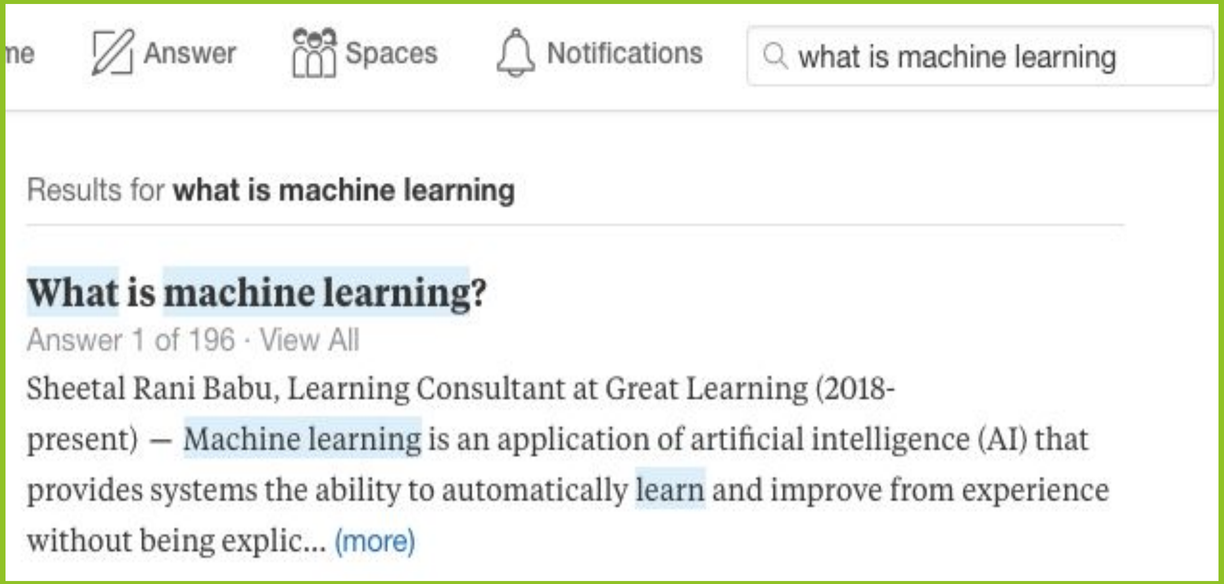
Quora Duplicate Questions Identification

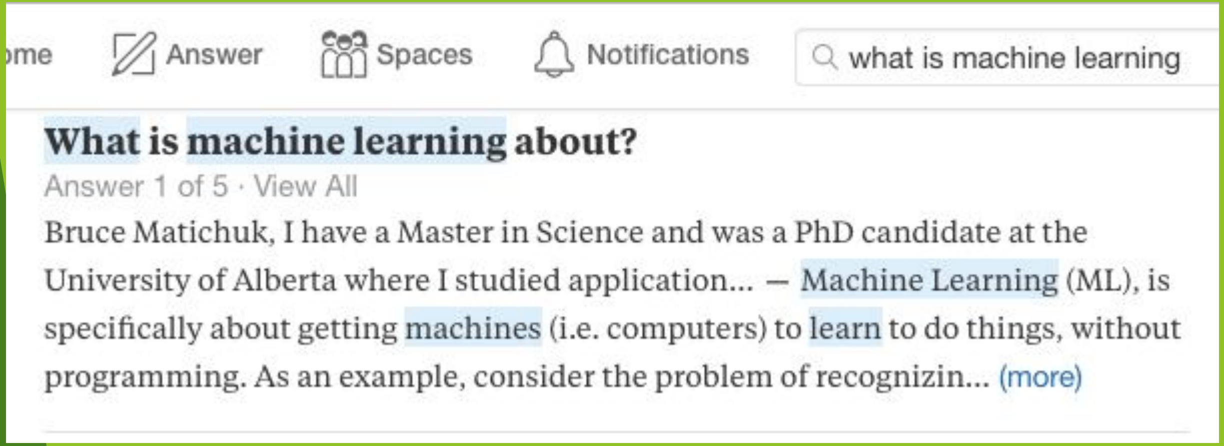
Amrita Sharma, Preethi Ranganathan

Table of Contents

Background

- ➢ Quora platform to ask questions, connect, contribute insights and quality answers
- ➢ 100 million people visit Quora every month
- ➢ Place to gain and share knowledge
- ➢ Empowers people to learn
- ➢ Many people ask similar worded questions
- ➢ Writers answer multiple versions of same question





Problem Definition

- The problem is to identify whether a given question is duplicate of another or not i.e. two questions contain the same meaning
- Binary Classification problem (Target variable - 0 or 1)

$$f(\text{Question 1}, \text{Question 2}) \rightarrow 0 \text{ or } 1$$

Data

- Data contains 400K observations
- Duplicates - 40% & Non-duplicates - 60%
- Dropped NA's

	id	qid1	qid2	question1	question2	is_duplicate
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and cap rising...what does that say about me?	I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) What does this say about me?	1
7	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
11	11	23	24	How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1

Duplicates

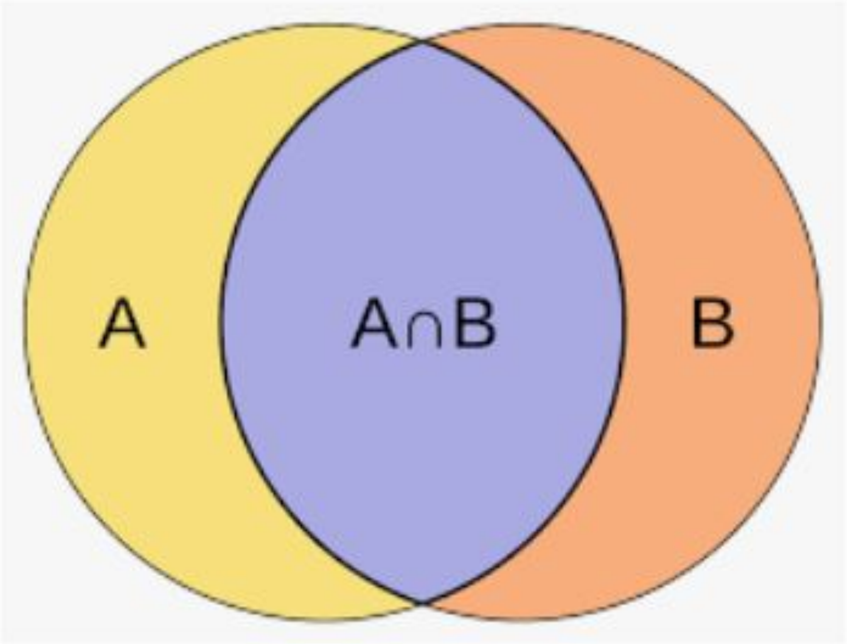
	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market?	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Diamond?	What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back?	0
2	2	5	6	How can I increase the speed of my internet connection while using a VPN?	How can Internet speed be increased by hacking through DNS?	0

Non-Duplicates

Jaccard Similarity

- Simplest similarity measure
- Proportion of common words between two questions
- <u>No. of Common Words</u>

  Total No. of Unique Words
- Range - (0,1)
- Semantic information not captured
- Threshold set to 0.5



Metrics Jaccard Similarity

- Example of Duplicates not captured by Jaccard

How can I be a good geologist?	What should I do to be a great geologist?
How do I read and find my YouTube comments?	How can I see all my Youtube comments?

Accuracy	0.6451
Precision	0.3227
Recall	0.5320
F1 Score	0.4018

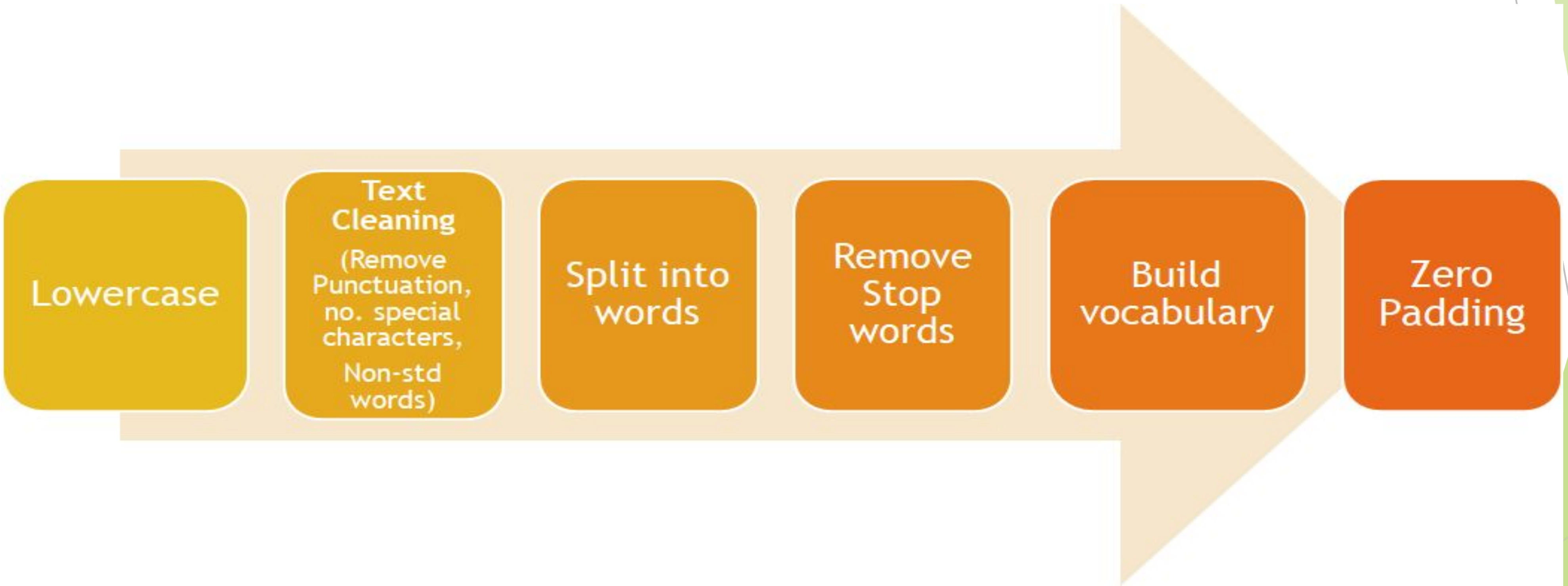
Cosine Similarity

- Cosine of the angle(θ) between two vectors.
- Questions represented as high dimensional, sparse tf-idf vectors (Term Frequency - Inverse Document Frequency) followed by cosine similarity computation.
- Higher the value of cos(θ) higher the similarity
- Threshold set to 0.5

Accuracy	0.6598
Precision	0.525
Recall	0.7988
F1 Score	0.6342

Deep Learning Models for NLP

Data Prep



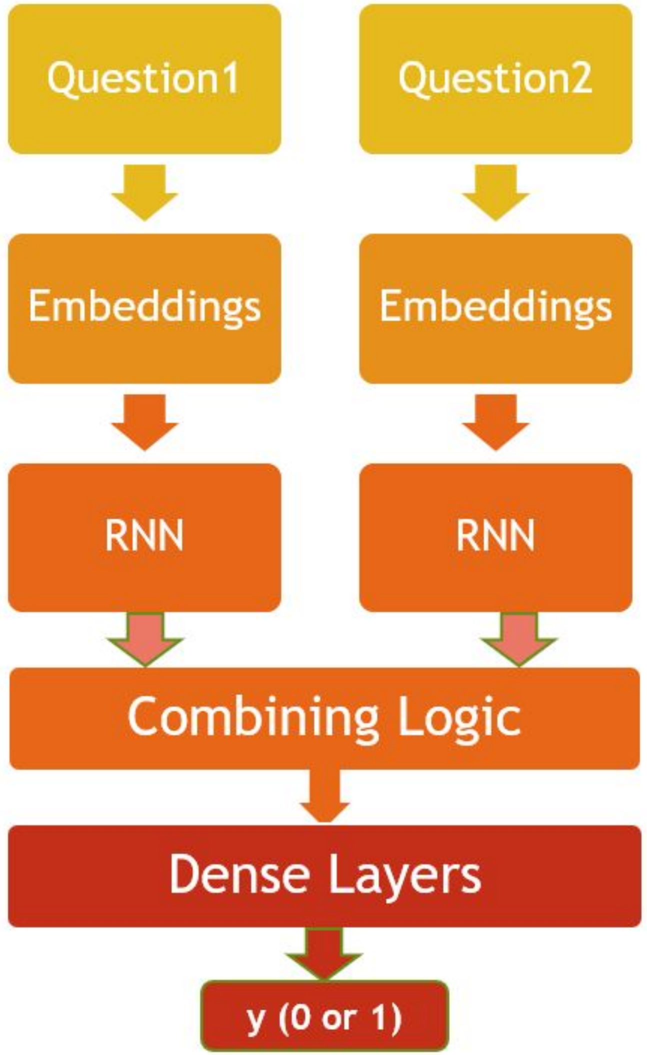
Lowercase
Text Cleaning
(Remove Punctuation, no. special characters,
Non-std words)
Split into words
Remove Stop words
Build vocabulary
Zero Padding

Word Embeddings

- Word embeddings are meant to map words into a geometric space
- Provides richer representations expressing semantic similarity
- Produce dense vector representations based on context/use of words
- Pre-trained Embeddings
  - Word2Vec
  - Glove

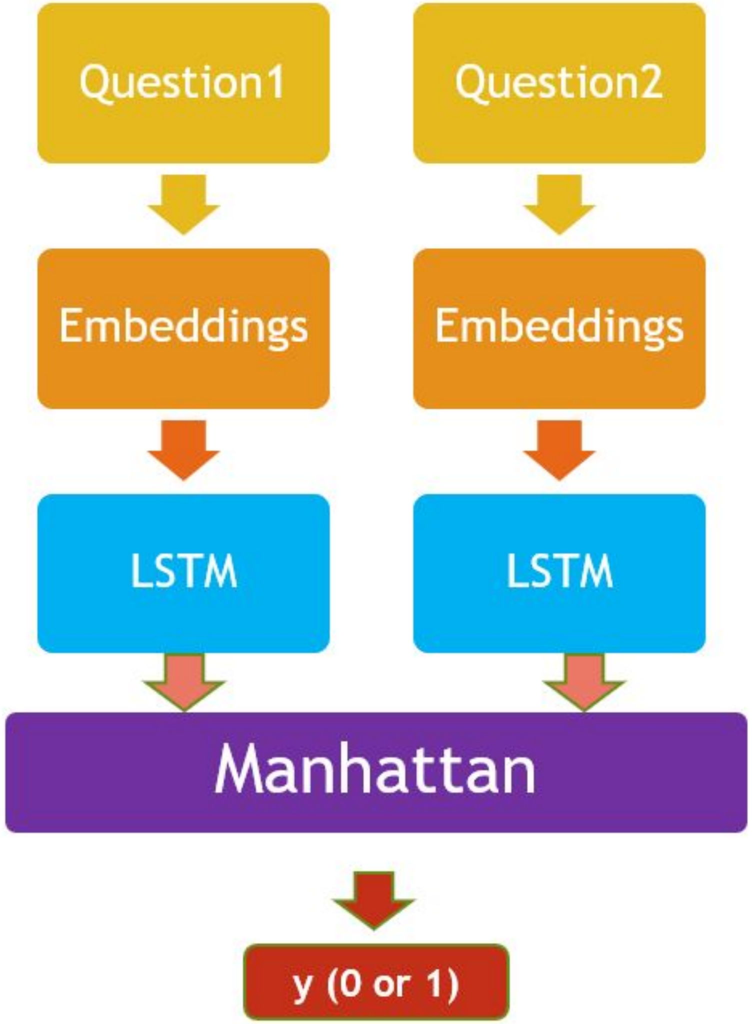
Siamese Networks Architecture

- Our inputs (questions) are of the same kind, we used similar models to process both inputs.
- Siamese networks - two or more identical sub-networks in them.
- Shared weights across subnetworks.
- Results in fewer parameters.
- Performs well on similarity tasks like sentence semantic similarity, recognizing forged signatures, etc.



LSTM Models

- LSTM better at capturing long-term dependencies
- Provides sentence representations that captures rich semantics
- Output of the LSTM for each question is a 50-dimensional vector
- Combining Logic - Negative Exponential of Manhattan Distance
  - Simple distance measure to compare feature vectors.
  - Output ranges from 0 to 1
- Threshold - 0.5
- Loss function - Mean Squared Error

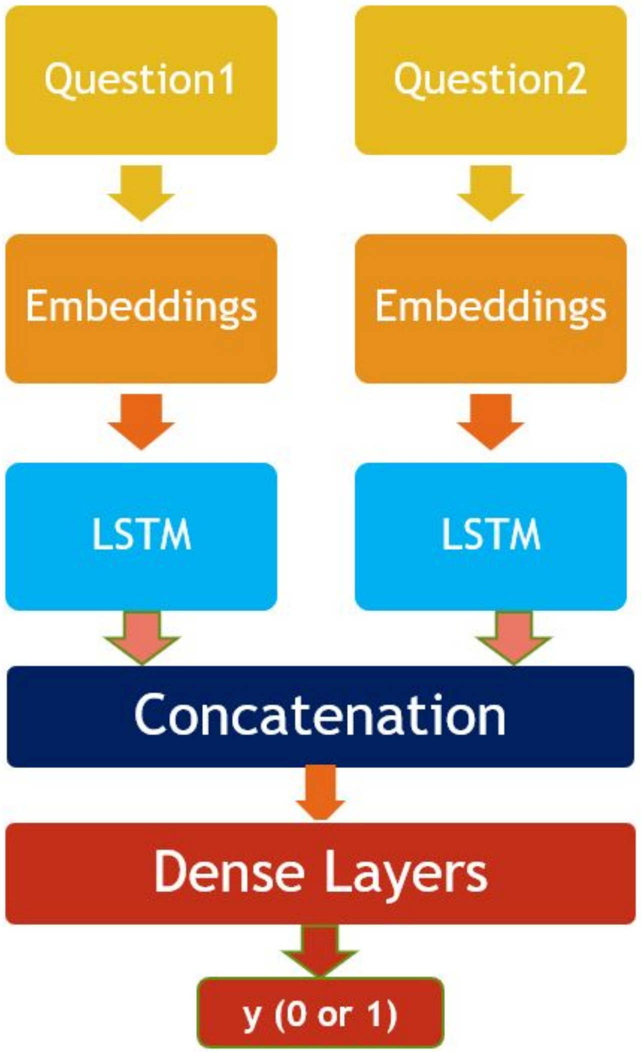


LSTM Results

Number of layers	Embedding	Optimizer	Number of training samples	Number of validation samples	Training accuracy	Validation accuracy	Regularization	Overfitting
2	Word2vec	Adam	9000	1000	0.9447	0.765	None	Yes
2	Word2vec	Adam	40000	10000	0.9333	0.7793	None	Yes
1	Glove	Adam	40000	10000	0.92	0.78	None	Yes
1	Glove	Adam	75000	25000	0.886	0.793	None	Yes
2	Word2vec	Adam	40000	10000	0.6615	0.6618	L1L2(0.01.0.01)	No
1	Word2vec	Adam	110000	40000	0.83	0.795	None	No

Combining Logic Variation - Concatenation

- Output of LSTM for each question is a 50-dimensional vector. When concatenated, results in a 100 dimensional vector
- 100 Dim vector fed into the dense layer to perform classification
- Used one dense layer
  - Activation function - Sigmoid
- Loss Function - Binary cross entropy

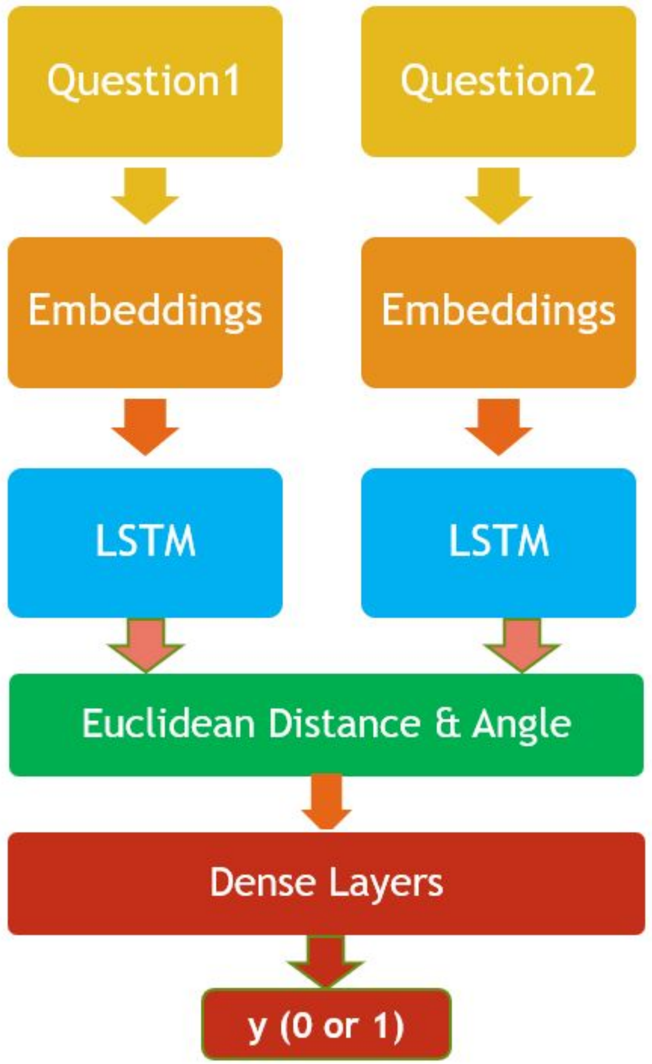


LSTM Concatenation - Results

No. of layers	Embedding	Optimizer	Number of training samples	Number of validation samples	Training accuracy	Validation accuracy	Overfitting
2	Word2vec	Adam	60000	10000	0.9578	0.7554	Yes
1	Word2vec	Adam	60000	10000	0.9143	0.7188	Yes
1	Word2vec	Adam	100000	25000	0.8972	0.7336	Yes
1	Glove	Adam	150000	25000	0.949	0.736	Yes
1	Word2vec	Adam	175000	25000	0.9071	0.75	Yes

Combining Logic Variation - Euclidean Distance and Angle

- Euclidean distance and Angle(dot product) between RNN representations of questions
- 50 dim vector fed into two dense layers
  - 1st layer - 32 nodes
  - 2nd layer - 1 node with sigmoid activation function
- Loss Function - Binary Cross Entropy

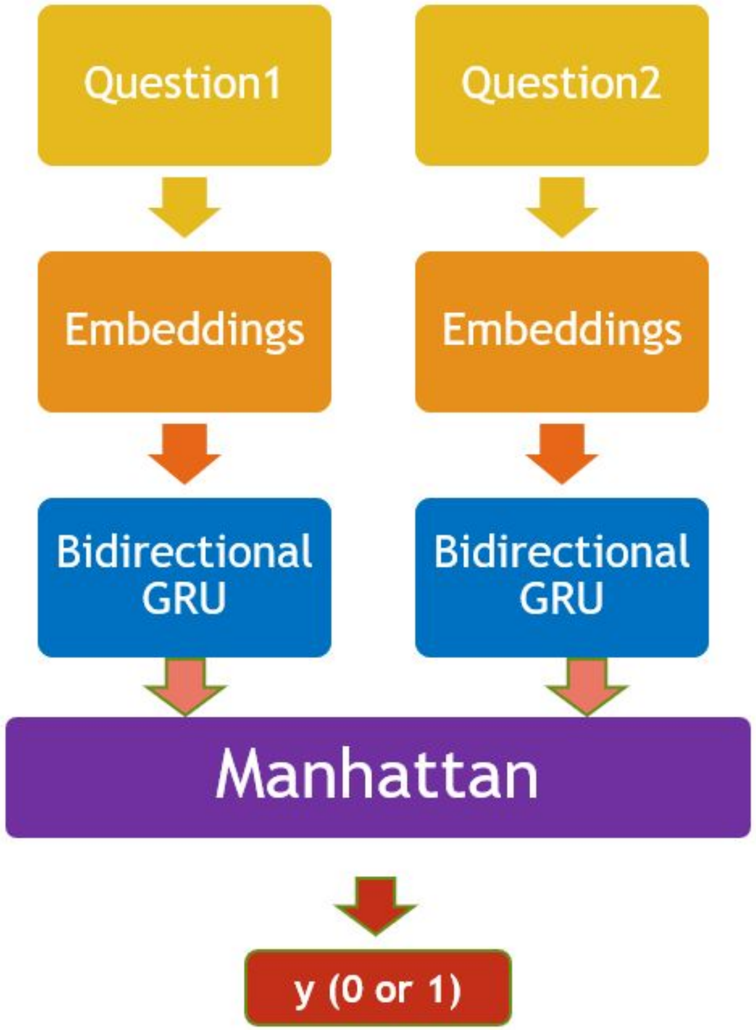


LSTM with Euclidean Angle and Distance Results

Number of layers	Embedding	Optimizer	Number of training samples	Number of validation samples	Training accuracy	Validation accuracy	Overfitting
1	Word2vec	Adam	80000	20000	0.9356	0.796	Yes
1	Glove	Adam	150000	25000	0.95	0.8	Yes

Bidirectional GRU Networks

- Gated Recurrent Units(GRU)- simplified version of LSTM
- GRUs have fewer parameters
  - Trains a bit faster
  - Need less data to generalize.
- Bidirectionality enables the network to understand the context and eliminate ambiguity
- Early Stopping based on validation loss



Bidirectional GRU Results

Number of layers	Embedding	Optimizer	Number of training samples	Number of validation samples	Training accuracy	Validation accuracy	Combining Logic	Overfitting
1	Word2vec	Adam	40000	10000	0.8522	0.7917	Manhattan	Yes
1	Word2vec	Adam	70000	10000	0.8436	0.7882	Manhattan	Yes
1	Glove	Adam	150000	25000	0.9	0.792	Manhattan	Yes
1	Word2vec	Adam	150000	25000	0.8412	0.8028	Manhattan	No
1	Word2vec	Adam	150000	25000	0.915	0.804	Euclidean	Yes
1	Word2vec	Adam	200000	30000	0.8566	0.8171	Manhattan	No

Bidirectional GRU with Custom-word Embeddings

- Capture Embeddings relevant to the task
- Initialize model with word2Vec embeddings
- Embeddings updated during the training process

No. of layers	Embedding	Optimizer	Number of training samples	Number of validation samples	Training accuracy	Validation accuracy	Regularization	Overfitting	Combining model outputs
1	Custom Word Embedding	Adam	300000	25000	0.9053	0.805	l2(0.001)	Yes	Manhattan
1	Custom Word Embedding	Adam	300000	25000	0.8836	0.82	l2(0.005)	No	Manhattan

Model Comparison - Results

Metrics	Jaccard Similarity	Cosine Similarity	Deep Learning Model (Bi-directional GRU) Test
Accuracy	0.6451	0.6598	0.8216
Precision	0.3227	0.525	0.7531
Recall	0.5320	0.7988	0.7710
F1 Score	0.4018	0.6342	0.7619

Conclusion & Key Takeaways

- Best model - Test accuracy - 82.16% and an F-1 Score - 76.19%
  - Bidirectional GRU (Hidden States: 50)
  - Custom word embedding
  - Combining Logic - Manhattan
  - L2 Regularization (0.005)
  - Adam Optimizer (Learning Rate: 0.001)
  - Early Stopping @ Epoch 10
  - Number of Layers - 1
  - Batch size- 64
- Classify 82% of the test samples accurately
- Custom word embeddings helped us to identify embeddings more relevant to the task
- Increasing the number of training samples reduced overfitting
- Capturing semantic similarity increased accuracy

Limitations & Future work

- Computational Power
  - Environment - Google Cloud
  - GPU-1 & CPU - 8
  - Our Final bidirectional GRU model with custom embeddings ran for 10 hours
  - Each LSTM model took 3-4 hours to run
  - Each bidirectional GRU model took 7-8 hours to run
- Attention mechanisms can be used with token alignment
- Fine tune models by searching for the best threshold for classification
- Use ensemble learning methods

Thank you!