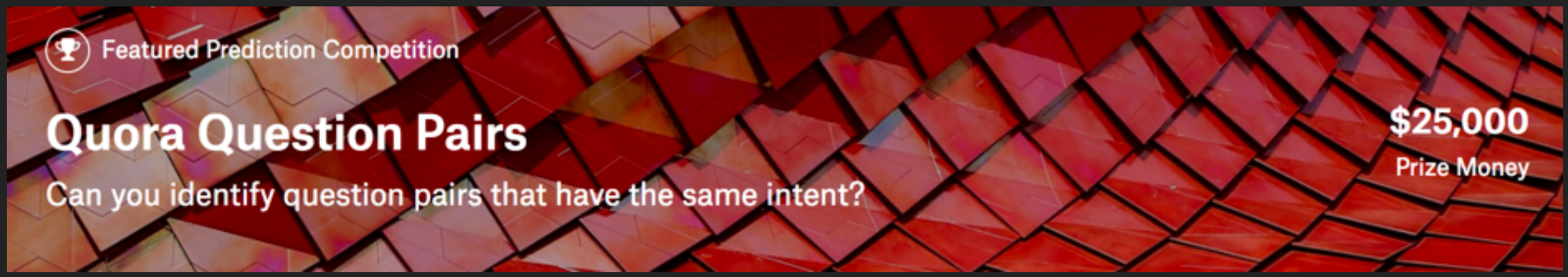
Maximilien BAUDRY




WINNING SOLUTION

# KAGGLE QUORA

# SUMMARY

# INTRODUCTION

- What is Quora?
  - World's biggest forum
  - Best place to share general knowledge
  - Topics are designed to only ask questions
- Problem
  - People may ask similar questions
  - Important interest to detect duplicated questions
- **Prediction problem : from a question pair, predict whether questions are the same or not**
- Metric: LogLoss

$$-\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right]$$

# PRESENTATION OF TEAM DL (DATA'LAB) GUYS

- Sebastien Conort, chief data scientist BNPP Cardif
- Lam Dang, data scientist, BNPP Cardif
- Guillaume Huard, data scientist, BNPP Cardif
- Paul Todorov, data scientist, BNPP Cardif
- Maximilien Baudry, PhD student, SAF lab, DAMI (Data Analytics and Models for Insurance) chair of research

In the money Gold Silver Bronze

| # | △pub | Team Name | Kernel | Team Members | Score | Entries | Last |
|---|---|---|---|---|---|---|---|
| 1 | — | DL guys | | | 0.11580 | 263 | 8d |
| 2 | — | Depp Learning | | | 0.11670 | 196 | 8d |
| 3 | — | Jared Turkewitz & sjv | | | 0.11756 | 178 | 8d |
| 4 | — | YesOfCourse | | | 0.11768 | 189 | 8d |
| 5 | — | Qingchen \| KazAnova \| Faron | | | 0.11851 | 219 | 8d |
| 6 | — | LAMAA power | | | 0.11887 | 406 | 8d |
| 7 | ▲2 | aphex34 | | | 0.12072 | 166 | 8d |
| 8 | — | NLPFakers | | +3 | 0.12239 | 250 | 8d |
| 9 | ▼2 | Unduplicated Duplicates | | +4 | 0.12248 | 314 | 8d |
| 10 | ▲1 | ♬ ♪ b.a.s.s. ♩ ♫ | | | 0.12296 | 271 | 8d |

# DATA OVERVIEW

| | question1 | question2 |
|---|---|---|
| **0** | What are some good movies to watch? | What are the best movies to watch? |
| **1** | Do dentists earn more than other doctors? | Do dentists earn more than other doctors? Why? |
| **2** | Should I wait for iPad Air 3 or purchase the iPad Air 2? | Should I buy the iPad Air or wait for the next iPad Air (iPad Air 2)? |
| **3** | What is the difference between Java and Android programming? | Are there major differences between programming in Android vs plain Java? |
| **4** | Why do you yawn when you are tired? | Why do we yawn when we are sleepy? |
| **5** | Who is Benjamin Netanyahu? | Why is Benjamin Netanyahu famous? |

# DATA OVERVIEW

| | question1 | question2 | is_duplicate |
|---|---|---|---|
| **0** | What are some good movies to watch? | What are the best movies to watch? | 0 |
| **1** | Do dentists earn more than other doctors? | Do dentists earn more than other doctors? Why? | 0 |
| **2** | Should I wait for iPad Air 3 or purchase the iPad Air 2? | Should I buy the iPad Air or wait for the next iPad Air (iPad Air 2)? | 0 |
| **3** | What is the difference between Java and Android programming? | Are there major differences between programming in Android vs plain Java? | 1 |
| **4** | Why do you yawn when you are tired? | Why do we yawn when we are sleepy? | 1 |
| **5** | Who is Benjamin Netanyahu? | Why is Benjamin Netanyahu famous? | 1 |

# DATA OVERVIEW

- Duplicates proportion: 36.9% in train, 17.4% in test
- Number of question pairs: ~400k in train, ~2,3M in test
- ~80% of test dataset contains fake question pairs, such that we can't hand label test question pairs (avoid cheating)
- ~530k unique questions in train dataset
- ~110k questions appear multiple times in train and test datasets
- Questions which contains:
  - Question mark: 99.87%
  - [math] tags: 0.12%
  - Capitalized first letter: 99.81%
  - Capital letters: 99.95%
  - Numbers: 11.83%

| | question1 | question2 |
|---|---|---|
| **0** | What | How |
| **1** | Is there move? | Is format immortality? |
| **2** | What are exactly? | How does akamai great money? |
| **3** | How cpu insomnia diagnosed? | How ssc is insomnia treated? |

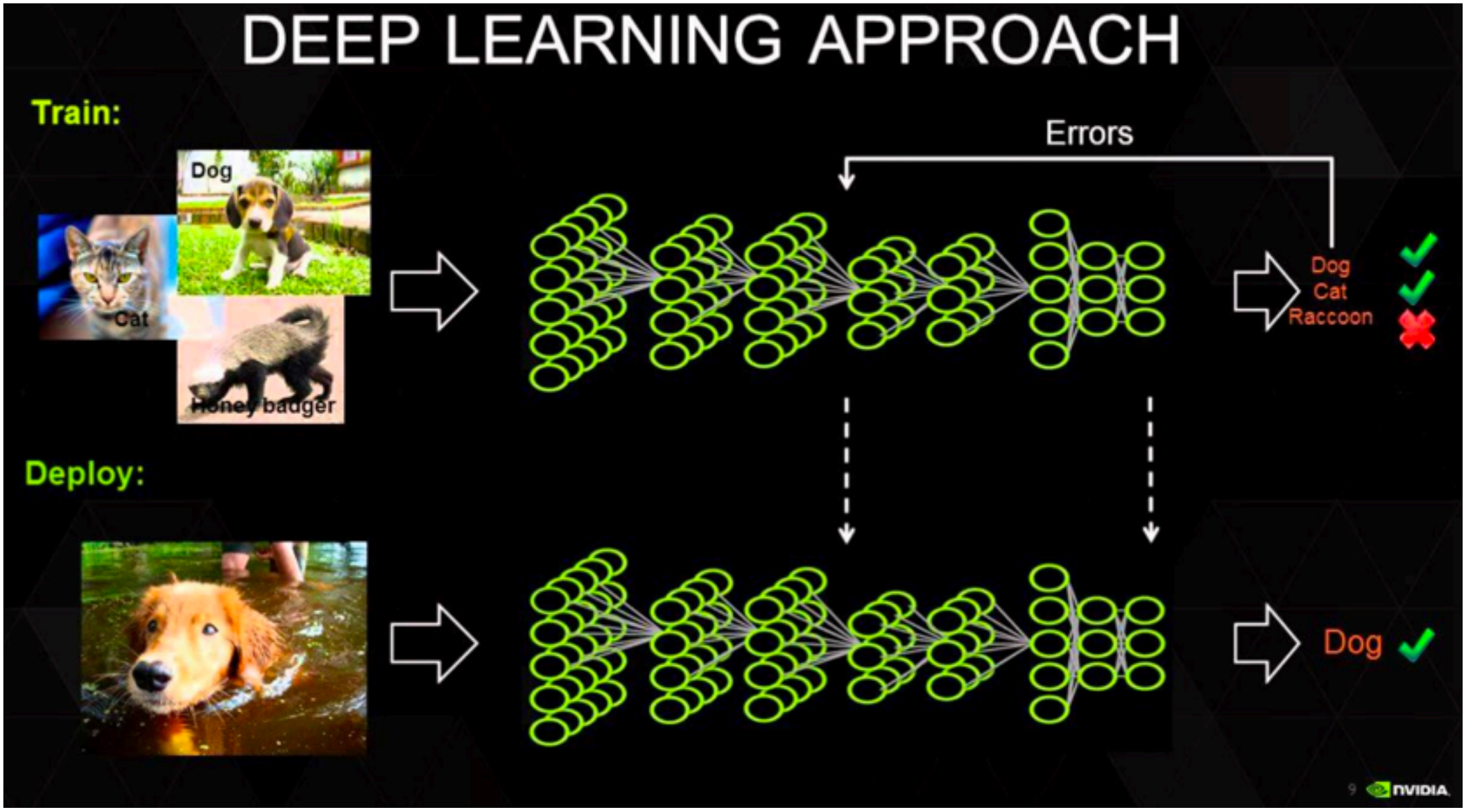
Examples of fake pairs

# OUR APPROACHES

- 3 main axes:

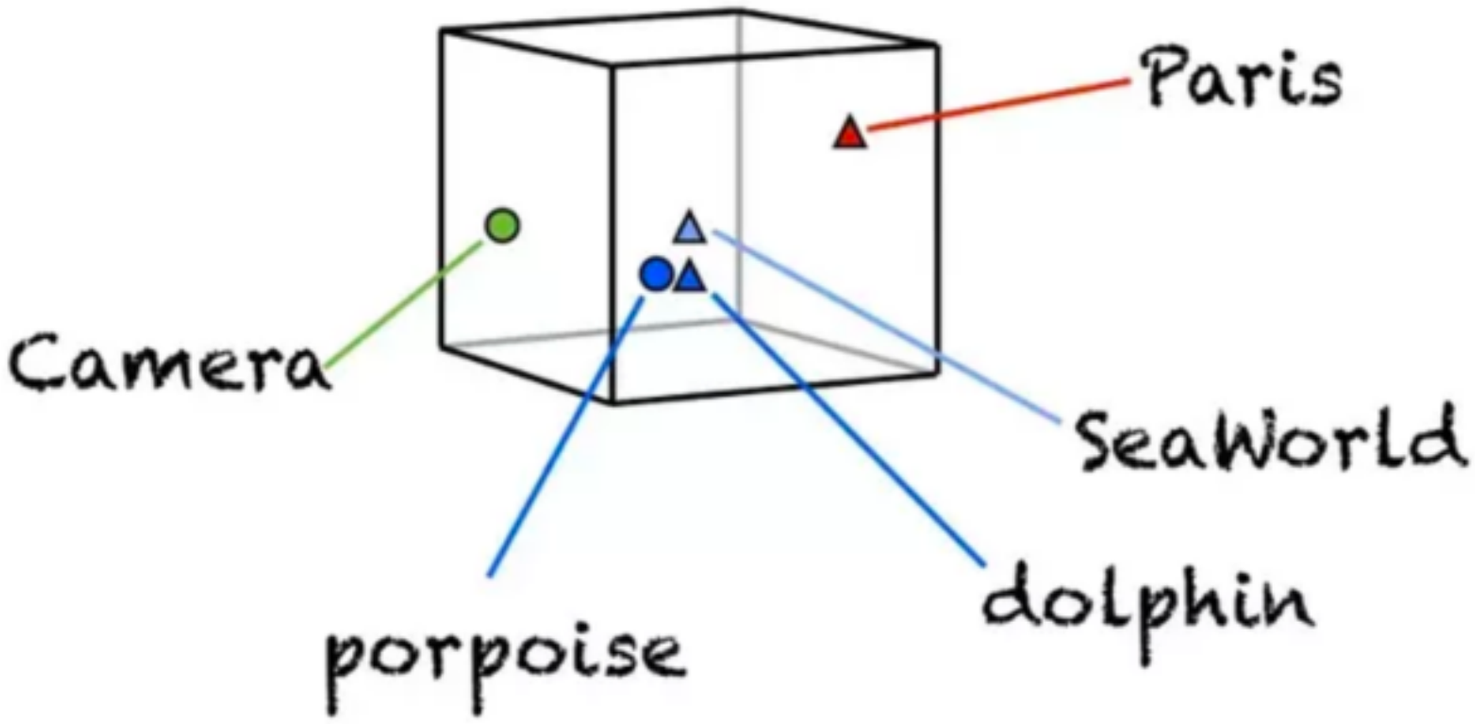
    1. Deep learning
    2. Neuro-Linguistic Programming (NLP)
    3. Graphical models

# FIRST APPROACH: DEEP LEARNING



# FIRST APPROACH: DEEP LEARNING

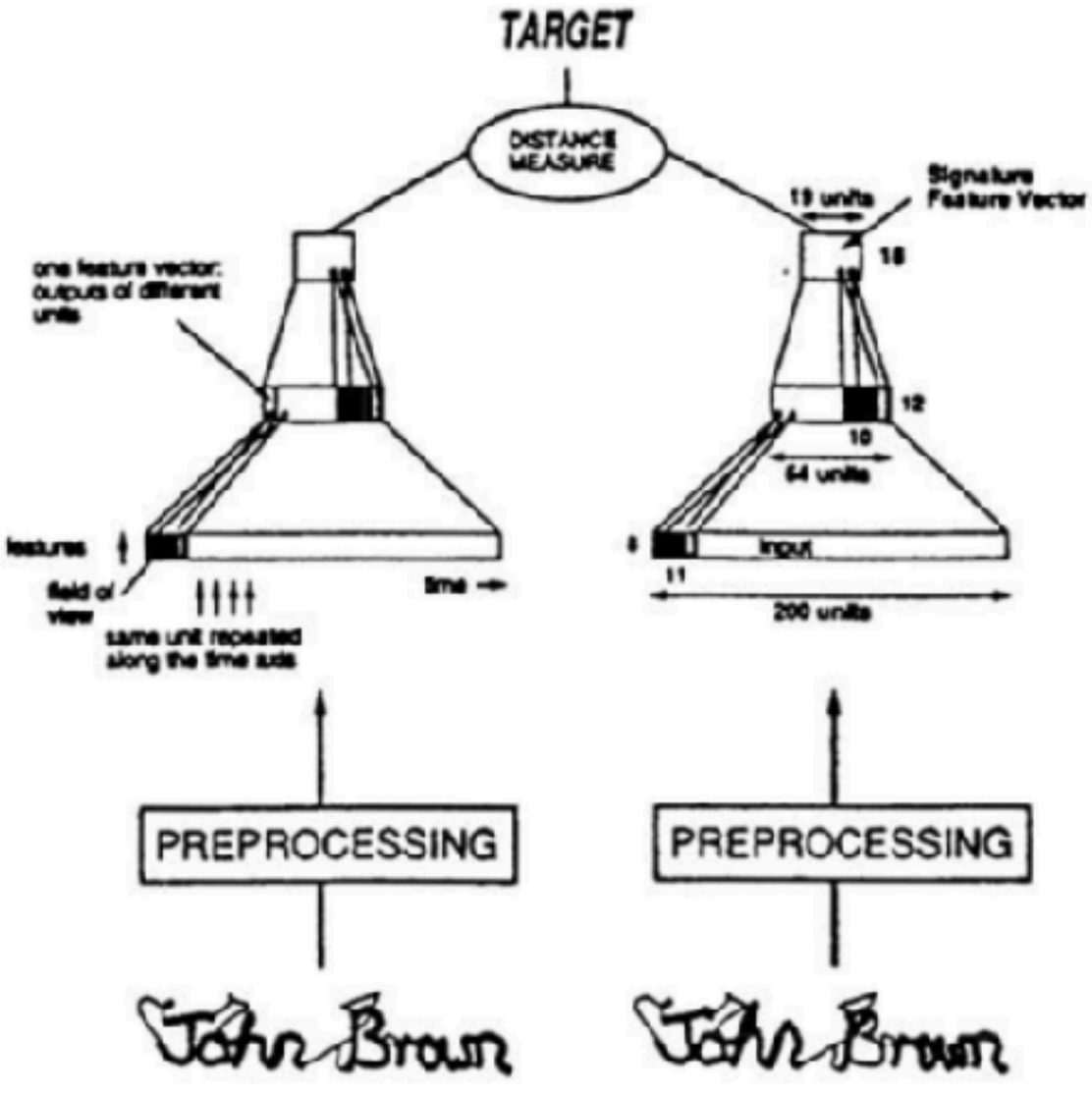
- Embedding of each questions
  - Word2Vec
  - Doc2Vec
  - Sent2Vec
- What is word embedding ?
  - Projection of each word/document/sentence in a very high dimensional space (we fixed dimension at 300)
  - In this space, each word is given coordinates such that words with common sense are close one an other
- Python library Gensim, pre-trained by Google
- Allows the following: PARIS - FRANCE + ENGLAND –> LONDON



# FIRST APPROACH: DEEP LEARNING

- Siamese Neural Network:
  - Two parallel networks
  - Same weights are trained with two inputs.
  - Dense layer to connect the two nets
  - Softmax activation on dense layer

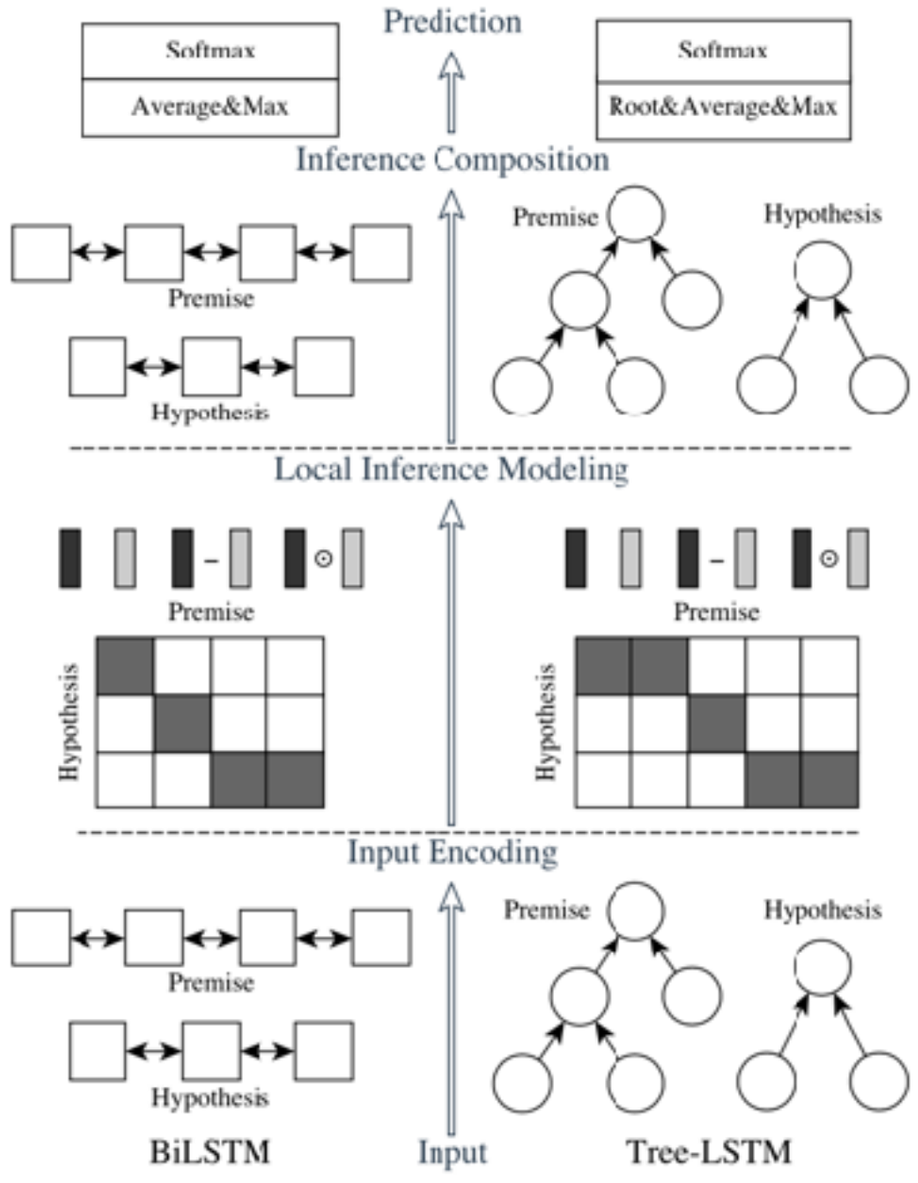
$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$




Siamese network illustration

# FIRST APPROACH: DEEP LEARNING

- Decomposable attention Neural Network:
  - (https://arxiv.org/abs/1606.01933)
  - Learn on word alignments
  - Detection of contradictory sentences
- ESIM
  - (https://arxiv.org/abs/1609.06038)




ESIM illustration

# SECOND APPROACH: NLP

- Similarity measures on LDA (Latent Dirichlet Allocation) and LSI (Latent Semantic Indexing) measures.
- Similarity measures on bags of character n-grams (TFIDF reweighed or not) from 1 to 8 grams.
- A lot of edit distance between character strings, such as Levenshtein distance, Jaro-Winkler distance, Bray-Curtis distance etc…
- Percentage of common tokens sized from 1 to 6, when question ends the same. Same work when questions starts the same.
- Length of questions, difference of length
- Number of capital letters, question marks, etc…

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

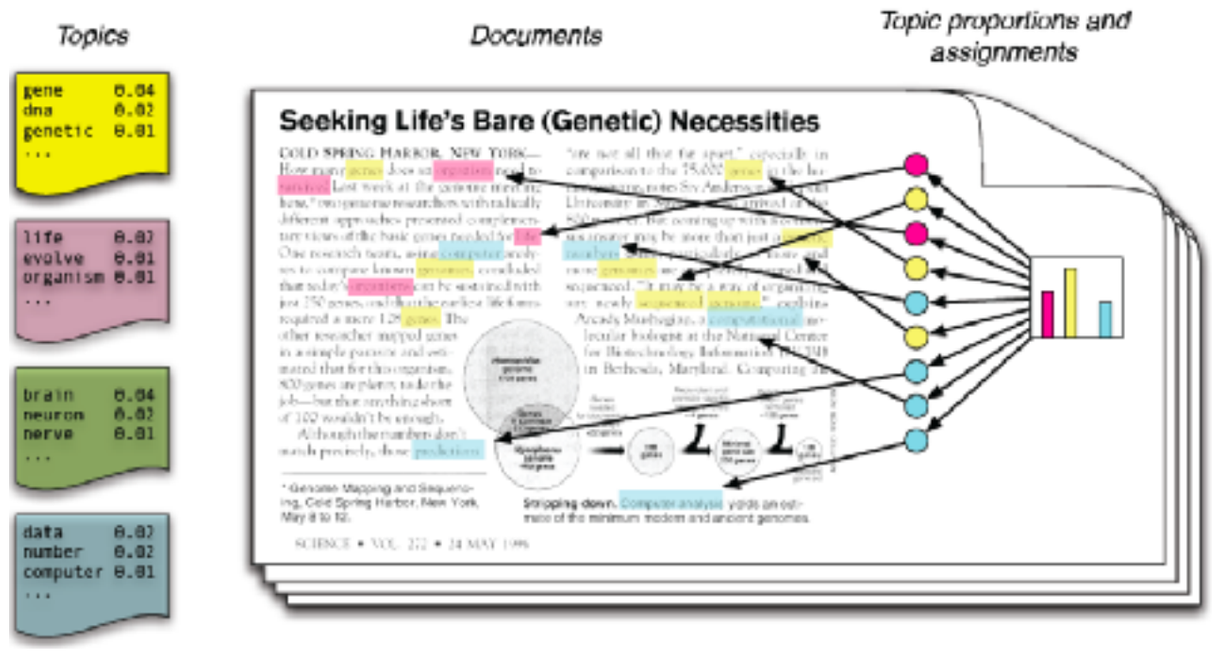
**TF-IDF**

Term x within document y

$tf_{x,y}$ = frequency of x in y

$df_x$ = number of documents containing x

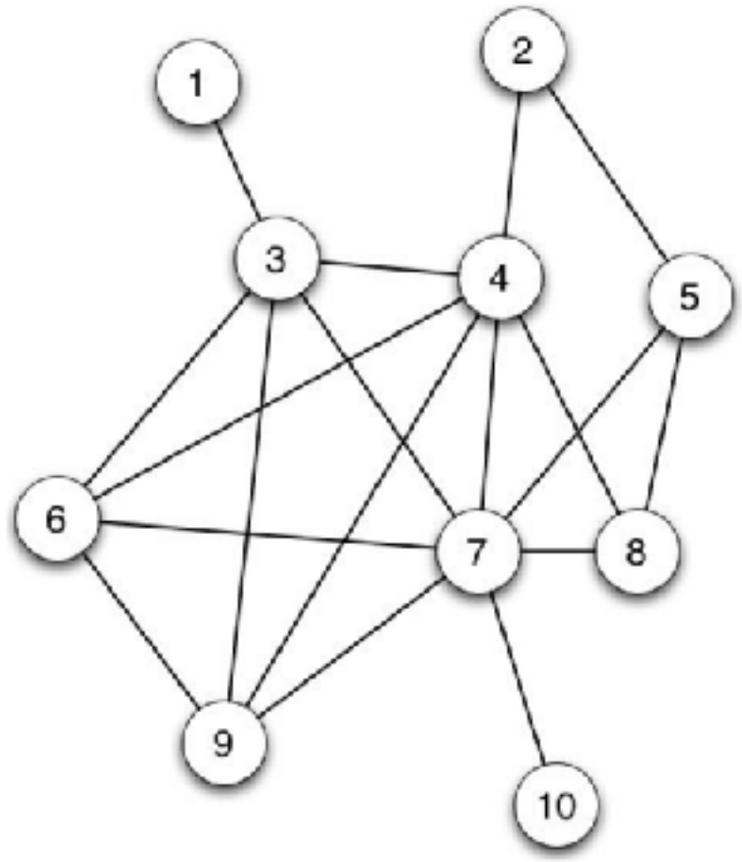
N = total number of documents




- Indicators for questions 1 and 2 starting with « Can », « Are », « Do », « Where » etc…
- Dictionaries on countries and cities to fuzzy match them (example : Paris 12, and Paris 8 –> Paris)

# THIRD APPROACH: GRAPHICAL MODELS

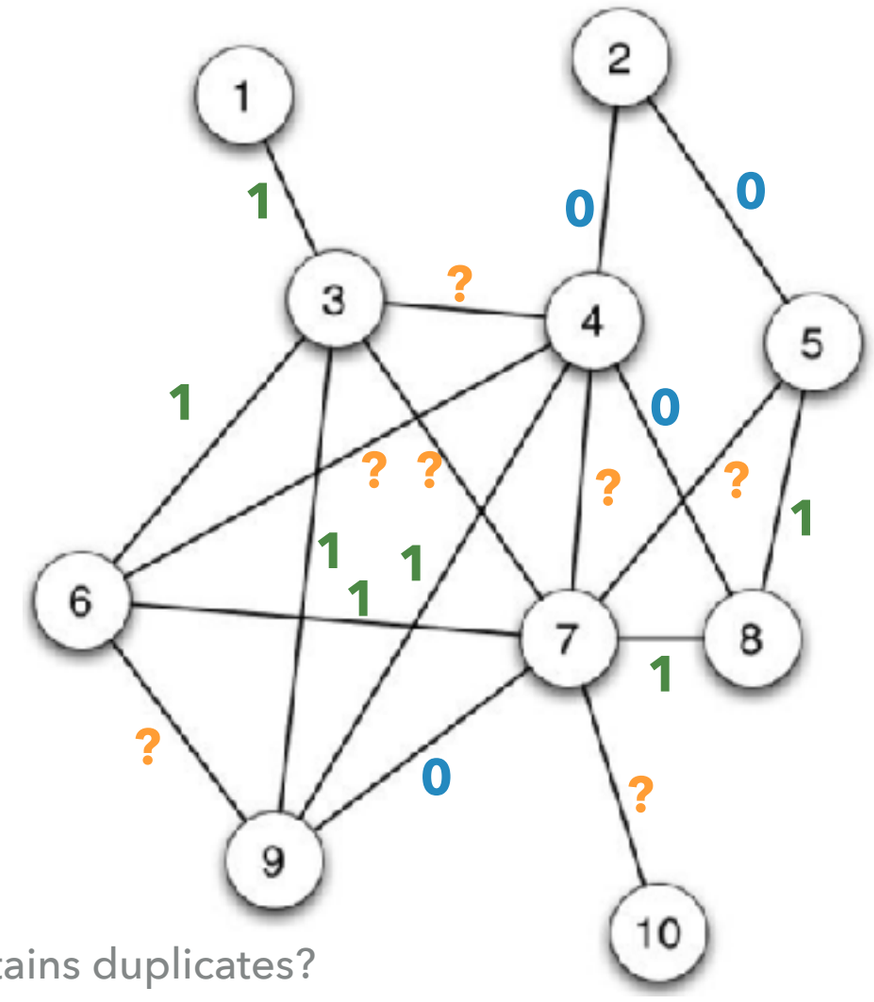
- We built the following graph:
  - Nodes: Questions
  - Edges: Question pairs
  - With train and test concatenated




- Why ?
  - Question pairs are pre-selected by a Quora's internal model
  - A lot of signal can be extracted from frequently asked questions

# THIRD APPROACH: GRAPHICAL MODELS

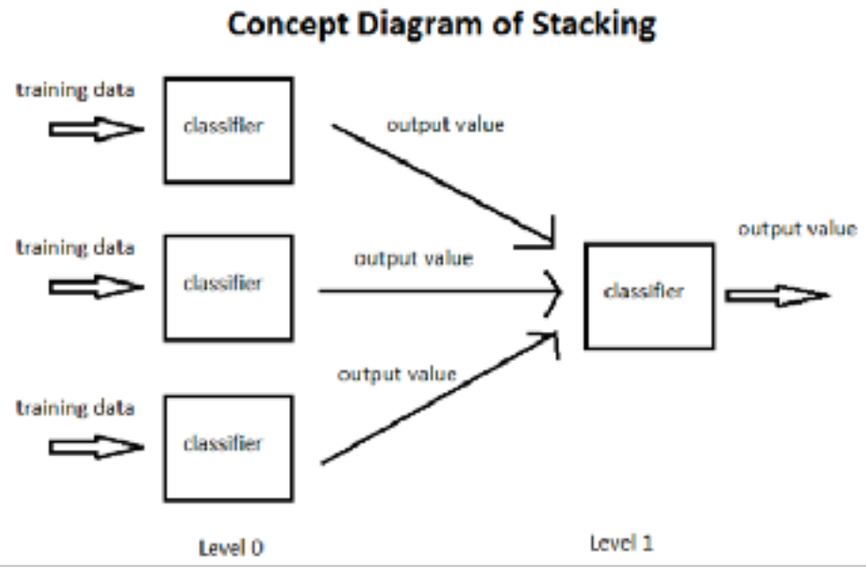
- For each pair of questions, we compute:
  - Min/Max/Intersection number of neighbors
  - Min/Max/Intersection of neighbors of order 2 (neighbors of neighbors), which aren't neighbors of order 1
  - Min/Max/Intersection of neighbors of order 3, which aren't neighbors of order 2 nor order 1
  - Shortest path from question 1 to question 2 when the edge is cut
- For each connex component in the graph, we compute:
  - Number of edges and nodes
  - % of pairs in train set
  - % of duplicated pairs in the component



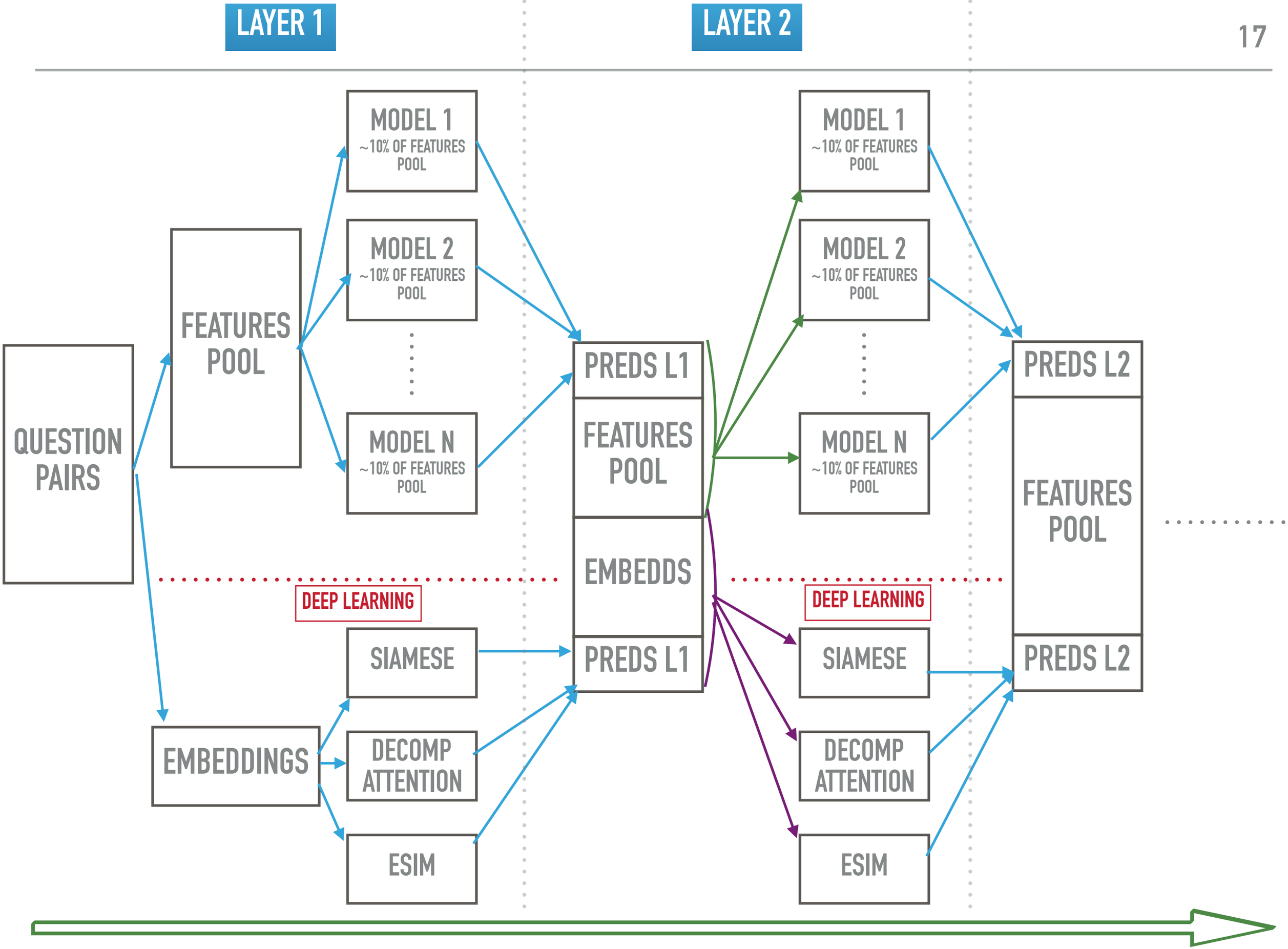

- Triangles and path features:
  - Triangle rule: 1/1 –> 1 and 1/0 –> 0
  - Indicator: Is there a path between the two question, which only contains duplicates?
- We re-computed above features on the weighted graph, weighted by our best model's predictions

# MODELIZATION: STACKING

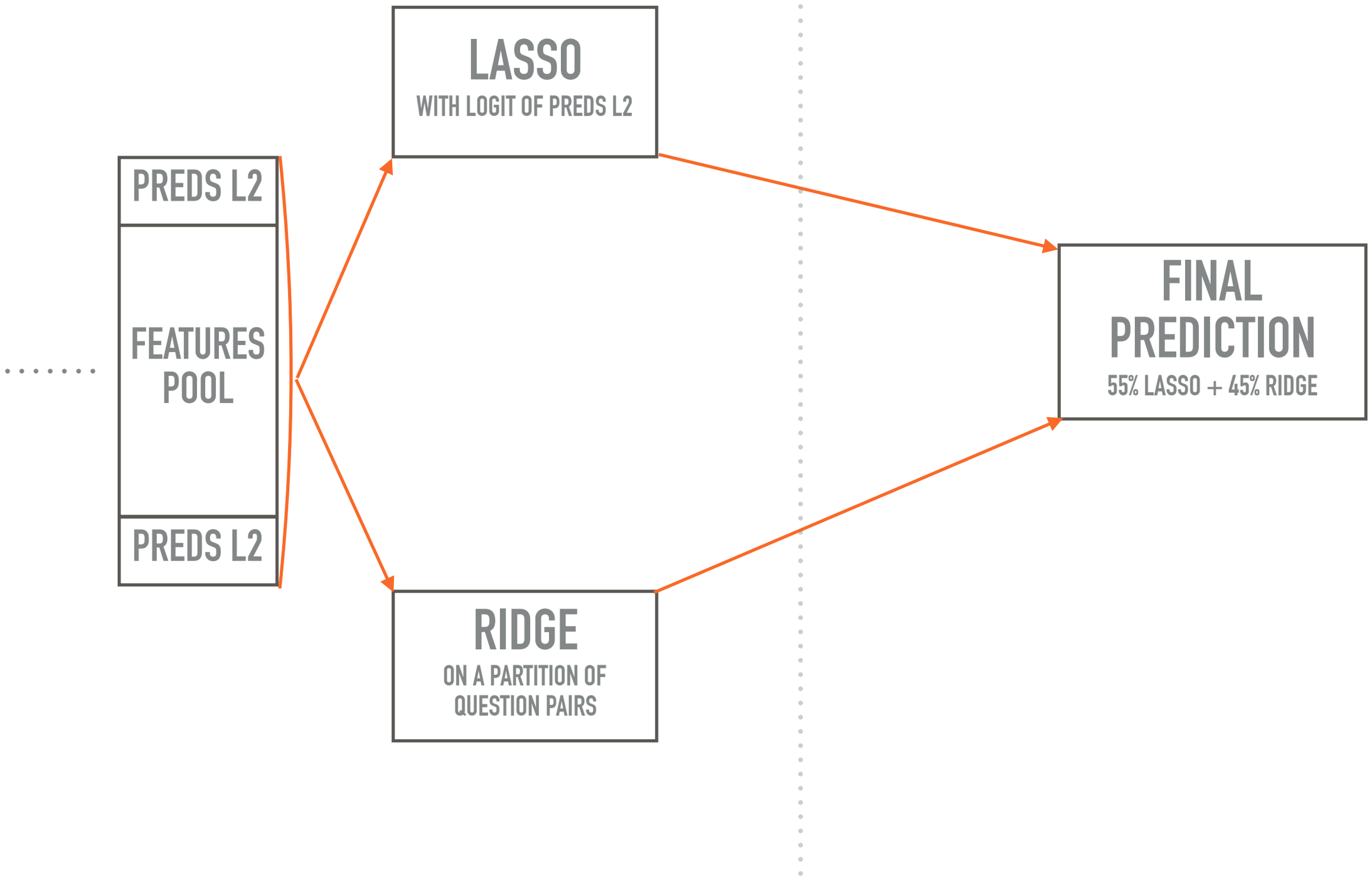
- ⚠ **WARNING:** This kind of modelization is very powerful, but requires to be made **properly**, there is a **HUGE** overfitting risk.




- What is stacking?
  - Models chaining
  - Predictions of models becomes input of next models
  - We make multiple layers, the first one takes our features as inputs, next layers take the same inputs + models' predictions
- Why stacking?
  - Some models are better than others on different parts of the data
  - Higher order layers' models will select the best models according to the dataset's properties


LAYER 1
LAYER 2
QUESTION PAIRS
FEATURES POOL
MODEL 1
~10% OF FEATURES POOL
MODEL 2
~10% OF FEATURES POOL
MODEL N
~10% OF FEATURES POOL
PREDS L1
FEATURES POOL
EMBEDDS
PREDS L1
MODEL 1
~10% OF FEATURES POOL
MODEL 2
~10% OF FEATURES POOL
MODEL N
~10% OF FEATURES POOL
PREDS L2
FEATURES POOL
PREDS L2
DEEP LEARNING
EMBEDDINGS
SIAMESE
DECOMP ATTENTION
ESIM
DEEP LEARNING
SIAMESE
DECOMP ATTENTION
ESIM

⚠ WARNING: HUGE OVERFITTING RISK HERE



# OUR STACKING IN A NUTSHELL

- 4 layers stacking
- Layer 1, ~300 models including:
  - Our two main Deep Learning architectures
  - A lot of classical algorithms, such as XGBoost, LightGBM, ExtraTrees, RandomForests, KNN, Logistic Regression, Naive Bayes, Stochastic Gradient Descent etc.
- Layer 2, ~150 models, including the same algorithms used in layer 1, trained with our base features, and Layer 1 predictions
- Layer 3, 2 models:
  - Lasso, with logit preprocessing on Layer 2 predictions
  - 3 Ridges, on a partition of the data in 3 chunks, trained each with the 3 Spearman's least correlated Layer 2 predictions
- Layer 4, blend of our layer 3 models, with coefficients 55/45 respectively, based on our CV score

# CONCLUSION

- We have around 450 models to generate the final submission
- At least 1 week to run all our models on huge hardware (10 GPU machines with 32Go RAM + 80 CPU machines with 120Go RAM).
- Our approaches' diversity deeply helped our stacking to optimize the LogLoss.
- This model cannot be used directly by Quora since it is way too complex –> Kaggle competitions are quite disconnected from a production environment.
- What's interesting for Quora is the way we analyzed their data, giving them a lot of insight for their own projects.

# THANK YOU!

Question pairs time!