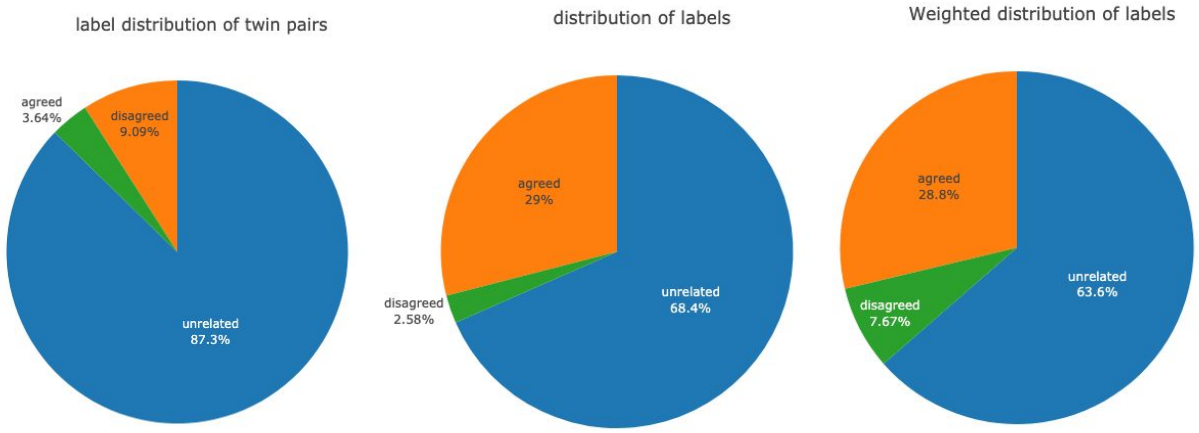
Fake News Detection

R07942057 黃于瑄 R07944013 林建甫 R07944059 黃國祐

Data Summarization:

Twin Pairs:

Twins: examples that tid1 = tid2, there are 110 twin pairs.



Word Analysis:

data missing:

- Id=42543 is an empty string, the relation with it should be agreed for all string.

Key word:

set 分析: 篩選

- unrelated: ('谣', 0.35),('言', 0.20),('辟', 0.19),('大', 0.19),('被', 0.15),('你', 0.14),('年', 0.13),('个', 0.13),('能', 0.13),('要', 0.13),('子', 0.13),('网', 0.13),
- agreed: ('你', 0.14),('大', 0.12),('个', 0.11),('要', 0.10),('能', 0.10),('来', 0.10),('用', 0.09),('在', 0.09),('都', 0.09),('上', 0.08),('可', 0.08),
- disagreed: ('谣', 0.96),('辟', 0.56),('言', 0.44),('网', 0.22),('传', 0.20),('被', 0.16),('方', 0.13),('友', 0.12),('在', 0.11),('发', 0.11),('真', 0.11),('警', 0.10),('出', 0.10),

Data Preprocessing:

Sentence to tokens — Jieba tokenizer

- Original sentence:
  - '2017养老保险又新增两项，农村老人人人可申领，你领到了吗
- Tokenized sentence:
  - [2017 | 养老保险 | 又 | 新增 | 两项 | ， | 农村 | 老人 | 人人 | 可 | 申领 | ， | 你 | 领到 | 了 | 吗]

Sentence to tokens — unigram

- Original sentence:
  - GDP首超香港？深圳澄清：还差一点点……
- Tokenized sentence:
  - [GDP | 首 | 超 | 香 | 港 | ？ | 深 | 圳 | 澄 | 清 | ： | 还 | 差 | 一 | 点 | 点 | … | … |]

Method:

Vector Space Model (TF-IDF):

After tokenizing all the sentences, we have:

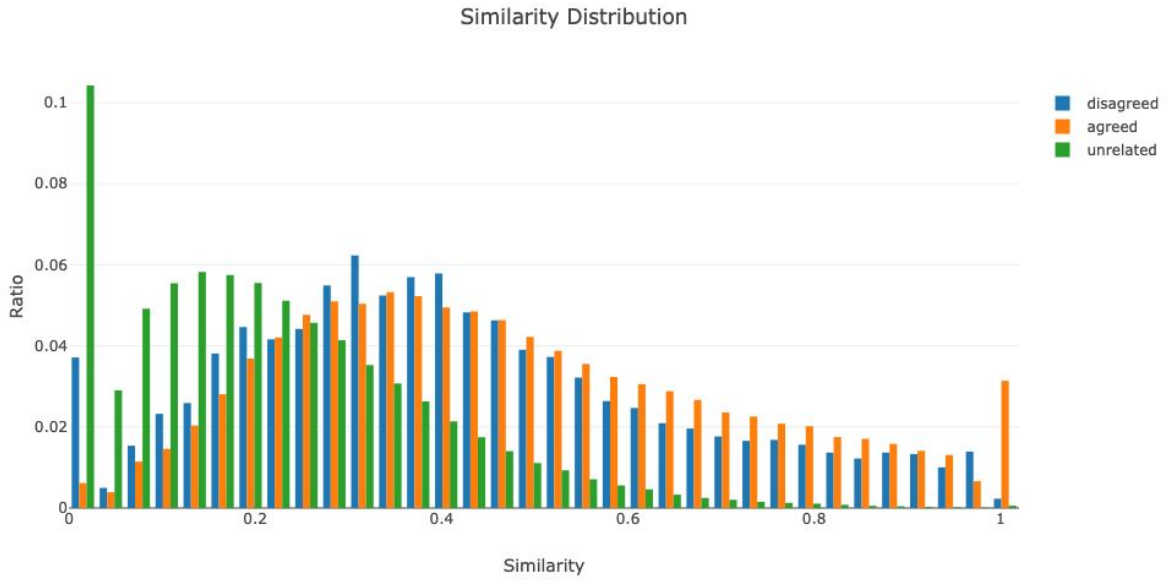
- 81,104 unique words

That is, for each sentence, we turn it into a vector with 81,104 dimensions. And we fill each dimension with its corresponding tf-idf, where

tf is defined as: **idf is defined as:** **similarity is defined as:**

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad \mathrm{idf}(t) = \log \frac{1+n}{1+\mathrm{df}(t)} + 1 \qquad similarity = cos(\theta) = \frac{A*B}{|A||B|}$$

Where A and B denotes the sentence vectors.
Similarity Distribution on training set



Scheme

id	tid1	tid2	title2_zh	label
17	21	22	「网警辟谣」飞机起飞前男人机舱口跪下？这故事居然是编的！	disagreed
70	83	86	辟谣：10秒钟工厂爆炸视频 实拍并不是常宁市水口山	disagreed
195	229	230	10000多个专偷孩子摘器官的外地人"谣言再次来袭	disagreed
198	229	236	网传我市静海区出现外地人偷小孩摘器官为谣言	disagreed
196	229	239	"乌市刑侦大队提醒：有外地人偷小孩、挖器官"的消息一出，吓坏了宝爸宝妈！	disagreed
199	229	242	辟谣｜10000多外地人解剖小孩偷器官？假的！	disagreed

```
key_words = ["辟谣", "网警", "谣言", "勿信", "传谣", "假的"]
```

Label examples having
(i) similarity >= 0.25 and at least one keyword as Disagreed,
(ii) similarity >= 0.25 and without any keyword as Agreed,
(iii) the rest as Unrelated.

Neural Network:

In this part, we try several structure and setting on the neural network model.

We first describe the general structure of our model.
The model is composed of two module, first is encoder and the second is predict layer. For each module, we test several testing.

Predict Layer:

- **Three class:**
  - loss function: cross entropy
- **hierarchical output:**
  If we consider the similarity between the output class. We could find out that agreed and disagreed share some probability. Therefore, we could build up a hierarchical structure to get more data and make the model more robust.
  - Two class: loss function: cross entropy
  - Two regression: loss function: binary cross entropy
  - Focal Two class: loss function: focal loss

Encoder Layer:

General setting:

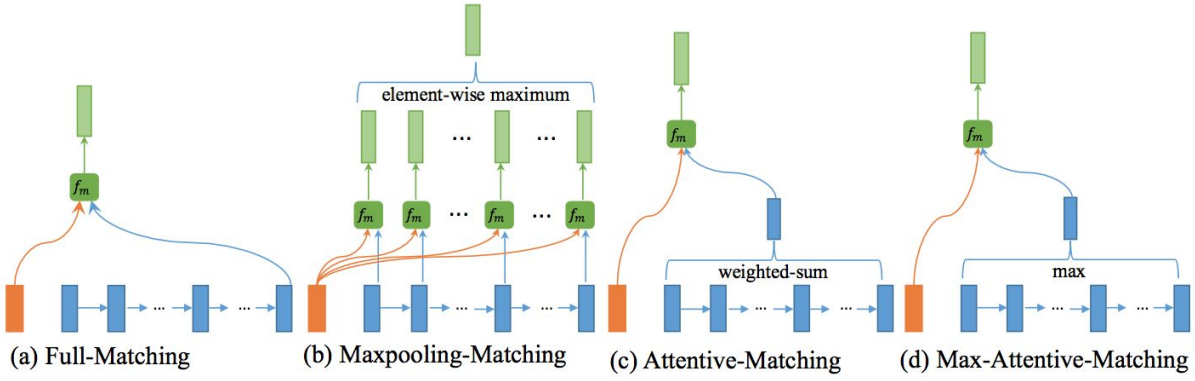
After the experiment, we find out that GRU works better than LSTM, therefore we choose GRU to do the following experiment.

For the attention mechanism, we use the Luong setting. $Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{h}})V$,

where K==V, h is the hidden dimension.

Difference:

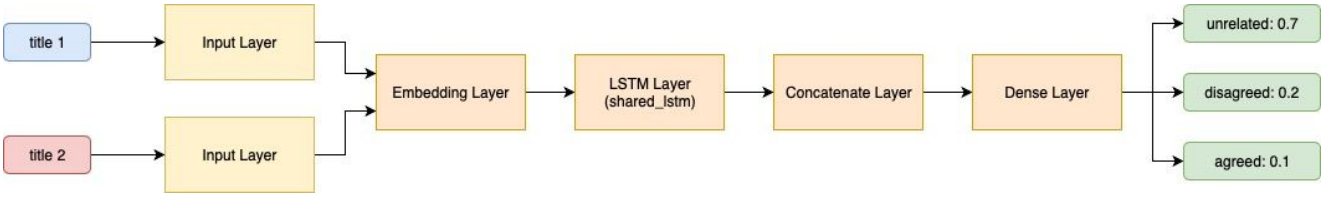
- **Siamese RNN:**
  - Using a single GRU(Gated recurrent unit) to encode the sentence and do the mean and max pooling on the output and concat it as the sentence representation.
- **RNN-Attention:**
  - Using a single GRU to encode the sentence and then pass it to the attention layer.
    Q: Average on the output from the sentence1 encoder,
    K,V: Output of the sentence2 encoder.
  - We concat the Q and the attention output as sentence representation.
- **Attention-RNN:**
  - Different from the last model, we first do the bidirectional attention. For each sentence, we treat itself as Q and the other one as K,V.
  - With the RNN, we aggregate the both sentence's information and we do mean and max pooling on the output and concat it as the representation.
- **Bimpm:**
  - Using a single GRU(Gated recurrent unit) to encode the sentence. We then use four kinds of attention mechanism to extract the relation.



  - We concat these output, and pass it to the second RNN layer to aggregate the information. We use the last timestep's output as the representation of the sentence.

Siamese RNN + different feature combination:

Model: siamese RNN (LSTM) network



Based on this model (Siamese RNN), different feature engineering & combination will be tried.

Feature Engineering

Data Augmentation

- Exchange the news title pair order form (title1, title2) into (title2, title1) to enrich the amount of data

Pre-trained Word Embeddings

- word2vec
  - Word-level
    - Use Jieba tokenizer to obtained tokernized titles, and collect tokens to form training text data
    - Train a model to learn vector representation (250d) of words using training text data
    - Calculate the mean vector of word vectors in a sentence to obtain the vector representation of each title.
  - Character-level
    - Collect characters from title pairs to form training text data
    - Train a model to learn vector representation (250d) of characters using training text data
    - Calculate the mean vector of character vectors in a sentence to obtain the vector representation of each title.
- doc2vec
  - Use Jieba tokenizer to obtained tokernized titles, and collect tokens to form training text data
  - Train a model to learn vector representation (250d) of words using training text data
  - Obtain each title's vectro representation directly.
- fastText
  - Word-level
    - Use Jieba tokenizer to obtained tokernized titles, and collect tokens to form training text data
    - Train a model to learn vector representation (250d) of words using training text data
    - Calculate the mean vector of word vectors in a sentence to obtain the vector representation of each title.
  - Character-level
    - Collect characters from title pairs to form training text data
    - Train a model to learn vector representation (250d) of characters using training text data
    - Calculate the mean vector of character vectors in a sentence to obtain the vector representation of each title.
- bert-as-service
  - Documnet link: https://bert-as-service.readthedocs.io/en/latest/
  - A sentence encoding service for mapping a variable-length sentence to a fixed-length vector.
  - Pre-trained BERT Model: BERT-Base, Chinese
    - Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters
  - Each title will be mapped to a 768d vector.

Handcrafted features

- Concatenate following handcrafted features with the output of Concatenate Layer of the original model:
  - Cosine similarity of title 1 and title 2
  - Statistics features of rumor keywords:
    - ```
      key_words = ["辟谣", "网警", "谣言", "勿信", "传谣", "假的"]
      ```
  - Overlap ratio of string matching between title 1 and title 2
  - Token set ratio matching

Performance

TFIDF:

Models	Private	Public
TF-IDF	0.75865	0.75122

Siamese RNN + different feature combination:

Model: siamese RNN (LSTM) network

Original	w/ data ugmentation
0.70377	0.71589

Pre-trained Word Embeddings

word2vec (word-level)	word2vec (char-level)	doc2vec	fastText (word-level)	fastText (char-level)	bert-as-service encoding
0.65316	0.65598	0.69042	0.71019	0.72766	0.80620

Handcrafted features

TF-IDF similarity	Statistics of keywords	Overlap ratio of string matching	Token set ratio matching	All the handcrafted features
0.70631	0.70582	0.72044	0.71305	0.73851

Neural Network:

	regression	Two class	Focal loss	Three class	regression	Two class	Focal loss	Three class
	unigram				jieba			
Siamese RNN	0.75590	0.74975	0.74407	0.76467	0.77376	0.76019	0.75334	0.77637
LSTM-Attn.	0.72597	0.71681	0.78262	0.71722	0.72061	0.72734	0.75812	0.73698
Attn-LSTM	0.82278	0.82264	0.82586	0.82499	0.80763	0.80943	0.81282	0.81378
Bimpm	0.82220	0.81812	0.80650	0.82628	0.81691	0.82964	0.82824	0.82726
Bert	0.86526(unigram three class)							

Discussion

Siamese RNN + different feature combination

- Data Augmentation
      ```

    - After performing data augmentation, the weighted categorization accuracy has been improved slightly. It seems like enriching training data has effect, but the data used for training should be processed carefully before training, like: elimination of confusing news pairs, elimination of strangely labelized title pairs.
- Pre-trained Word Embeddings
    - It seems like pre-trained word embeddings cannot always perform better than the original model's. (word2vec, doc2vec).
    - When it comes to word2vec, it can be observed from the above table that the model using char-level word2vec model performs better than the one using word-level word2vec. Similar phenomenon can also be observed in fastText. The reason may be that since there are quite lots of specific words whose occurrence is very low, these words cannot provide useful or significant information to help model learn in this task.
    - The performance of the model using bert encoding is the best among these pre-trained word embedding, and it can be seen that the weighted categorization accuracy has been significantly improved.
- Handcrafted features
    - From the observation of these experiments, it can be deduced that overlap ratio of string matching between title1 and title2 has more importance among these handcrafted features.
    - The introduction of these additional handcrafted features can actually improve model's performance.

Neural Network:

Network structure:

- We could find out, attention could help the model work better. Beside the attention mechanism also need an aggregation method to encode the output.
- Focal loss does help in some model, maybe we need to turn the parameter more to make it works better.

Division of works

黃國祐	黃于瑄	林建甫
Data Preprocessing ● Word Analysis Model ● Neural Network Discussion ● Neural Network	Model ● Siamese RNN + different feature combination Discussion ● Analysis of different feature combination	Data Summarization ● Data distribution ● title(tid) distribution ● Twin Pairs Data Preprocessing ● Tokenizing Model ● TF-IDF