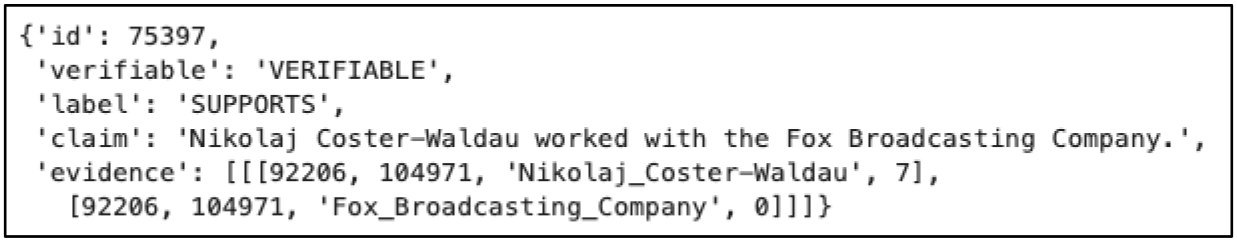
# 3-Stage Deep Semantic Network for Evidence Retrieval and Claim Verification

## 1. Dataset

The dataset being used for our model is the **Fact Extraction and VERification (FEVER)** dataset. The dataset consists of a total of 145449 claims, where each is labelled as either SUPPORTS, REFUTES or NOT ENOUGH INFO, depending on whether relevant evidence from the corpus can support/refute it.



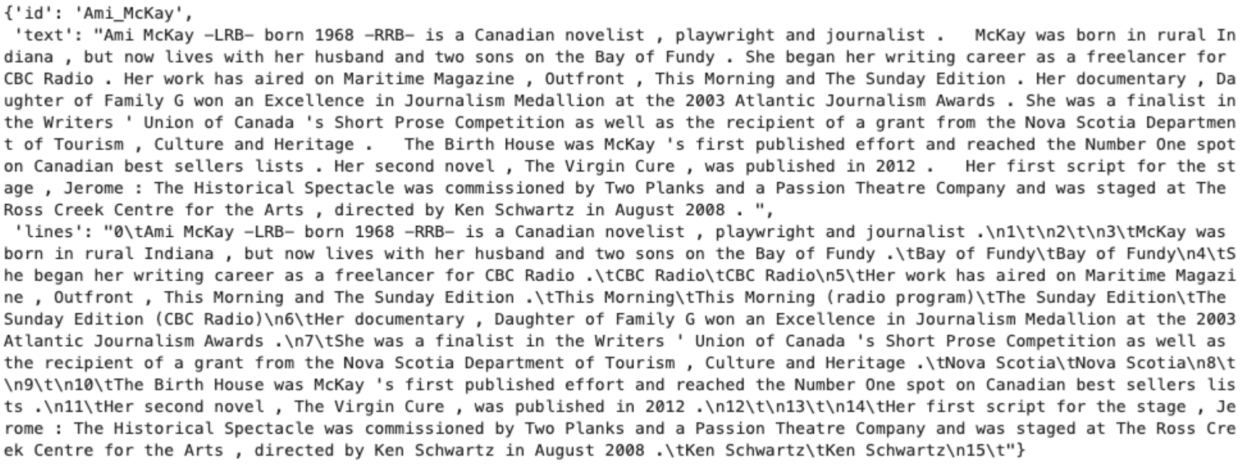
```
{'id': 75397,
 'verifiable': 'VERIFIABLE',
 'label': 'SUPPORTS',
 'claim': 'Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.',
 'evidence': [[[92206, 104971, 'Nikolaj_Coster-Waldau', 7],
   [92206, 104971, 'Fox_Broadcasting_Company', 0]]]}
```

Figure 1 - An instance of training data

As per Figure 1, each training instance contains the following information -

- **Id** - A unique identifier for that instance
- **Verifiable** - If the claim can be verified as SUPPORTS or REFUTES, it is "VERIFIABLE" otherwise "NOT VERIFIABLE"
- **Label** - Claim label, either "SUPPORTS, REFUTES or NOT ENOUGH INFO"
- **Claim** - The claim sentence
- **Evidence** - The list of evidence sets. Each set contains a list of sentences which when combined validates the claim as either SUPPORTS or REFUTES. If the claim is labelled NOT ENOUGH INFO, the evidence is not provided and thus empty.

The FEVER datasets' evidence has been formed through the 2017 WIKIPEDIA dump which is a humongous file of 7GigaBytes containing approximately 550,000 Wikipedia articles.



```
{'id': 'Ami_McKay',
 'text': "Ami McKay -LRB- born 1968 -RRB- is a Canadian novelist , playwright and journalist .   McKay was born in rural In
diana , but now lives with her husband and two sons on the Bay of Fundy . She began her writing career as a freelancer for
CBC Radio . Her work has aired on Maritime Magazine , Outfront , This Morning and The Sunday Edition . Her documentary , Da
ughter of Family G won an Excellence in Journalism Medallion at the 2003 Atlantic Journalism Awards . She was a finalist in
the Writers ' Union of Canada 's Short Prose Competition as well as the recipient of a grant from the Nova Scotia Departmen
t of Tourism , Culture and Heritage .   The Birth House was McKay 's first published effort and reached the Number One spot
on Canadian best sellers lists . Her second novel , The Virgin Cure , was published in 2012 .   Her first script for the st
age , Jerome : The Historical Spectacle was commissioned by Two Planks and a Passion Theatre Company and was staged at The
Ross Creek Centre for the Arts , directed by Ken Schwartz in August 2008 . ",
 'lines': "0\tAmi McKay -LRB- born 1968 -RRB- is a Canadian novelist , playwright and journalist .\n1\t\n2\t\n3\tMcKay was
born in rural Indiana , but now lives with her husband and two sons on the Bay of Fundy .\tBay of Fundy\tBay of Fundy\n4\tS
he began her writing career as a freelancer for CBC Radio .\tCBC Radio\tCBC Radio\n5\tHer work has aired on Maritime Magazi
ne , Outfront , This Morning and The Sunday Edition .\tThis Morning\tThis Morning (radio program)\tThe Sunday Edition\tThe
Sunday Edition (CBC Radio)\n6\tHer documentary , Daughter of Family G won an Excellence in Journalism Medallion at the 2003
Atlantic Journalism Awards .\n7\tShe was a finalist in the Writers ' Union of Canada 's Short Prose Competition as well as
the recipient of a grant from the Nova Scotia Department of Tourism , Culture and Heritage .\tNova Scotia\tNova Scotia\n8\t
\n9\t\n10\tThe Birth House was McKay 's first published effort and reached the Number One spot on Canadian best sellers lis
ts .\n11\tHer second novel , The Virgin Cure , was published in 2012 .\n12\t\n13\t\n14\tHer first script for the stage , Je
rome : The Historical Spectacle was commissioned by Two Planks and a Passion Theatre Company and was staged at The Ross Cre
ek Centre for the Arts , directed by Ken Schwartz in August 2008 .\tKen Schwartz\tKen Schwartz\n15\t"}
```

Figure 2 - An instance of the Wikipedia dump

As per Figure 2, each training instance contains the following information -

1. **Id** - Unprocessed Wikipedia article name.
2. **Text** - Wikipedia introductory section for the article
3. **Lines** - Wikipedia introductory section with lines mentioned.

## 2 Document Retrieval

**Objective** - Document retrieval consists of parsing the claim to find relevant keywords and selecting the Wikipedia articles which might contain the evidence to validate the claim.

One of the initial approaches that we thought of was to find similar documents using keywords through Cosine Similarity. However, the huge amount of data makes it impractical to go ahead with this approach as for each claim we would have to go through every Wikipedia article (550,000).

The next approach we thought of was to create a MySQL database of Wikipedia dumps which we would query against to find relevant Wikipedia articles. For the baseline, we decided to go ahead with Apache Solr because we have experience working with it and the query process would be simpler.

Therefore, for document retrieval, we index all the Wikipedia articles on Solr and query it to find relevant articles. Now, we need to decide on how to formulate the query and how many documents need to be retrieved.

The brute force approach would be to query the whole claim against the index and select the top K documents. It's clear that this would result in poor precision and recall which we found. Therefore, we formulated our query.

For the baseline system, we fire an initial query based on noun chunks, i.e. we utilize spacy's powerful parser to find noun chunks in our claim and concatenate all of them to create a single query. If no results are fetched we then send the whole claim as a query. From the retrieved documents, we select the top 5 documents.

Precision and Recall formulas for the document retrieval process is defined as follows:

- **Precision** = Number of correctly predicted documents/ Total number of predicted documents
- **Recall** = Number of matched documents/ Total number of documents in evidence set

| | **Precision (%)** | **Recall (%)** | **F1-Score** |
|---|---|---|---|
| **Document Retrieval** | 19.22 | 85.88 | 31.05 |
| **FEVER BASELINE** | 11.28 | 47.87 | 18.26 |

Our baseline document retrieval system works significantly better than the FEVER baseline system. Moreover, it is customized for our future system and hence a much better evaluation metric to compare against. There is lots of scope for improvement, such as -

1) Improve SOLR Query Parameters
2) Number of documents to be retrieved can be considered as a hyper-parameter which can be tweaked for optimum results
3) Proper retrieval of "disambiguitive" Wikipedia article titles.
4) Experiment with creating a Neural Network model for "disambiguitive" document selection

## 2 Sentence Selection

**Objective -** Given a claim, find all sentences relevant to the validation of the claim from the retrieved documents.

Once again, the most basic approach would be to find similar sentences through tf-idf ranking and then prune the selected sentences to the top K sentences. This approach was utilized by the FEVER baseline system.

However, we wanted to build a baseline through which we could more efficiently evaluate our full system. We already have the FEVER baseline scores, a new baseline modelled on our strategy would benefit us in the long run.

Therefore, we went ahead with the strategy discussed during Phase 1 of the project - Building a deep learning model to select sentences. There are two approaches here -

1. Given a claim and list of sentences, output the selected sentences.
2. Given a claim and sentence, output whether to select the sentence.

To keep the baseline system simple enough such that we could build on it in the future, Approach 2 was selected. Hence, the updated objective now is - "Given a claim and sentence, classify whether the sentence is relevant to the claim or not."

Now, we need to clean and prepare the training data for sentence selection according to our objective - In the FEVER training data provided, evidence sentences are provided for each **verifiable** claim. These are our truth labels as we want our model to learn that given a claim these particular sentences are what we want.

Now comes the difficulty of selecting false labels. For a claim, which sentence should we select as false? Logically, any sentence outside of its evidence set is a false sentence. But then how many should we select? Would they even have any contextual resemblance with the claim? All these questions need to be answered.

We tackled this problem using the following approach - For a particular claim, all the sentences from the retrieved documents in STEP 1 Document retrieval are the candidate sentences for false labels. These sentences have the highest probability of having contextual resemblance with the claim. Moreover, since the truth and false labels in the training data are from the same pool, the model trained would be a much better and robust classifier so as to segregate the truth and false labels.

Another question still remains and that is how many false sentences to select? We still can't select all the sentences from the candidate pool as false sentences because the ratio of truth and false sentences would explode and our training would become imbalanced. Hence, annealed sampling strategy comes into picture.

Annealed sampling strategy is that after every epoch, the ratio of truth to false labels would increase by 10%. Initially, it is set to around 50%. The false sentences are selected randomly.

The deep learning model - Defined as simple as possible for the baseline system, it consists of a single unidirectional lstm layer. The claim is tokenized and sent through an embedding layer initialized with CharNgram word vectors, then through the lstm layer. The same flow is followed with the sentence. Point to be noted is that the lstm layer is shared between both the claim and sentence; it was done so that the lstm layer learns the language model for the task at hand. The outputs of the lstm layer are concatenated and passed through a linear layer to output a single sigmoidal value. This value is threshold to predict the correct label.

This concludes our entire sentence selection pipeline.

| | **Precision (%)** | **Recall (%)** | **F1-Score** |
|---|---|---|---|
| **Sentence Selection** | 52.85 | 53.44 | 53.34<br>(Positive Class F1 - 21.0) |

Our baseline sentence selection system performs reasonably well. The overall F1 score is 53.34 whereas the F1 (true) is 21 after training for only 3 epochs. Given that our model was just a simple single layer unidirectional lstm layer, it sorts of validates our approach to go with a deep learning method. There are lots of experiments and improvements which can and are to be performed -

1) Use attention mechanism
2) Increase the dimension of Glove Vector encoding from 100 to 300
3) Use ELMO encodings
4) Improve data cleaning and tokenization modules
5) Use Document TF-IDF score as a feature for Model training

## 2 Claim Verification

We consider the sentence selection task and claim verification task to be very similar at large. Hence, all the questions raised in sentence selection are raised over here too.

Similarly to sentence selection, we go ahead with a deep learning model for the task of claim verification though for different reasons as tf-idf ranking for verification makes no sense. The objective for claim verification now is - "Given a claim and a sentence, classify whether the sentence supports, refutes or provides not enough info based on the claim"

Again, cleaning and formulating for training data needs to take place. There are two cases here -

1. The claim is verifiable, i.e. it's label is SUPPORT or REFUTE - In this case, the FEVER training data provided is relevant without any further processing for the training of the model. That is, if a claim's label is SUPPORT, then for the claim we generate tuples of (claim, sentence) where the sentence is from the evidence set and the label for this instance is the claims' label, SUPPORT.
2. The claim is not verifiable, i.e. it's label is NOT ENOUGH INFO - In this case, the FEVER training data provides no evidence but we need to have training data for the model to train on such that it learns that it's not enough info. Therefore, from the candidate pool of sentences from document retrieval we randomly sample 4-5 sentences for each claim and generate the training tuple.

The sentences that make up the evidence set are used to train the claim-verification model. Similar to sentence selection, each claim will be paired with each of the sentences in the evidence set and will be given as an input to a Single Layer Unidirectional LSTM model. In case the claim is not verifiable, a set of sentences are randomly picked from the candidate pool to form it's training sample.

The deep learning model - Defined as simple as possible for the baseline system, it consists of a single unidirectional lstm layer. The claim is tokenized and sent through an embedding layer initialized with CharNgram word vectors, then through the lstm layer. The same flow is followed with the sentence. Point to be noted is that the lstm layer is shared between both the claim and sentence; it was done so that the lstm layer learns the language model for the task at hand. The outputs of the lstm layer are concatenated and passed through a linear layer to output 3 labels - SUPPORT, REFUTE and NOT ENOUGH INFO which is then passed through a Softmax layer to get our prediction.

| | **Precision (%)** | **Recall (%)** | **F1-Score** |
|---|---|---|---|
| **Claim Verification** | 70.74 | 77.23 | 73.85 |

Our baseline claim verification system also performs reasonably good giving an overall F1 score of 73.85 after training for 3 epochs only. We aren't able to report results of the claim verification system against some test data, however, it worked more than good enough than the baseline system which reports a **label accuracy** of **48.84%**. Some of the future experiments and improvements -

1) Use attention mechanism
2) Use ELMO encodings
3) Use sentence selection score as a feature for Model training

## 3. Remarks

The above defined baseline system is a crude implementation of our final system. This helps us in evaluating our initial ideas and explore possible tweaks and improvements for the next phase. Hence, the final system will be built on top of the baseline architecture with the improvements mentioned above.