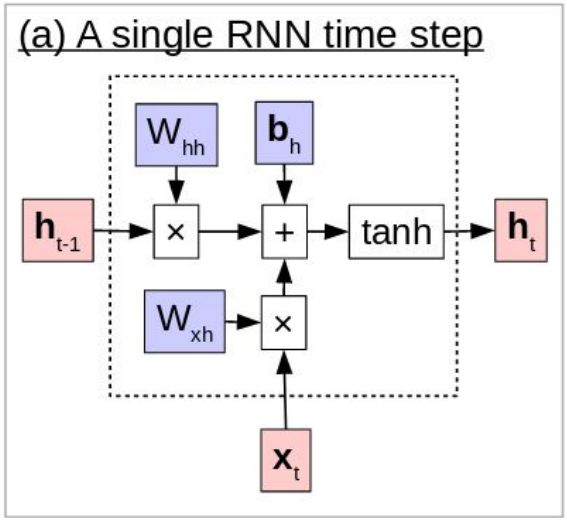
Neural Encoder-Decoder Architecture for Text Generation

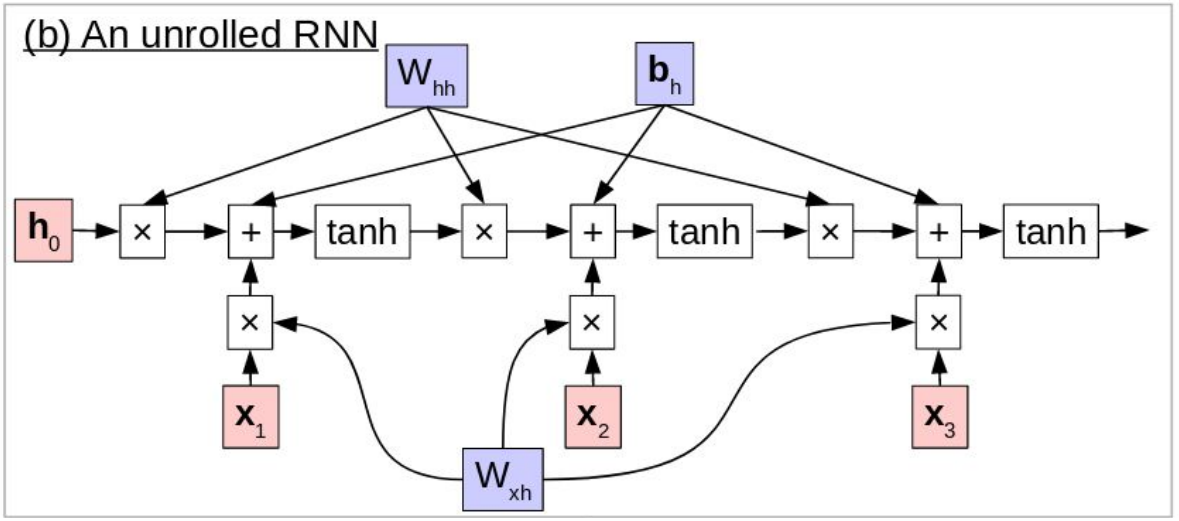
Reem, François, Badr

Outline

- Neural architectures for sequential data
- RDF to Text as seq2seq task
- Preprocessing pipeline
- Some ideas
- Plan

Recurrent Neural Networks

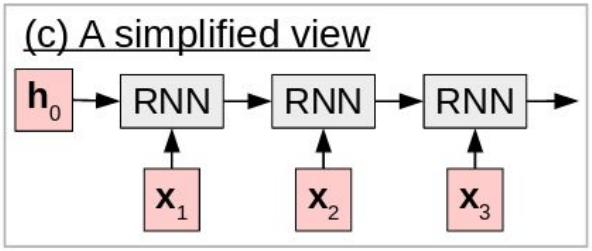




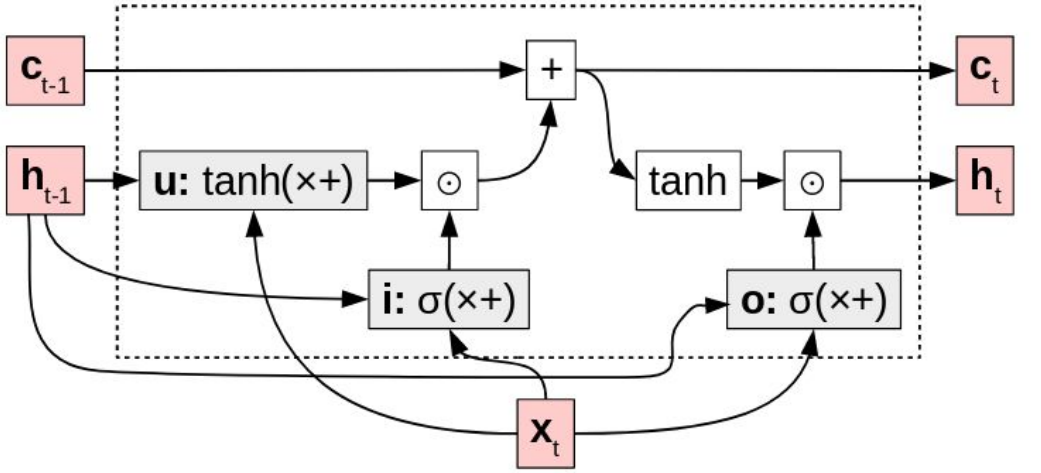
$$\boldsymbol{m}_t = M_{\cdot,e_{t-1}}$$

$$\boldsymbol{h}_t = \begin{cases} \tanh(W_{mh}\boldsymbol{m}_t + W_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h) & t \geq 1, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

$$\boldsymbol{p}_t = \text{softmax}(W_{hs}\boldsymbol{h}_t + b_s).$$



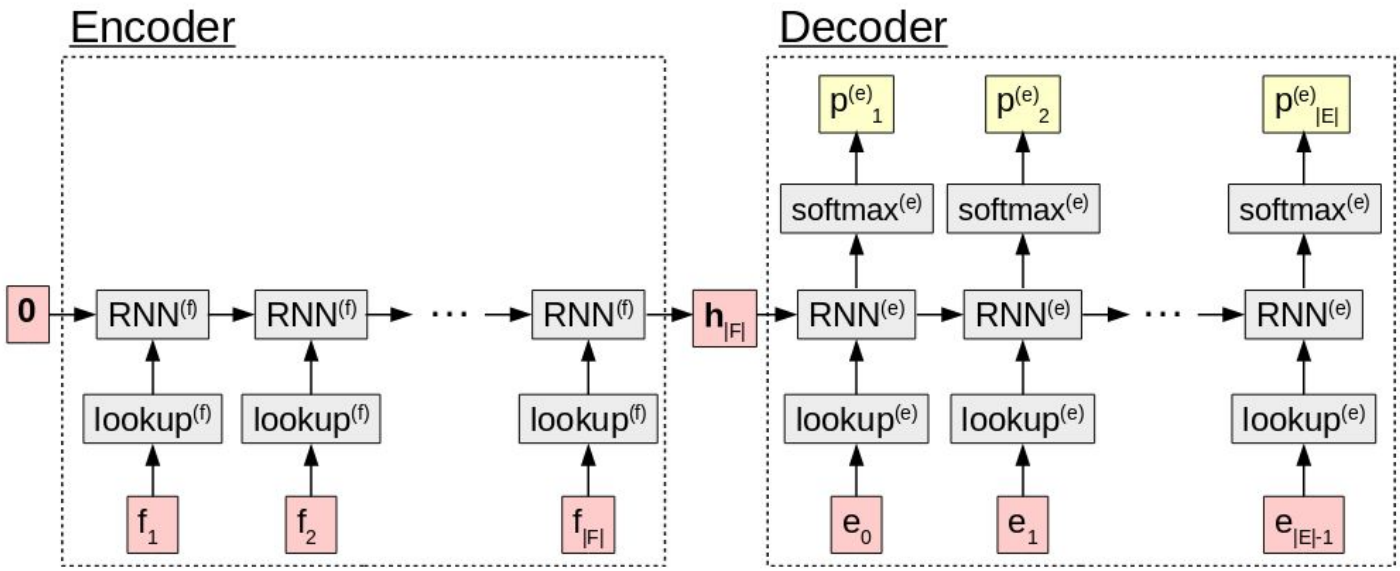
LSTM



update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

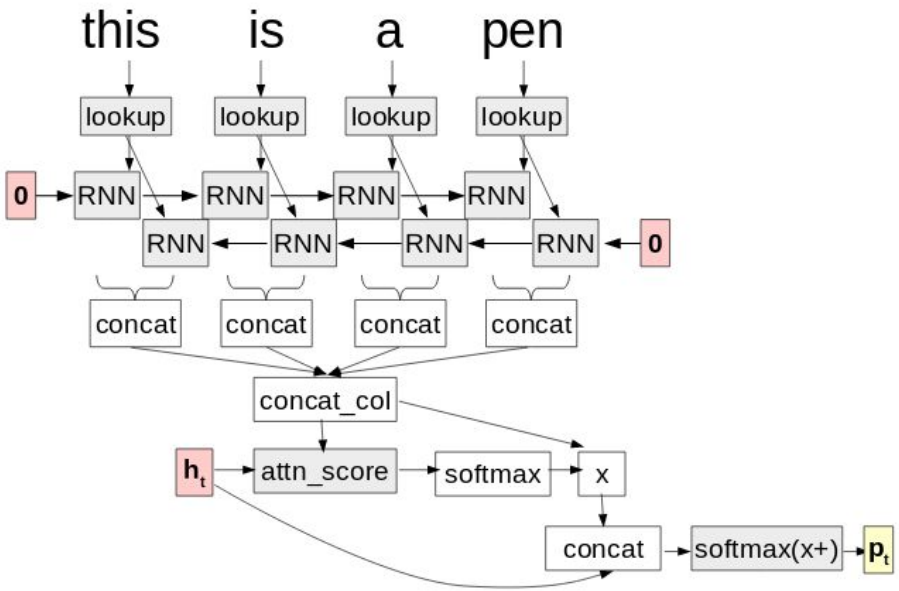
$$\boldsymbol{u}_t = \tanh(W_{xu}\boldsymbol{x}_t + W_{hu}h_{t-1} + \boldsymbol{b}_u)$$
$$\boldsymbol{i}_t = \sigma(W_{xi}\boldsymbol{x}_t + W_{hi}h_{t-1} + \boldsymbol{b}_i)$$
$$\boldsymbol{o}_t = \sigma(W_{xo}\boldsymbol{x}_t + W_{ho}h_{t-1} + \boldsymbol{b}_o)$$
$$\boldsymbol{c}_t = \boldsymbol{i}_t \odot \boldsymbol{u}_t + \boldsymbol{c}_{t-1}$$
$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t).$$

RNN Encoder-Decoder



$$
\begin{aligned}
\boldsymbol{m}_t^{(f)} &= M_{\cdot,f_t}^{(f)} \\
\boldsymbol{h}_t^{(f)} &= \begin{cases} \mathrm{RNN}^{(f)}(\boldsymbol{m}_t^{(f)}, \boldsymbol{h}_{t-1}^{(f)}) & t \geq 1, \\ \mathbf{0} & \text{otherwise.} \end{cases} \\
\boldsymbol{m}_t^{(e)} &= M_{\cdot,e_{t-1}}^{(e)} \\
\boldsymbol{h}_t^{(e)} &= \begin{cases} \mathrm{RNN}^{(e)}(\boldsymbol{m}_t^{(e)}, \boldsymbol{h}_{t-1}^{(e)}) & t \geq 1, \\ \boldsymbol{h}_{|F|}^{(f)} & \text{otherwise.} \end{cases} \\
\boldsymbol{p}_t^{(e)} &= \mathrm{softmax}(W_{hs}\boldsymbol{h}_t^{(e)} + b_s)
\end{aligned}
$$

Bidirectional Encoder - Attentional Decoder

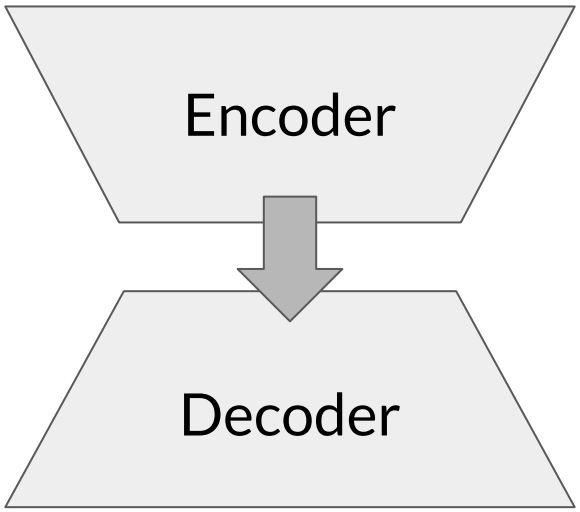


$$\boldsymbol{p}_t^{(e)} = \text{softmax}(W_{hs}[\boldsymbol{h}_t^{(e)}; \boldsymbol{c}_t] + b_s).$$

RDF to Text as seq2seq Task

(Philippines | leaderName | Rodrigo_Duterte) (Binignit | region | Philippines)

(Binignit | ingredient | Sago) (Binignit | ingredient | Banana)

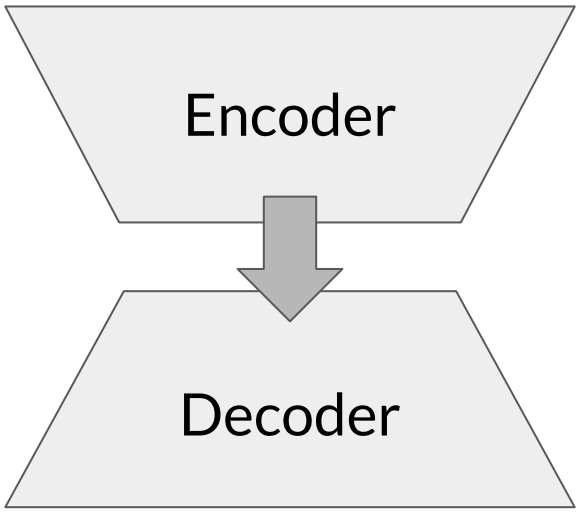


The main ingredient of binignit is sago, and it also contains banana. The dish comes from the Philippines which is led by president Rodrigo Duterte.

Entity to Text Matching

(Philippines | leaderName | Rodrigo_Duterte) (Binignit | region | Philippines)

(Binignit | ingredient | Sago) (Binignit | ingredient | Banana)

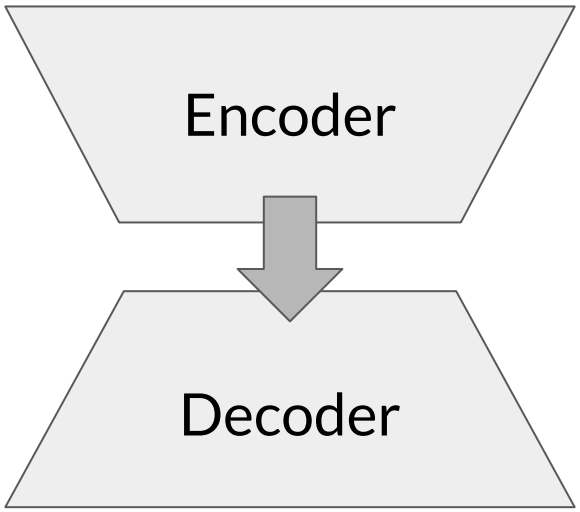


The main ingredient of binignit is sago, and it also contains banana. The dish comes from the Philippines which is led by president Rodrigo Duterte.

Entity to Text Matching (sometimes difficult)

(The United States of America | leaderName | Donald_Trump)

(The United States of America | capital | Washington_D.C.)



The current president of the US is Donald Trump who's working from the capital of the state Washington DC.

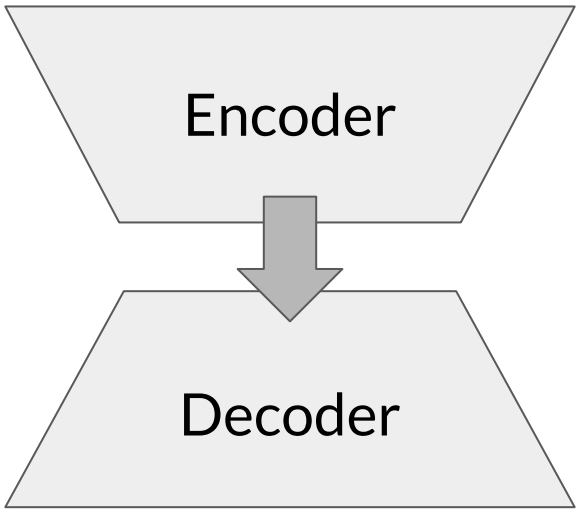
Improving Entity to Text Matching

- For now, we only use exact matching and lowercase search
- We are willing to use *named entity recognition* (NER) to improve this search
- Also, a substring matching can be a good idea for abbreviations
- A fancy heuristic-based text similarity procedure!
- Every entity in the RDF graph ***must*** be matched with a string in the target sequence

Delexicalisation

(COUNTRY | leaderName | PERSON) (FOOD | region | COUNTRY)

(FOOD | ingredient | FOOD) (FOOD | ingredient | FOOD)



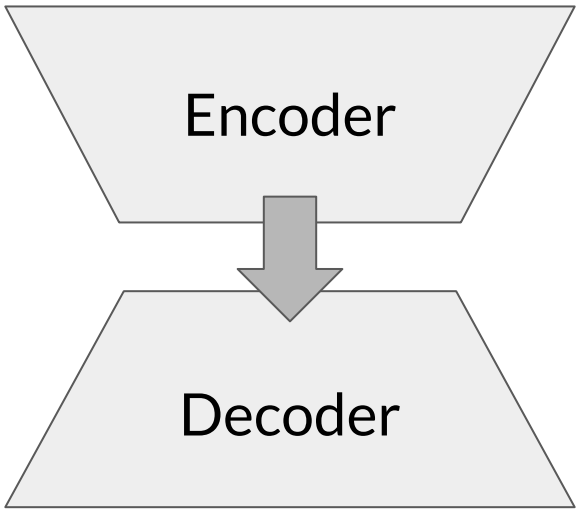
The main ingredient of FOOD is FOOD, and it also contains FOOD. The dish comes from the COUNTRY which is led by president PERSON.

DBpedia Semantic Types

- Problematic!
  - Many semantic types for one entity
  - Noisy; e.g. USA → PERSON
  - Not clear how to deal with hierarchy of semantic types (LEADER → PERSON → AGENT)
- Solution (schema of properties)
  - Each property is associated with a schema
  - For example,
    - leaderName (domain: POPULATEDPLACE, range: PERSON)
  - Use the class of domain and range instead of querying DBpedia

Linearization

COUNTRY leaderName PERSON FOOD region COUNTRY FOOD ingredient FOOD FOOD ingredient FOOD



The main ingredient of FOOD is FOOD, and it also contains FOOD. The dish comes from the COUNTRY which is led by president PERSON.

Automated Metrics for MT Evaluation

Idea of MT Evaluation Metrics :

Compare output of an MT system to a "reference"(usually human) translation.

How close is the MT output to the reference translation?

Advantages:

- Fast and cheap, minimal human labor, no need for bilingual speakers
- Can be used on an ongoing basis during system development to test changes

Disadvantages:

- Current metrics are still relatively crude, do not distinguish well between subtle differences in systems
- Individual sentence scores are often not very reliable, aggregate scores on a large test set are more stable

BLEU (bilingual evaluation understudy)

BLEU : is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another.
Quality is considered to be the correspondence between a machine's output and that of a human.

Scores are calculated for individual translated segments—generally sentences—by comparing them with a set of good quality reference translations.
Those scores are then averaged over the whole corpus to reach an estimate of the translation's overall quality.

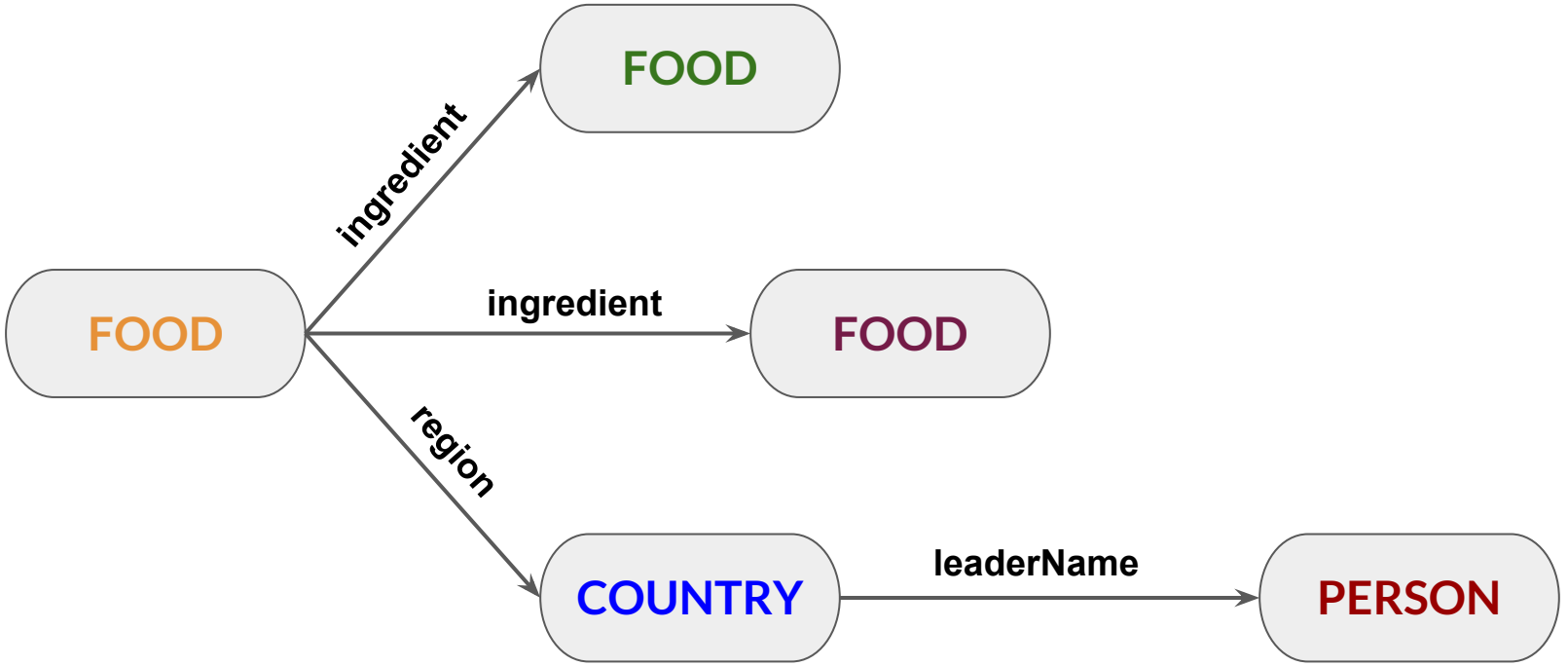
Grammatical correctness are not taken into account.

Baseline

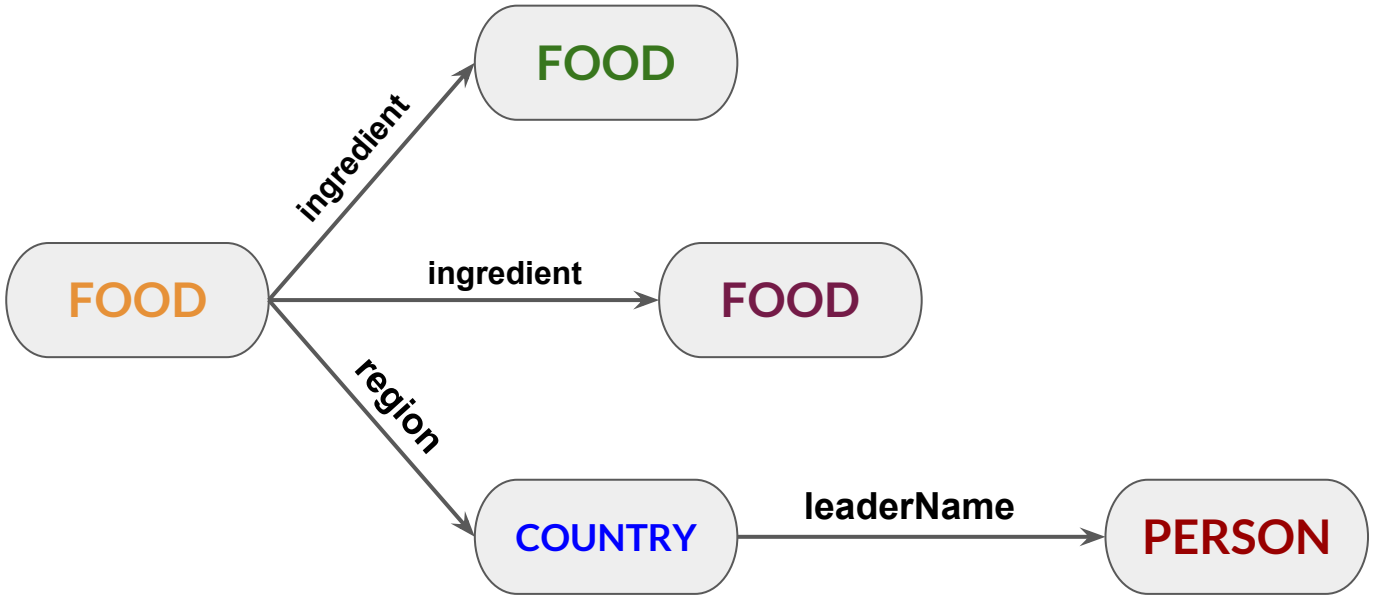
- Used OpenNMT-Py to to train seq2seq model
  - RNN Encoder with 500 hidden units
  - Attentional Decoder with 500 hidden units
  - Dropout = 0.3
  - BLEU score on dev set: 50.08

But ...

- Our input is actually a graph with structure



Structured Sequence




```
(
  ( FOOD
        ( region ( COUNTRY -1 ) )
        ( ingredient (
                        FOOD-1
                        FOOD-2
                    )
        )
  )
  ( COUNTRY-1
        ( leader ( PERSON ) )
  )
)
```

Plan

- Design and code the (fancy) text matching procedure
- Retrieve the schemas for properties
- Design and code an OOP class for RDF graph that encapsulates all procedures
- Build a stronger baseline
- Train NN using structuredrepresentations
- Run error analysis on the output of both models
- Study graph convolutional networks

Translation Edit Rate (TER)

Translation Edit Rate (TER): Number of edits needed to change a system output so that it exactly matches a given reference.

Formula of Translation Edit Rate (TER) :

With more than one reference:

TER = <# of edits> / <avg # of reference words>

TER is calculated against best (closest) reference

Edits include insertions, deletions, substitutions and shifts

All edits count as 1 edit

Shift moves a sequence of words within the hypothesis , where shift of any sequence of words (any distance) is only 1 edit.

Capitalization and punctuation errors are included

Meteor

Meteor consists of two major components: a flexible monolingual word aligner and a scorer.
In **MT** evaluation, hypothesis sentences are aligned to reference sentences. Alignments are then scored to produce sentence and corpus level scores.

Meteor use four flexible matching to align words and phrases,which are :

- **Exact**: Match words if their surface forms are identical.
- **Stem**: Stem words using a language appropriate Snowball Stemmer (Porter, 2001) and match if the stems are identical.
- **Synonym**: Match words if they share membership in any synonym set according to the WordNet database (Miller and Fellbaum, 2007).
- **Paraphrase**: Match phrases if they are listed as paraphrases in a language appropriate paraphrase table.

Differences between BLEU score & TER score :

The TER score measures the amount of editing that a translator would have to perform to change a translation so it exactly matches a reference translation. TER scores range from 0-1. But, with TER a higher score is a sign of more post-editing effort and so LOWER score indicates less post-editing is required. TER gives an indication as to how much post-editing will be required on the translated output of a MT.

The BLEU score measures how many words overlap in a given translation when compared to a reference translation, giving higher scores to sequential words. BLEU scores range from 0-100, the higher the score, the more the translation correlates to a human translation.

BLEU provides some insight into how good the fluency of the output from a MT will be.

Differences between BLEU & Meteor :

- **Difference in computing the matches**: Blue takes a n-gram approach on the surface forms, while Meteor only does unigram matches but uses stemming and synonyms.

- **Difference of oriented type :** If they are precision oriented or recall oriented. Blue only takes into account the precision at various n-gram lengths and uses a length penalty to make up for recall, while Meteor measure where parameters are tuned to get high correlation with human evaluations.