



# Sentiment Analysis via DNNs

Bitu Nejat



# Definitions

By now we know:

- The definition of Sentiment
- What it means to do sentiment analysis
- How Neural Nets work
- Recursive ways for learning the representation of a text
  - Using syntactic tree

# PART 1: Replication of previous results

## Supervised Sentiment Analysis [Socher et al. 2013]

Recursive Deep Models for Semantic Compositionality  
Over a Sentiment Treebank

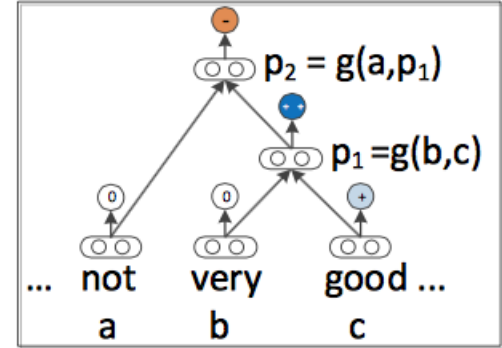
A Sentiment Neural Net

Learnt over binarized syntactic tree

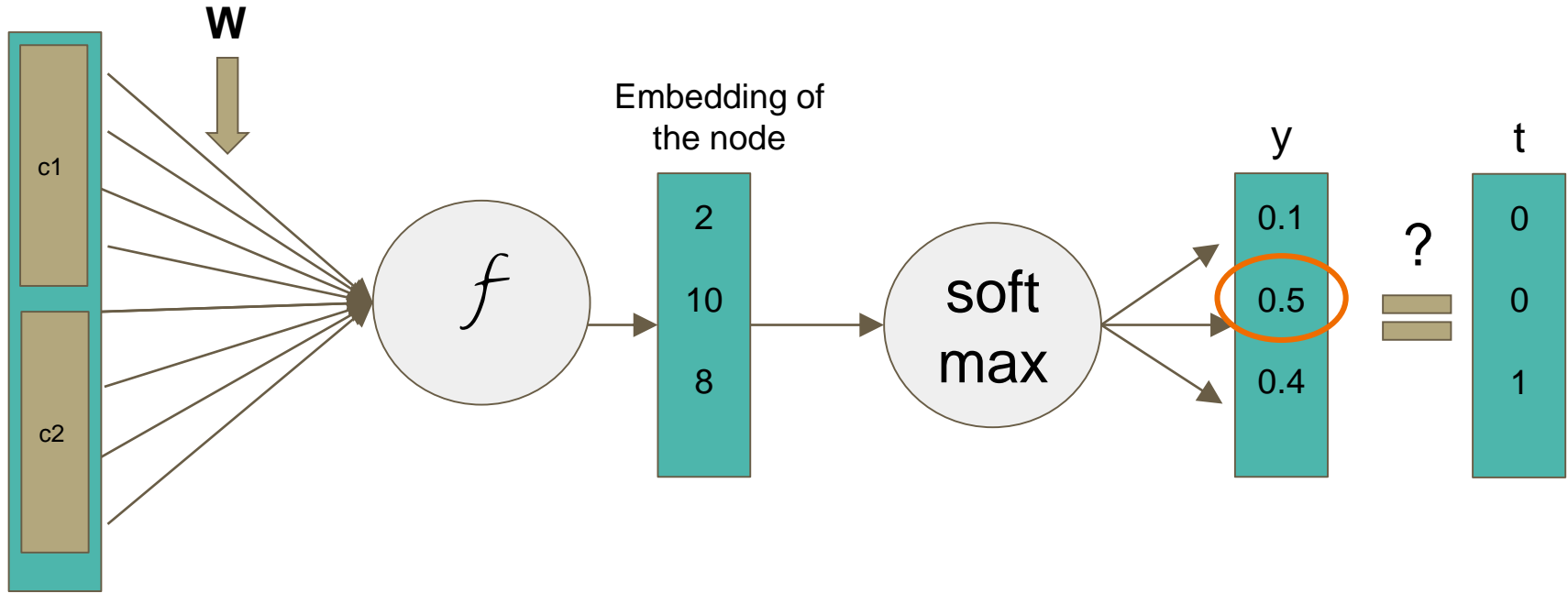
## Sentiment Treebank:

Dataset of 11850 ish sentences and their syntactic trees

Labeled at each node with sentiment values.



# RNN structure



# Progress?

- Writing the code from scratch produced the following results:

Model	Fine-grained	
	All	Root
NB	67.2	41.0
SVM	64.3	40.7
BiNB	71.0	41.9
VecAvg	73.3	32.7
RNN	79.0	43.2
MV-RNN	78.7	44.4
RNTN	<b>80.7</b>	<b>45.7</b>
<b>Bit-RNN</b>	67.3	24.1

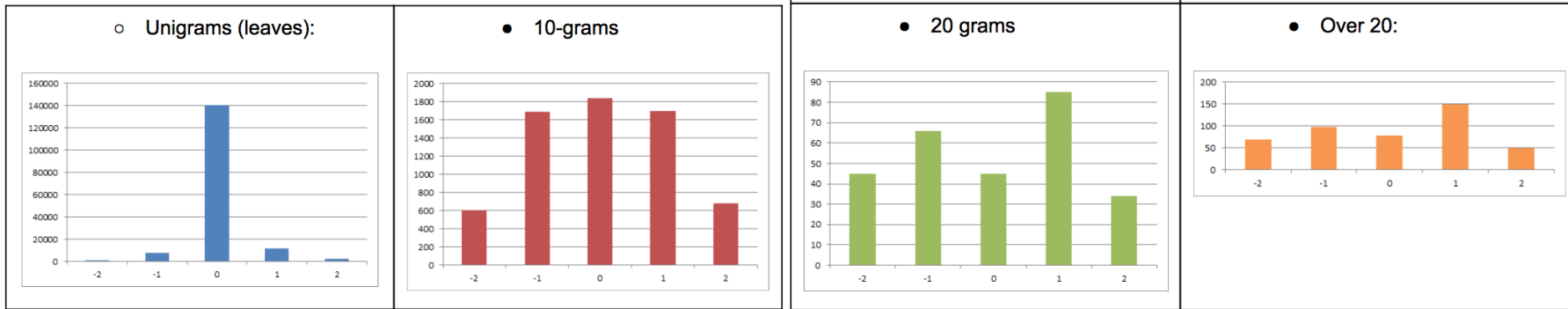
# Progress?

- I re-trained the original model and got the following results:

Model	Fine-grained	
	All	Root
NB	67.2	41.0
SVM	64.3	40.7
BiNB	71.0	41.9
VecAvg	73.3	32.7
RNN	79.0	43.2
MV-RNN	78.7	44.4
RNTN	<b>80.7</b>	<b>45.7</b>
<b>Retrained-RNN</b>	78.8	39.5

# Observations of dataset

- Unbalanced data



- Incorrect sentiment labels?

- “Wasabi is a good place to start.” → Neutral

# PART 2: Learn Sentiment over Discourse Tree

Sentiment Analysis

[Bhatia et al. 2015]

Better document-level sentiment analysis  
from RST discourse parsing.

A Sentiment Neural Net

Learnt over discourse tree



# Part 2: Neural Net Description

The task specific vector representation of two nodes

Sentiment Neural Net

Sentiment of the parent node



# Dataset Creation

Create Discourse Trees for every sentence in Sentiment Treebank

# Dataset Creation

Create Discourse Trees for every sentence in Sentiment Treebank

**On-going:** ~12,000 sentences takes a long time! Already have about 3600 trees.

```
1 ( Root (span 1 4)
2   ( Nucleus (span 1 2) (rel2par Same-Unit)
3     ( Nucleus (leaf 1) (rel2par span) (text _!If you sometimes like to go to the movies_!) )
4     ( Satellite (leaf 2) (rel2par Enablement) (text _!to have fun ,_!) )
5   )
6   ( Nucleus (span 3 4) (rel2par Same-Unit)
7     ( Nucleus (leaf 3) (rel2par span) (text _!Wasabi is a good place_!) )
8     ( Satellite (leaf 4) (rel2par Elaboration) (text _!to start ._!) )
9   )
10 )
```

# Dataset Creation

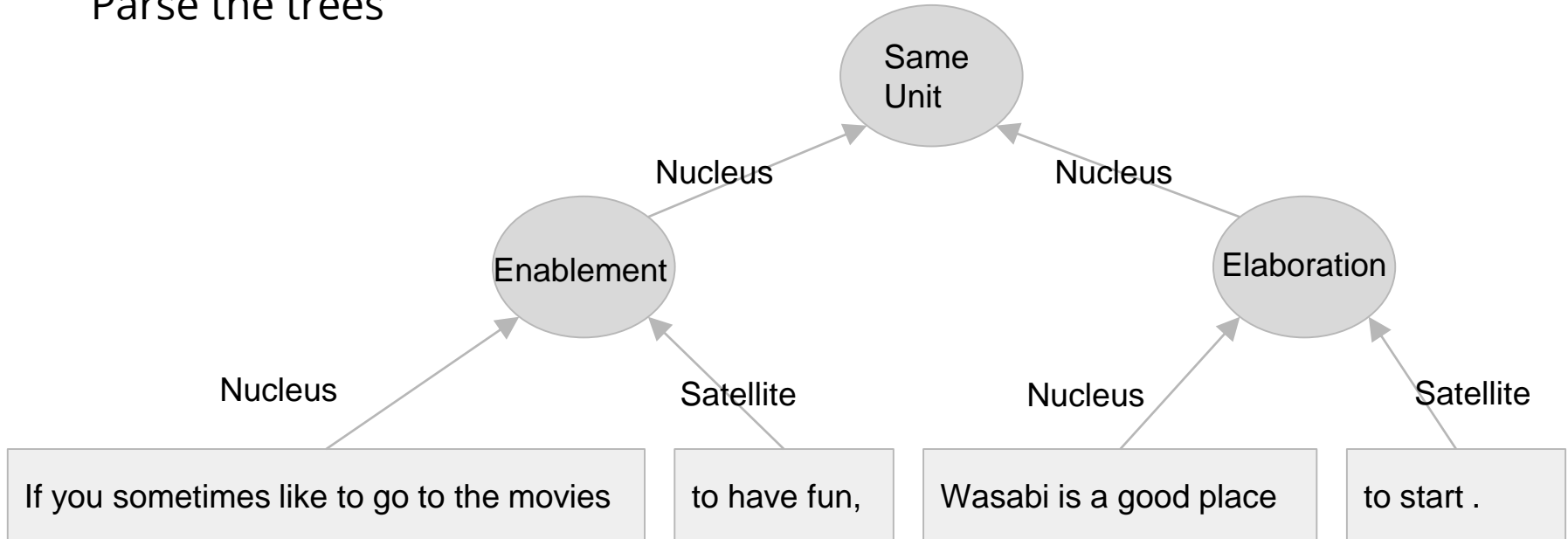
Create Discourse Trees for every sentence in Sentiment Treebank

Parse the trees

# Dataset Creation

Create Discourse Trees for every sentence in Sentiment Treebank

Parse the trees

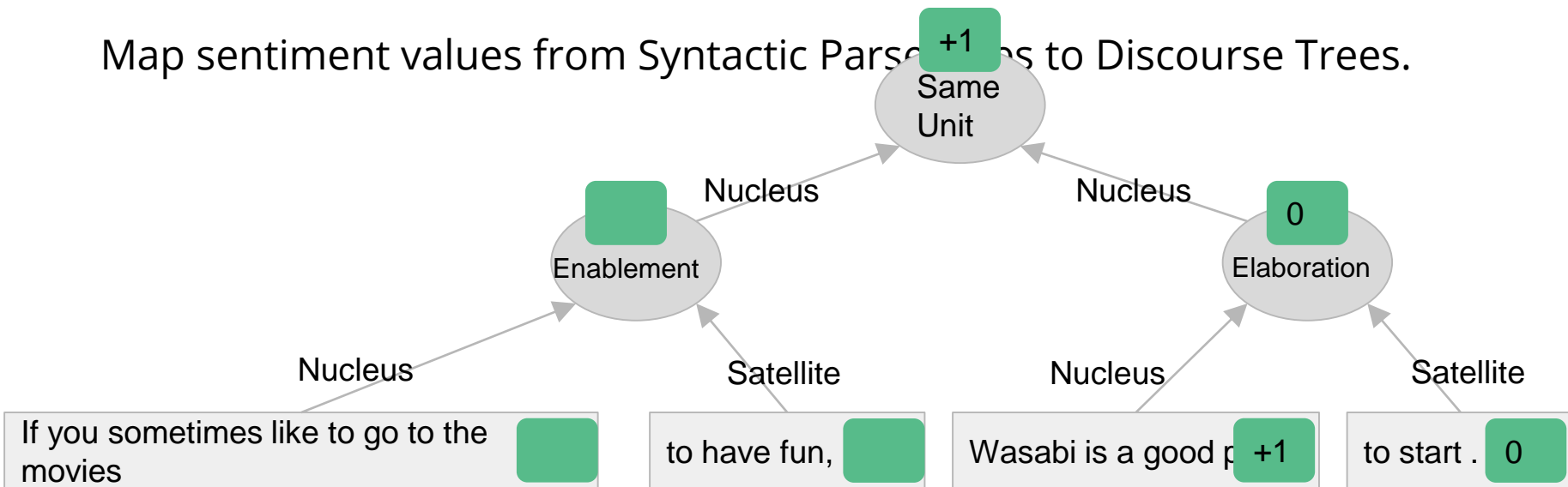


# Dataset Creation

Create Discourse Trees for every sentence in Sentiment Treebank

Parse the trees

Map sentiment values from Syntactic Parse trees to Discourse Trees.



# Dataset Creation

Create Discourse Trees for every sentence in Sentiment Treebank

Parse the trees

Map sentiment values from Syntactic Parse trees to Discourse Trees.

**Problem:** Syntactic Trees don't properly map to Discourse Trees.

Exploring ways that a reasonable way of mapping can be done.

# Next steps:

- Once the dataset is ready, training can be done by:
- Using Skip-Thought vectors to initialize the representation for each EDU,
  - Skip-Thoughts are a form of Recurrent Neural Networks
  - Input: a phrase
  - Output: vector representation for that phrase
- For each parent,
  - use the representation of its two children → RecursiveNN approach.
  - Use Skip-Thought Vectors to initialize representation → RecurrentNN approach



# Steps until done:

- Set up Skip-Thought vectors
- Train over Discourse Tree with Skip-Thought vector initialization.

Questions? :)

