Lattice and Hypergraph MERT

Graham Neubig
Nara Institute of Science and Technology (NAIST)
12/20/2012

Papers Introduced:

- “Lattice-based Minimum Error Rate Training for Statistical Machine Translation”
  Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, Jakob Uszkoreit (Google)
  EMNLP 2008
- “Efficient Minimum Error Rate Training and Minimum Bayes-Risk Decoding for Translation Hypergraphs and Lattices”
  Shankar Kumar, Wolfgang Macherey, Chris Dyer, Franz Och (Google/University of Maryland)
  ACL-IJCNLP 2009

Summary

- Minimum error rate training (MERT) is used to train the parameters for machine translation
- Normal MERT uses n-best lists
- However, there is not enough diversity in n-best lists,
  → unstable training & large accuracy fluctuations
- As a solution these papers perform MERT over
  - lattices for phrase-based translation [Macherey+ 08]
  - hypergraphs for tree-based translation [Kumar+ 09]
- This leads to more stable training in fewer iterations

Tuning/MERT

Tuning

- Scores of translation, reordering, and language models

	LM	TM	RM		
○ Taro visited Hanako	-4	-3	-1	-8	
X the Taro visited the Hanako	-5	-4	-1	-10	
X Hanako visited Taro	-2	-3	-2	-7	← Best Score X

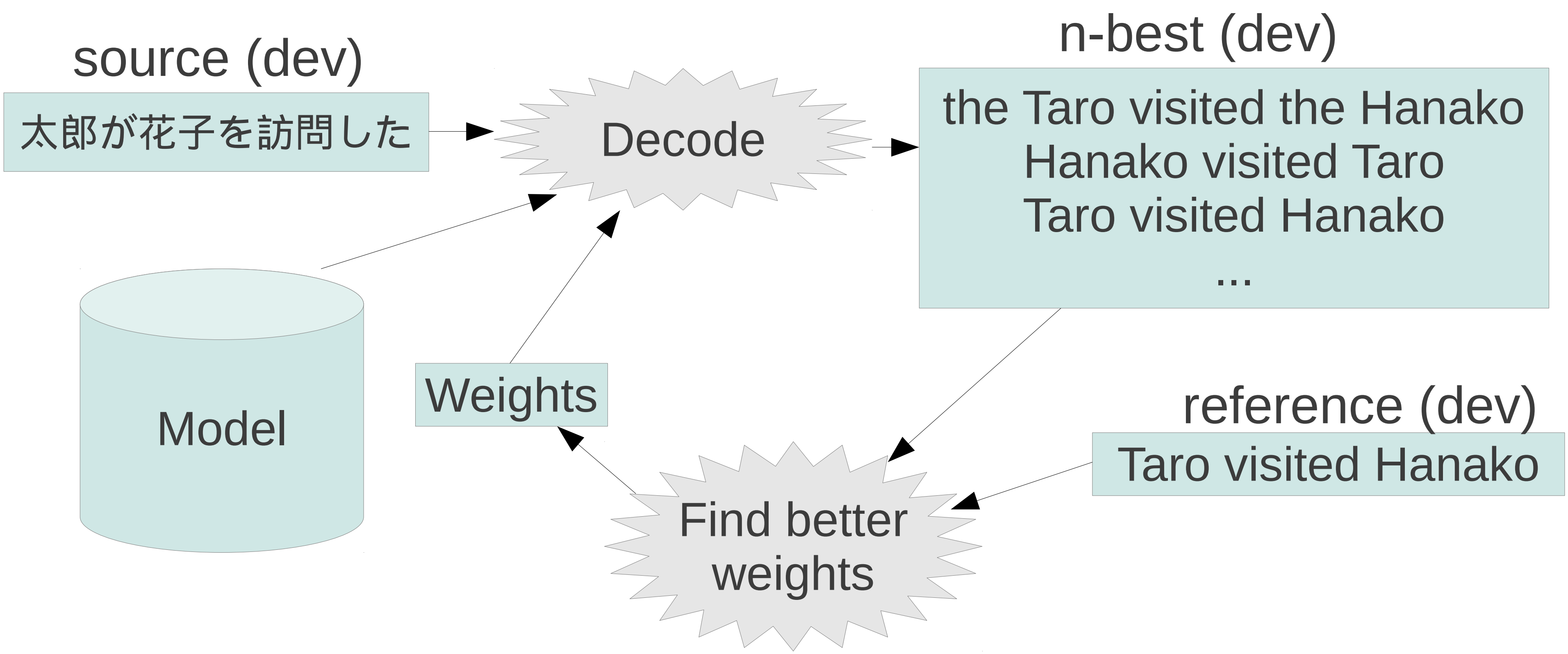
- If we add weights, we can get better answers:

	LM	TM	RM		
○ Taro visited Hanako	0.2*-4	0.3*-3	0.5*-1	-2.2	← Best Score ○
X the Taro visited the Hanako	0.2*-5	0.3*-4	0.5*-1	-2.7	
X Hanako visited Taro	0.2*-2	0.3*-3	0.5*-2	-2.3	

- Tuning finds these weights: $w_{LM}=0.2$ $w_{TM}=0.3$ $w_{RM}=0.5$

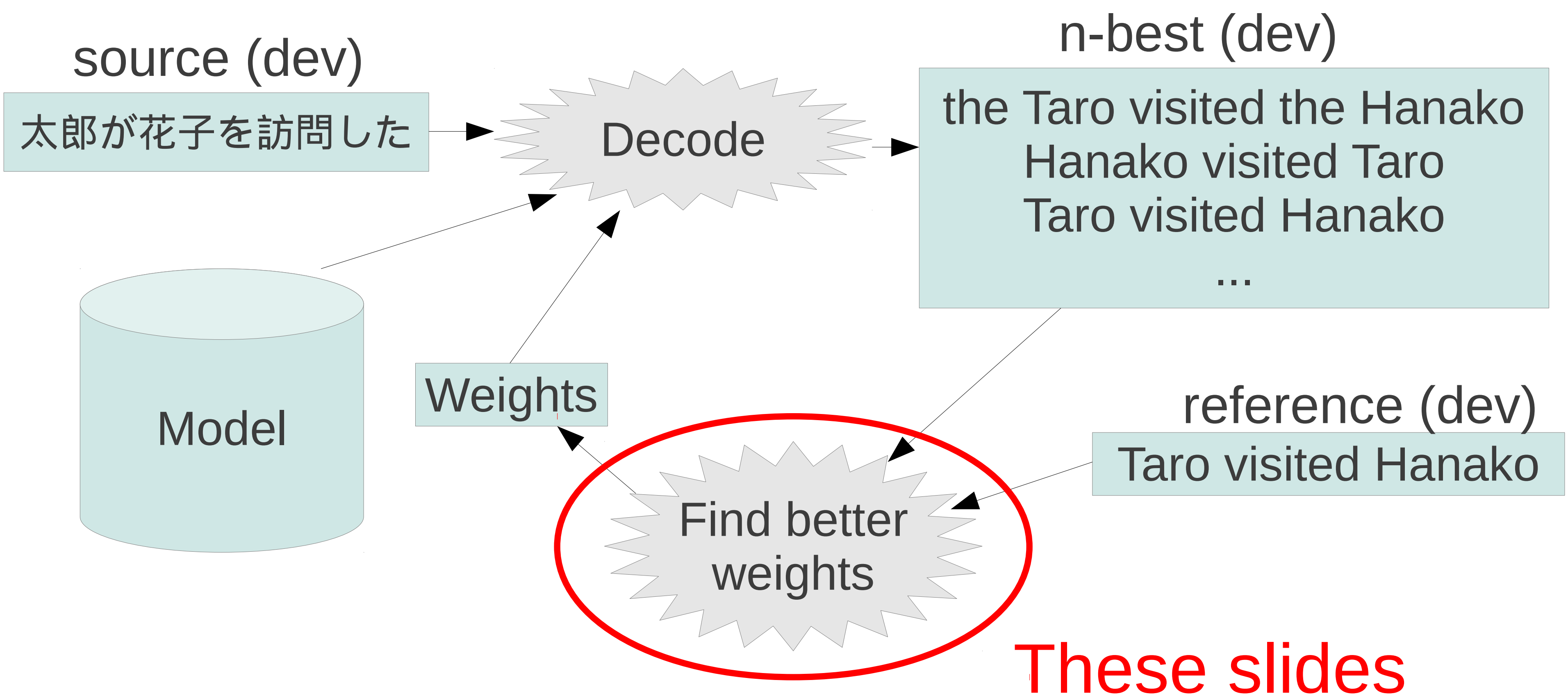
MERT

- MERT performs iterations to increase the score [Och 03]



MERT

- MERT performs iterations to increase the score [Och 03]



MERT Weight Update:

- Adjust one weight at a time

	Weights			Score
	w_{LM}	w_{TM}	w_{RM}	
Initial:	0.1	0.1	0.1	0.20
Optimize w_{LM}:	↓			
	0.4	0.1	0.1	0.32
Optimize w_{TM}:		↓		
	0.4	0.1	0.1	0.32
Optimize w_{RM}:			↓	
	0.4	0.1	0.3	0.4
Optimize w_{LM}:	↓			
	0.35	0.1	0.3	0.41
Optimize w_{TM}:		↓		

Updating One Weight:

- We start with:

  n-best list

f_1	ϕ_{LM}	ϕ_{TM}	ϕ_{RM}	BLEU*
$e_{1,1}$	1	0	-1	0
$e_{1,2}$	0	1	0	1
$e_{1,3}$	1	0	1	0

f_2	ϕ_{LM}	ϕ_{TM}	ϕ_{RM}	BLEU*
$e_{2,1}$	1	0	-2	0
$e_{2,2}$	3	0	1	0
$e_{2,3}$	2	1	2	1

  fixed weights:

$$w_{LM}=-1,\ w_{TM}=1$$

  weight to be adjusted:

$$w_{RM}=???$$

* Calculating BLEU for one sentence is a bit simplified, usually we compute for the whole corpus

Updating One Weight:

- Next, transform each hypothesis into lines:

$$y = a\,x + b$$

- Where:
    - *a* is the value of the feature to be adjusted
    - *b* is the weighted sum of the fixed features
    - *x* is the weight to be adjusted (unknown)

Updating One Weight:

- Example:

w_{LM}=-1, w_{TM}=1, w_{RM}=???

$$y = a x + b$$

$$a = \phi_{RM} \qquad b = w_{LM} \phi_{LM} + w_{TM} \phi_{TM}$$

f_1	ϕ_{LM}	ϕ_{TM}	ϕ_{RM}
$e_{1,1}$	1	0	-1
$e_{1,2}$	0	1	0
$e_{1,3}$	1	0	1

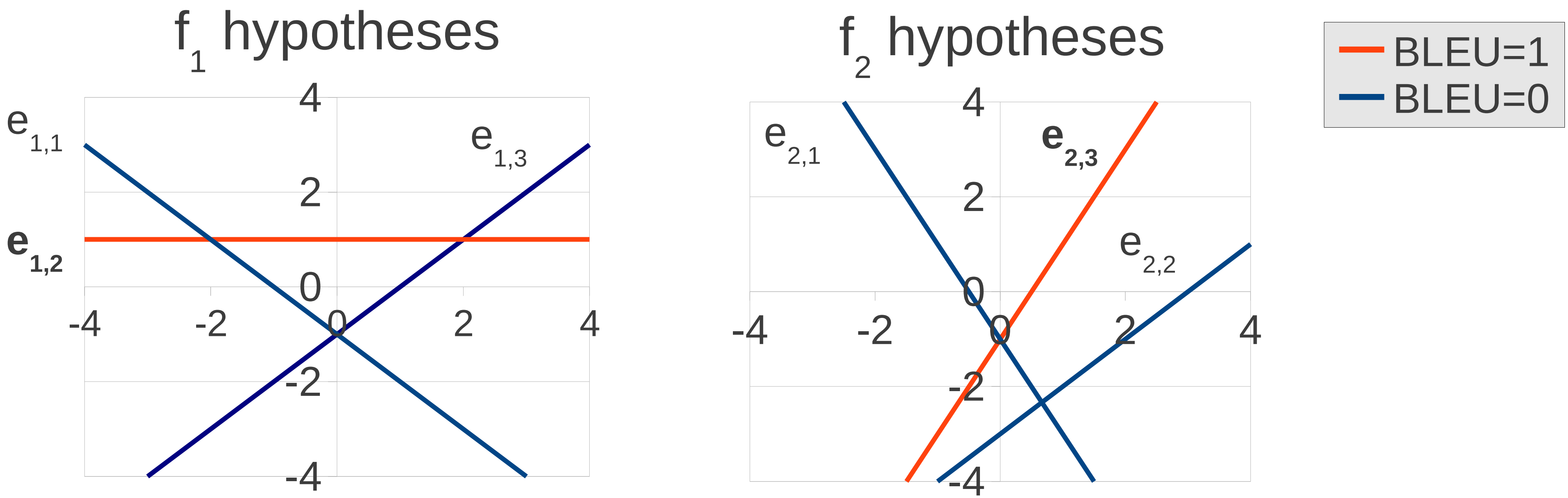
$a_{1,1}$=-1 $b_{1,1}$=-1

$a_{1,2}$=0 $b_{1,2}$=1

$a_{1,3}$=1 $b_{1,3}$=-1

Updating One Weight:

- Draw lines on a graph: $y = a\,x + b$



$a_{1,1}$=-1 $b_{1,1}$=-1

$a_{1,2}$=0 $b_{1,2}$=1

$a_{1,3}$=1 $b_{1,3}$=-1

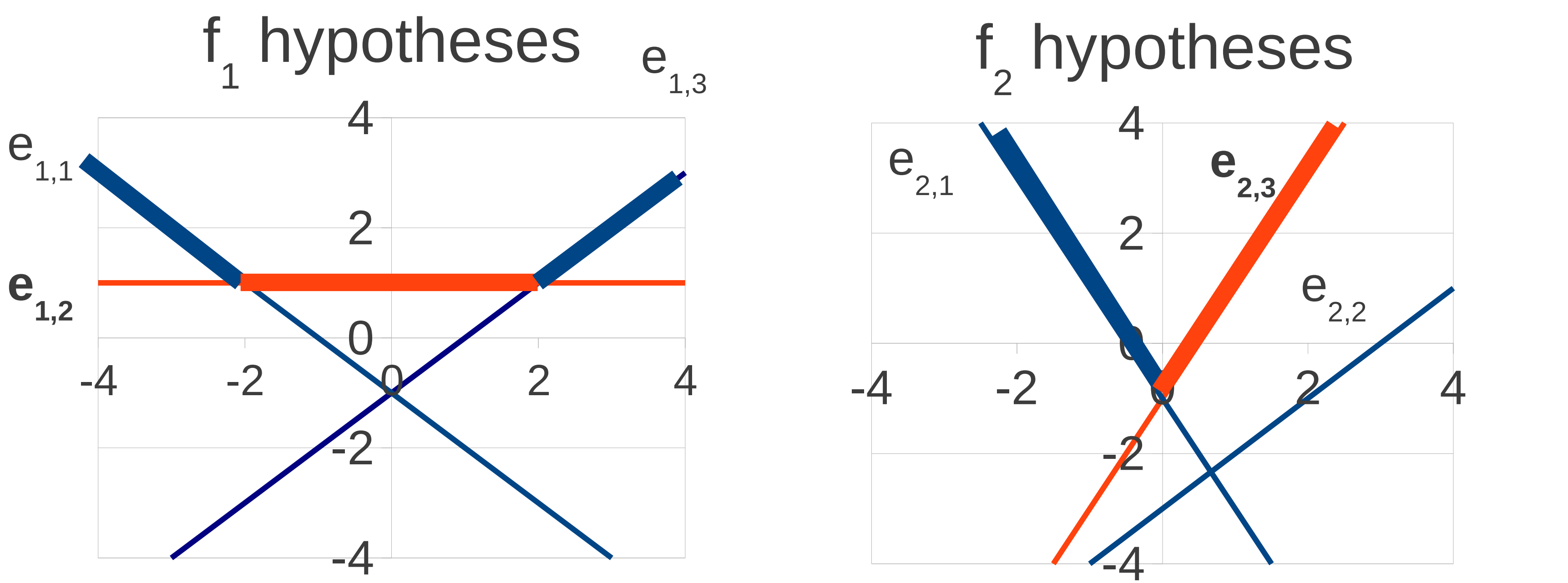
$a_{2,1}$=-2 $b_{2,1}$=-1

$a_{2,2}$=1 $b_{2,2}$=-3

$a_{2,3}$=-2 $b_{2,3}$=1

Updating One Weight:

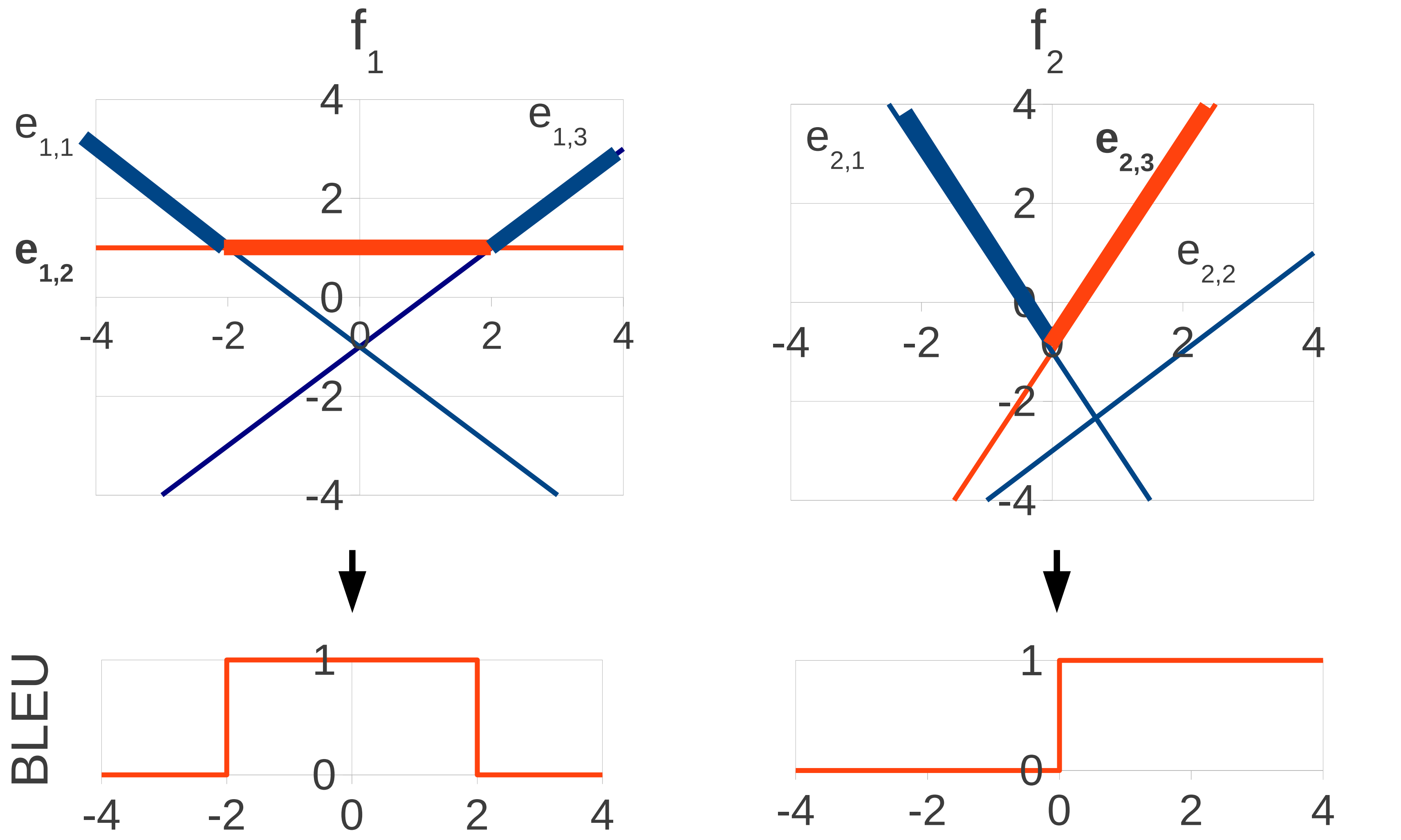
- Find the lines that are highest for each range of *x*:



- This is called the convex hull (or upper envelope)

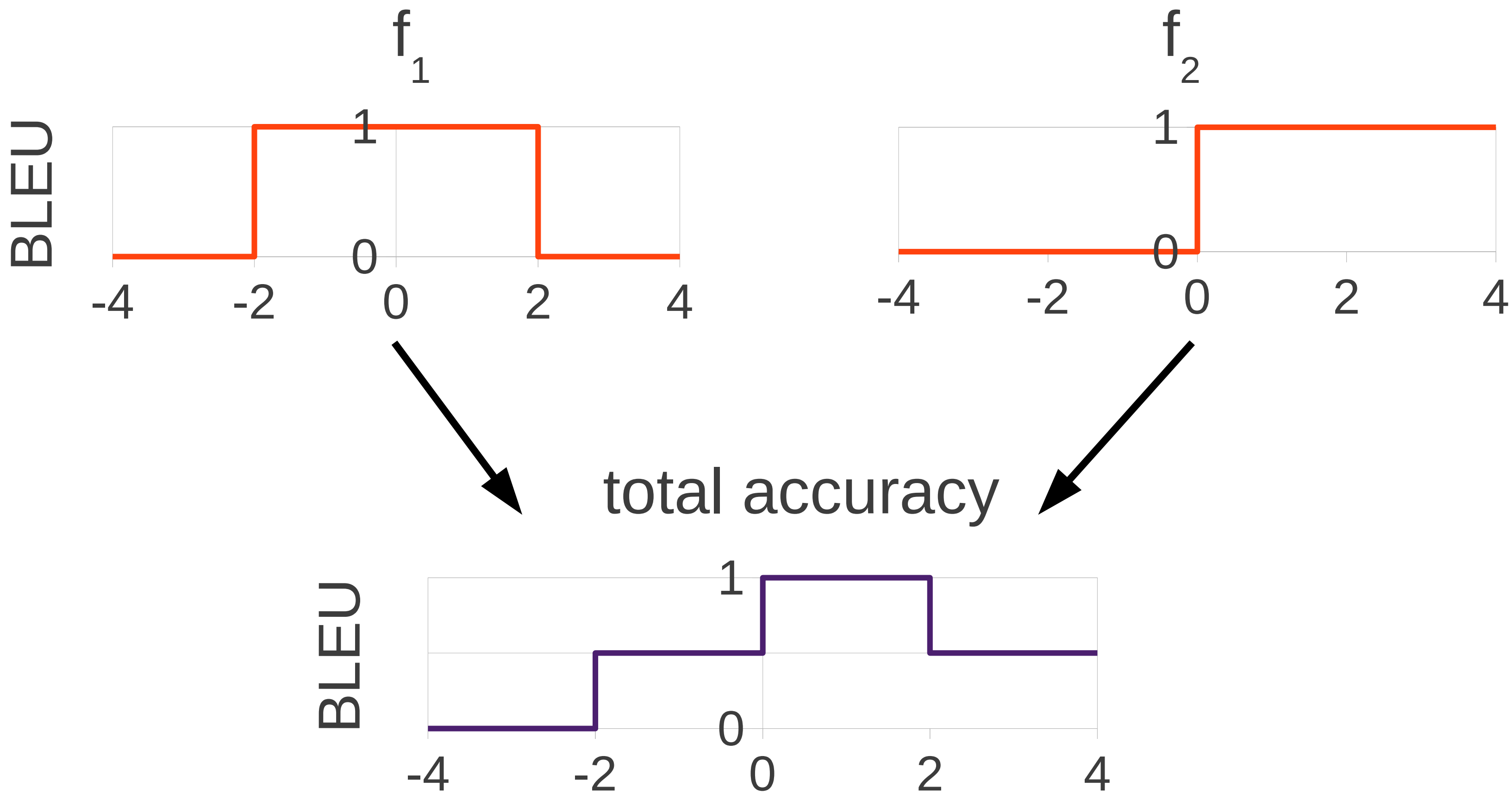
Updating One Weight:

- Using the convex hull, find scores at each range:



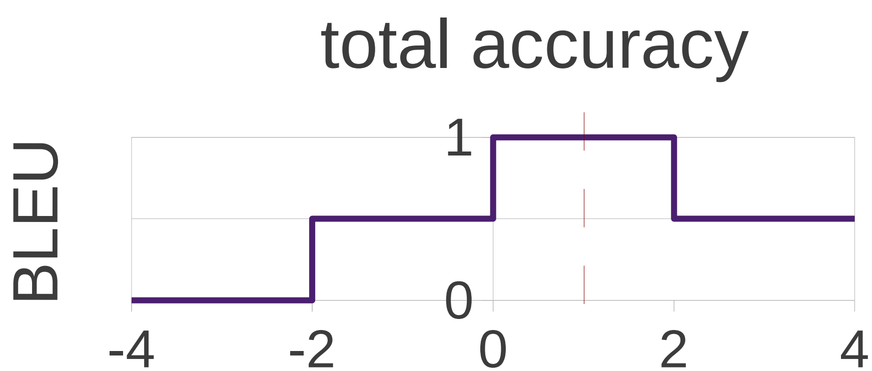
Updating One Weight:

- Combine multiple sentences into a single error plane:



Updating One Weight:

- Choose middle of best region:



$w_{RM} \leftarrow 1.0$

Summary

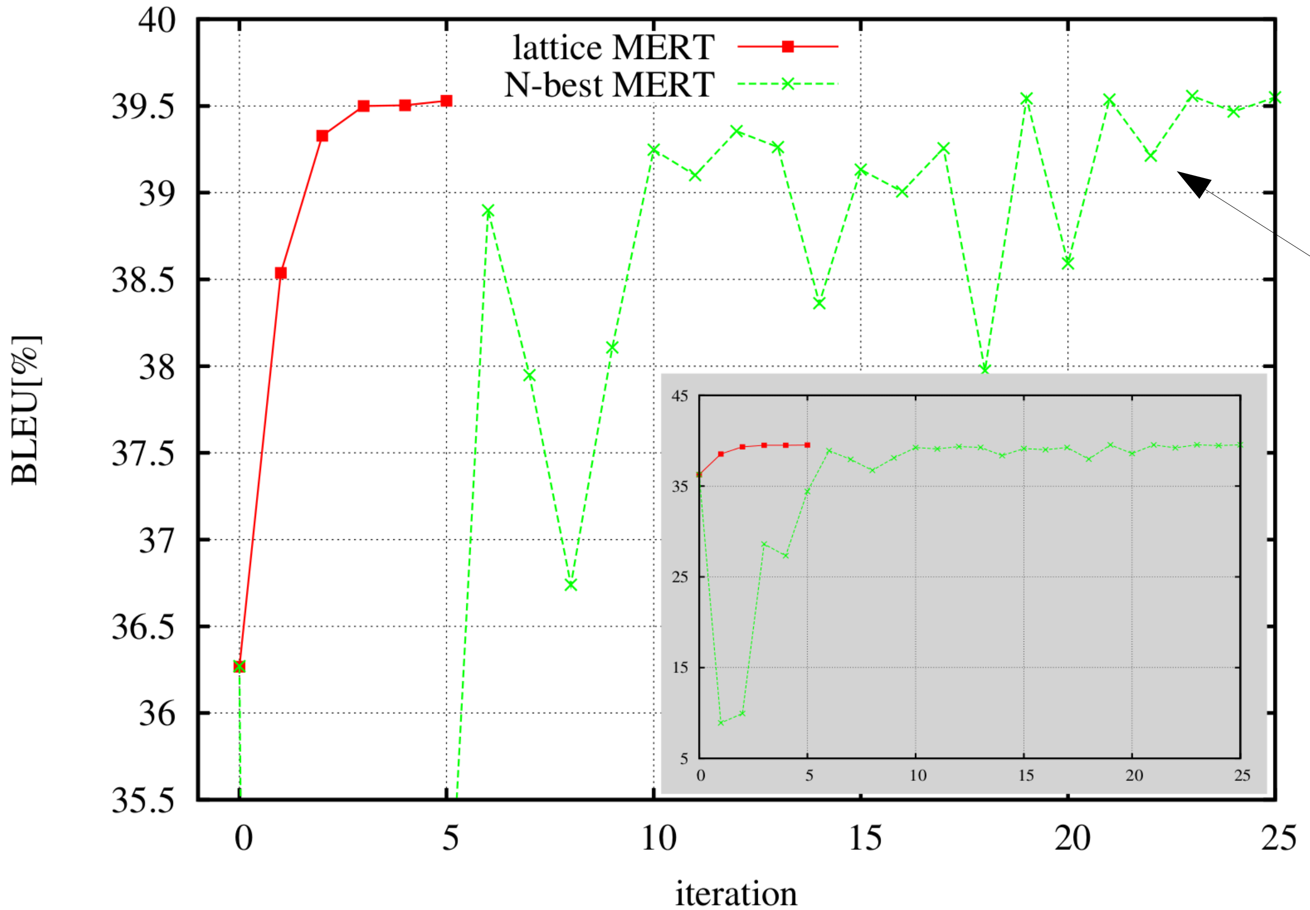
- For each sentence:
  - Create lines for each n-best hypothesis
  - Combine lines and find upper envelope
  - Transform upper envelope into error surface
- Combine error surfaces into one
- Find the range with the highest score
- Set the weight to the middle of the range

Summary

- For each sentence: **Problem!** (not enough diversity)
  - Create lines for each **n-best** hypothesis
  - Combine lines and find upper envelope
  - Transform upper envelope into error surface
- Combine error surfaces into one
- Find the range with the highest score
- Set the weight to the middle of the range

Result of Lack of Diversity

- Unstable training:



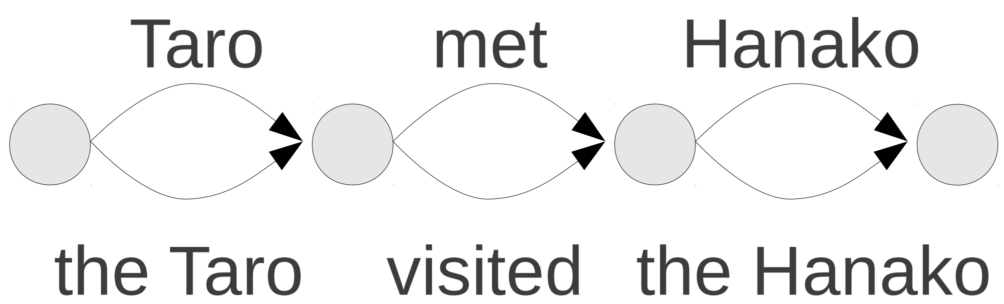
Traditional MERT in Green

[Macherey 08]

Lattice MERT

Translation Lattice

- Represent many hypotheses compactly:

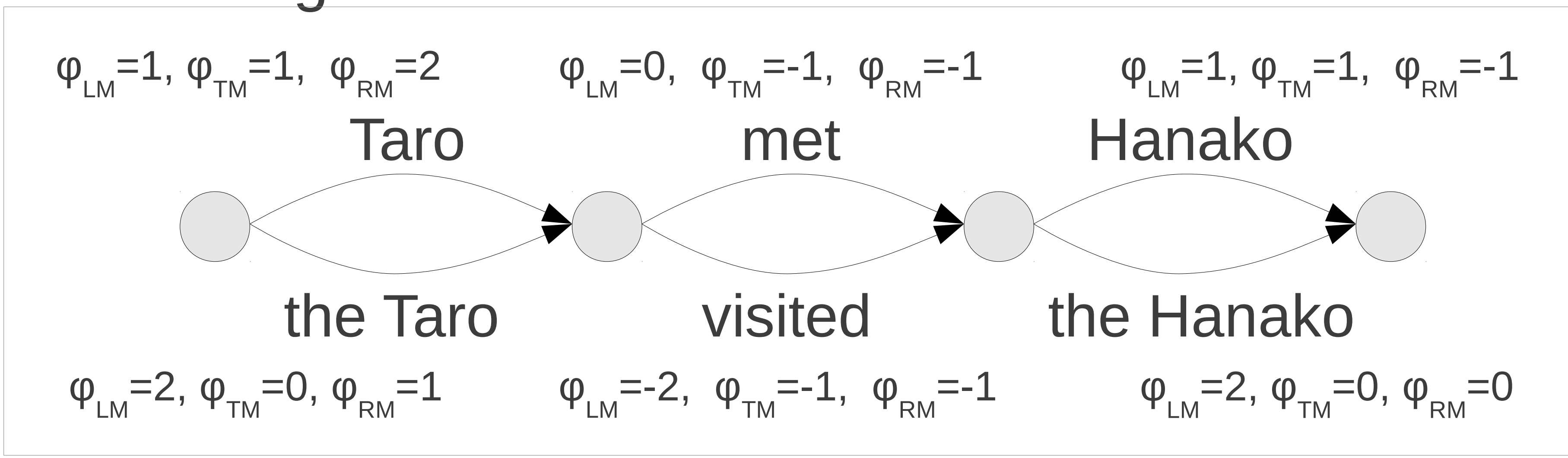


8 hypotheses
in only
6 edges

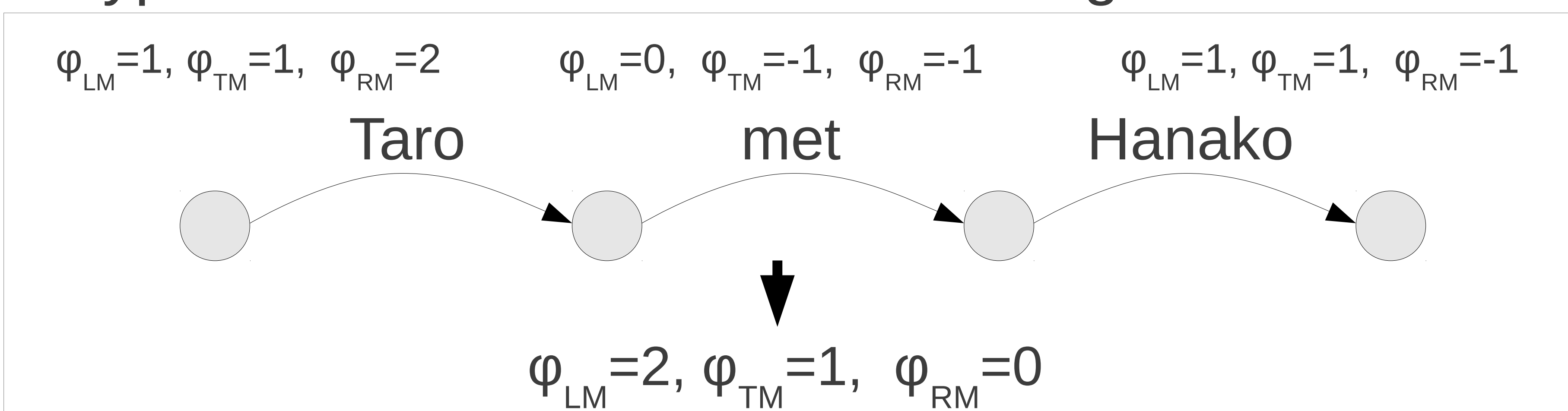
- MERT on lattices can solve the diversity problem

Factoring Feature Functions

- Each edge in the lattice has a feature value:



- Hypothesis's features are sum of edge features:

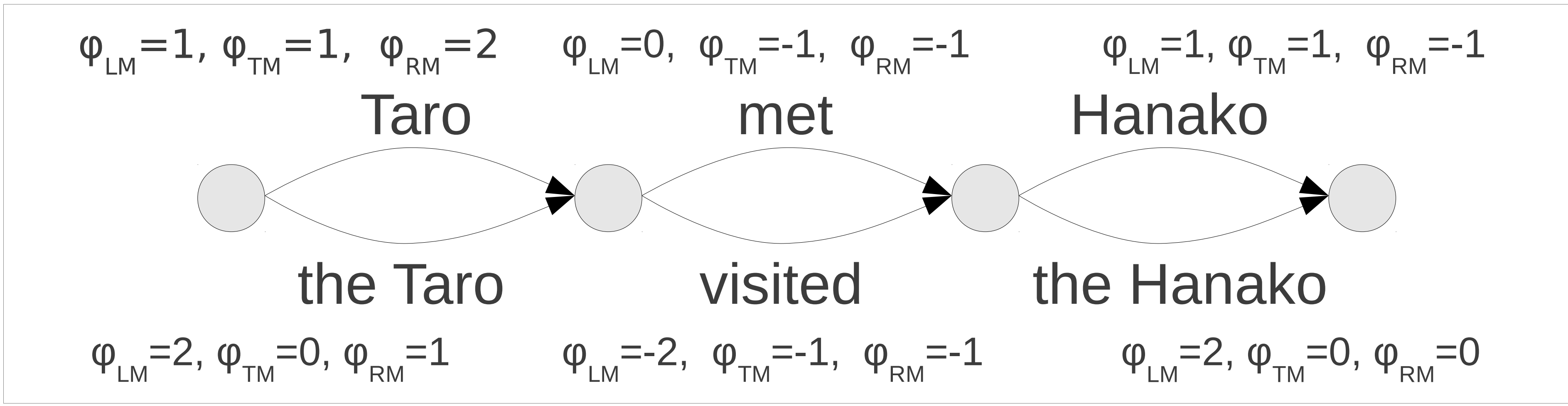


MERT on Lattices:

Only different part!!

- For each sentence:
  - **Transform each edge into lines**
  - **Find the upper envelope for the lattice**
  - Transform upper envelope into error surface
- Combine error surfaces into one
- Find the range with the highest score
- Set the weight to the middle of the range

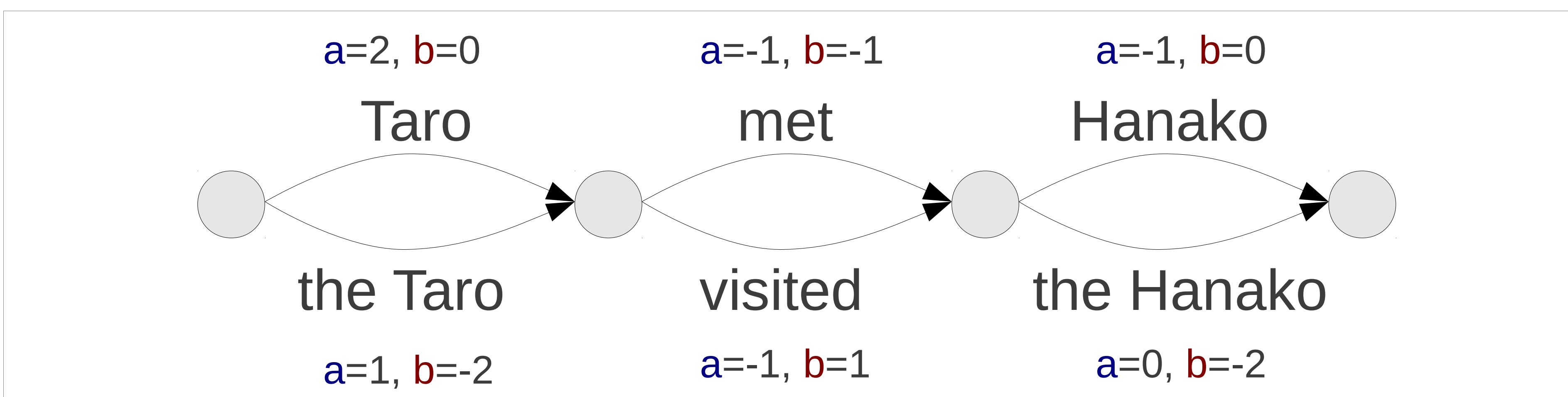
First, Transform each Edge into Lines



w_{LM}=-1, w_{TM}=1, w_{RM}=???

$$y = a\,x + b$$

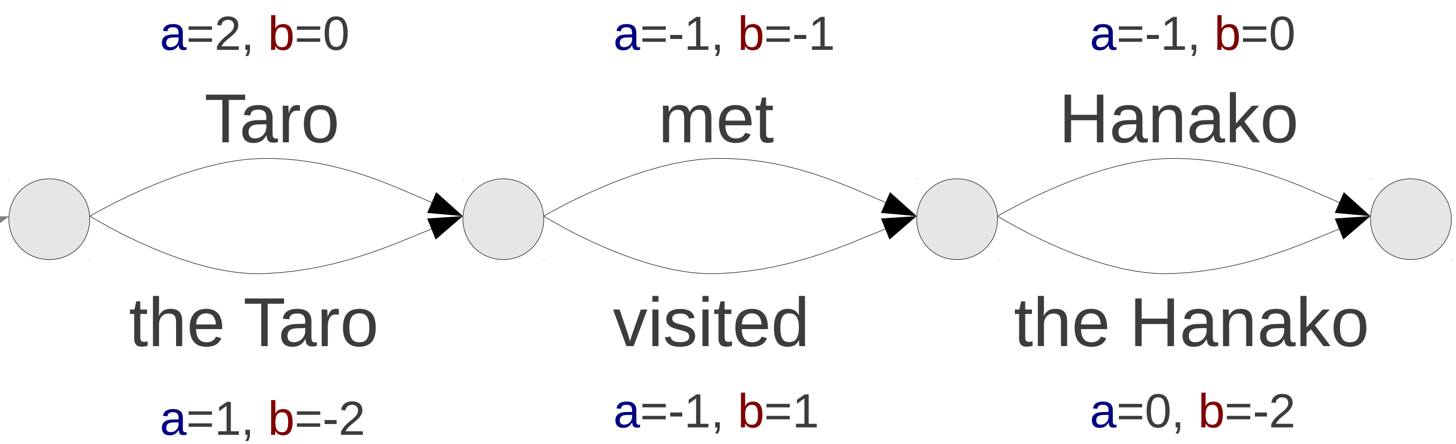
$$a = \phi_{RM} \quad b = w_{LM}\,\phi_{LM} + w_{TM}\,\phi_{TM}$$

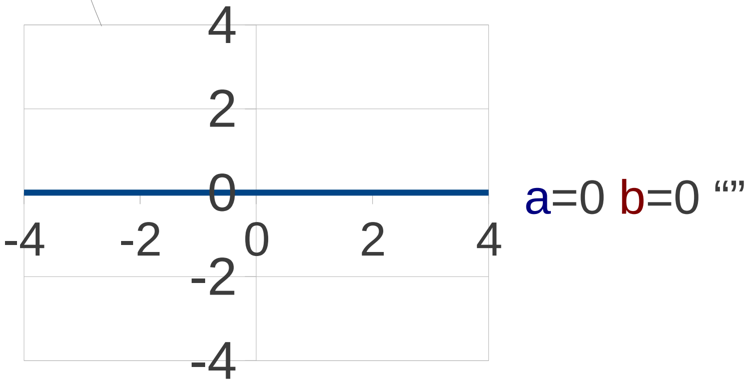


Finding the Upper Envelope for Lattices:

- Can be done with dynamic programming
- 1) Start with flat envelope for initial node
- 2) Calculate upper envelope for next node using previous nodes

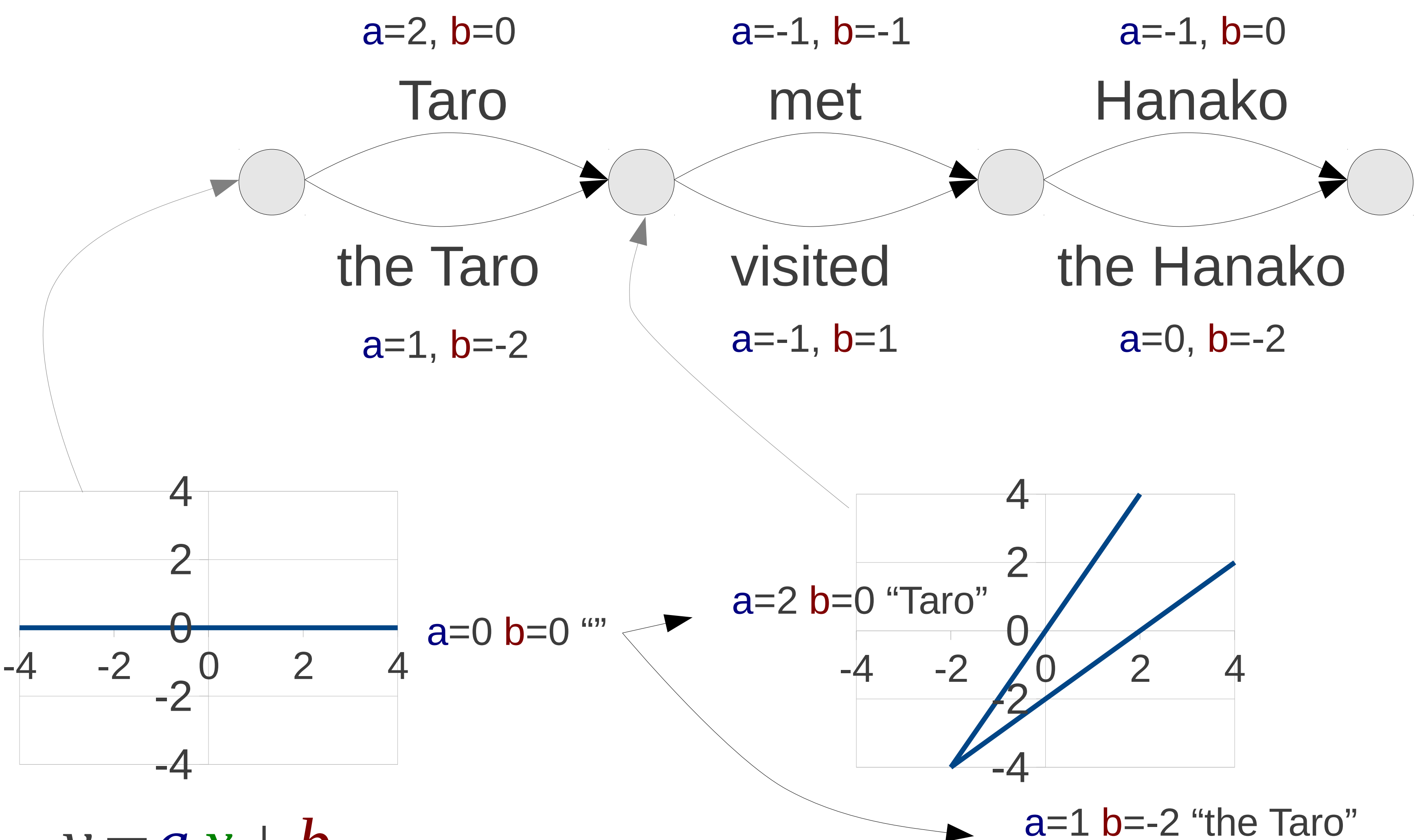
Start with Flat Envelope





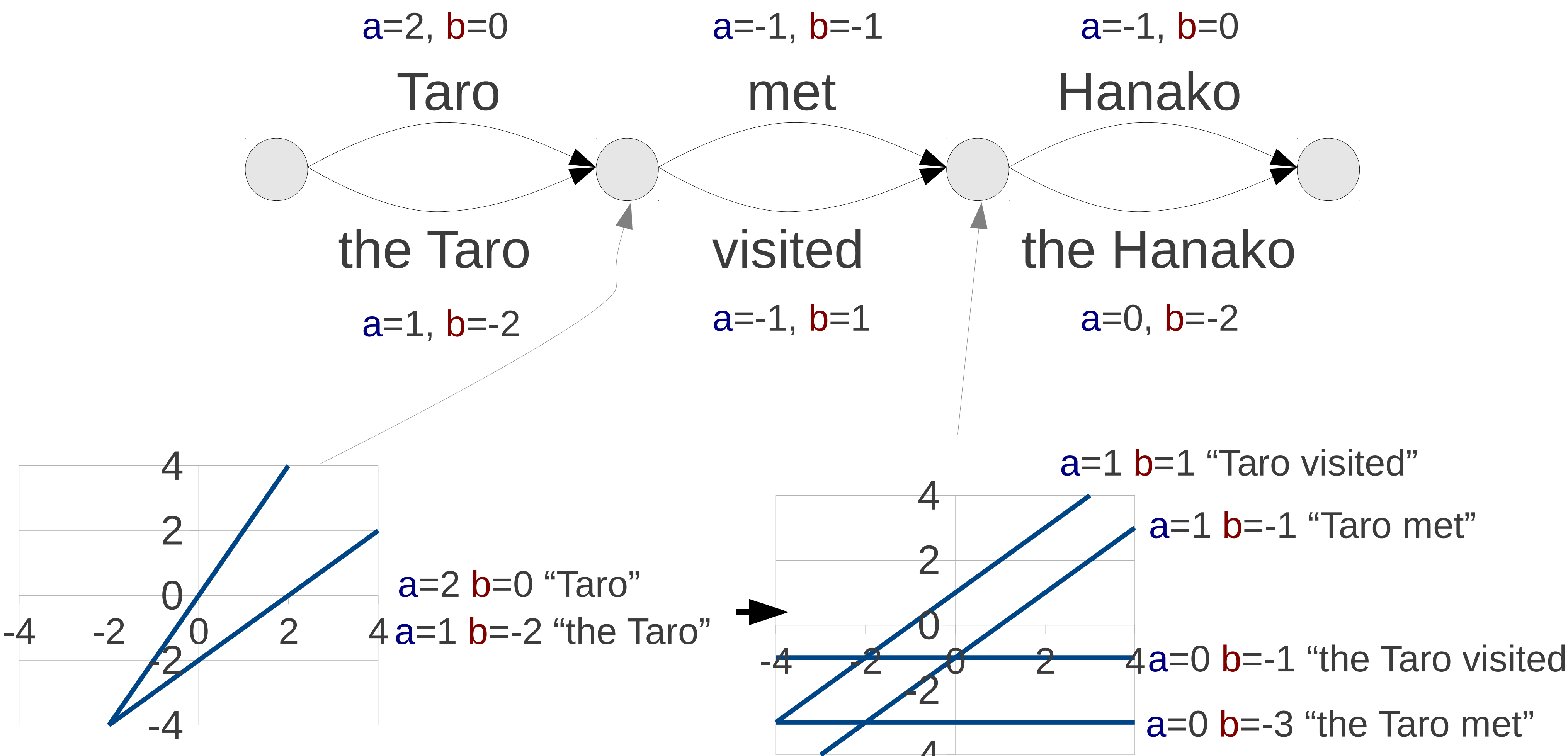
$$y = a x + b$$

Add First Node



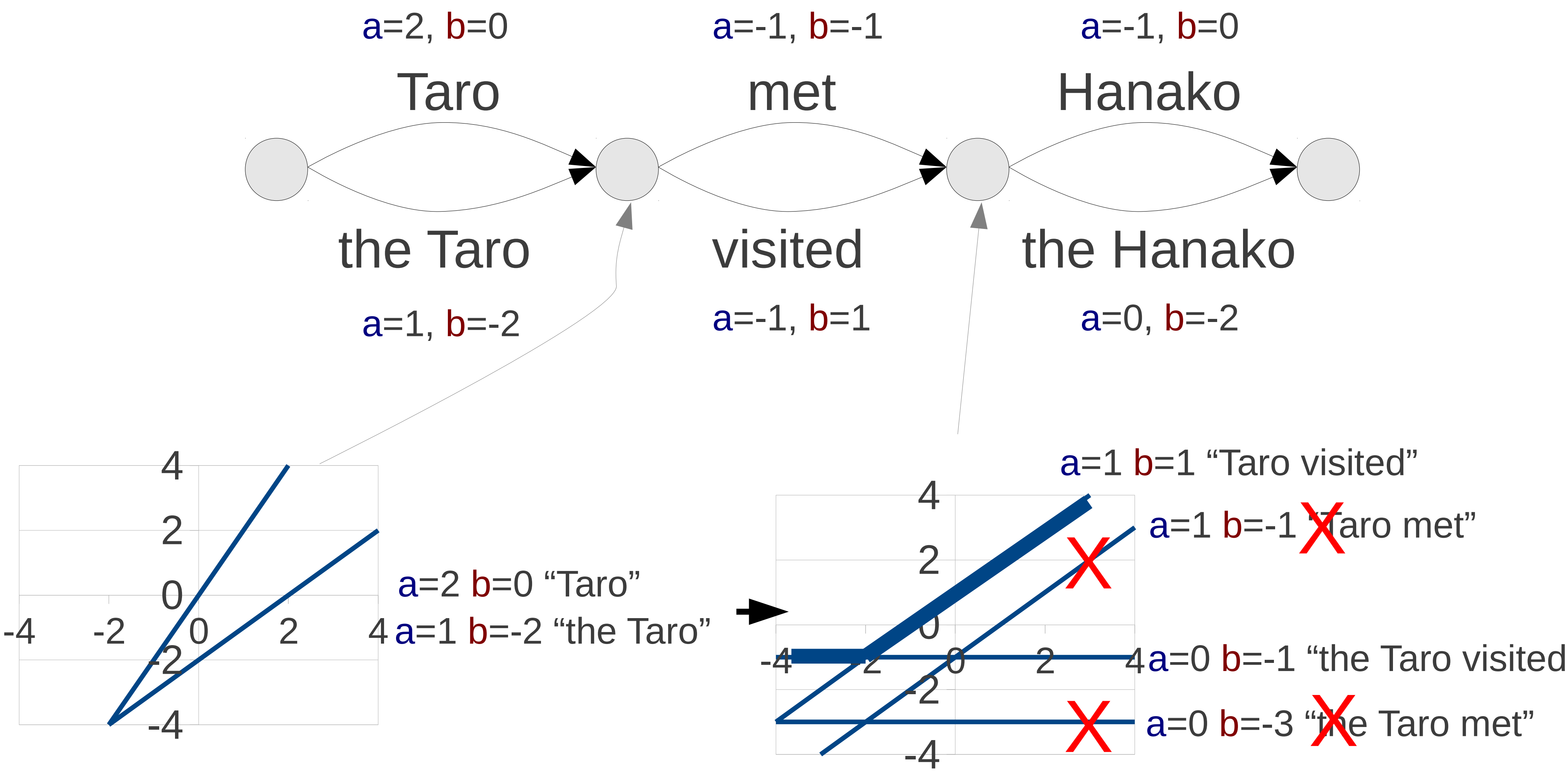
$$y = a x + b$$

Add Second



$$y = a x + b$$

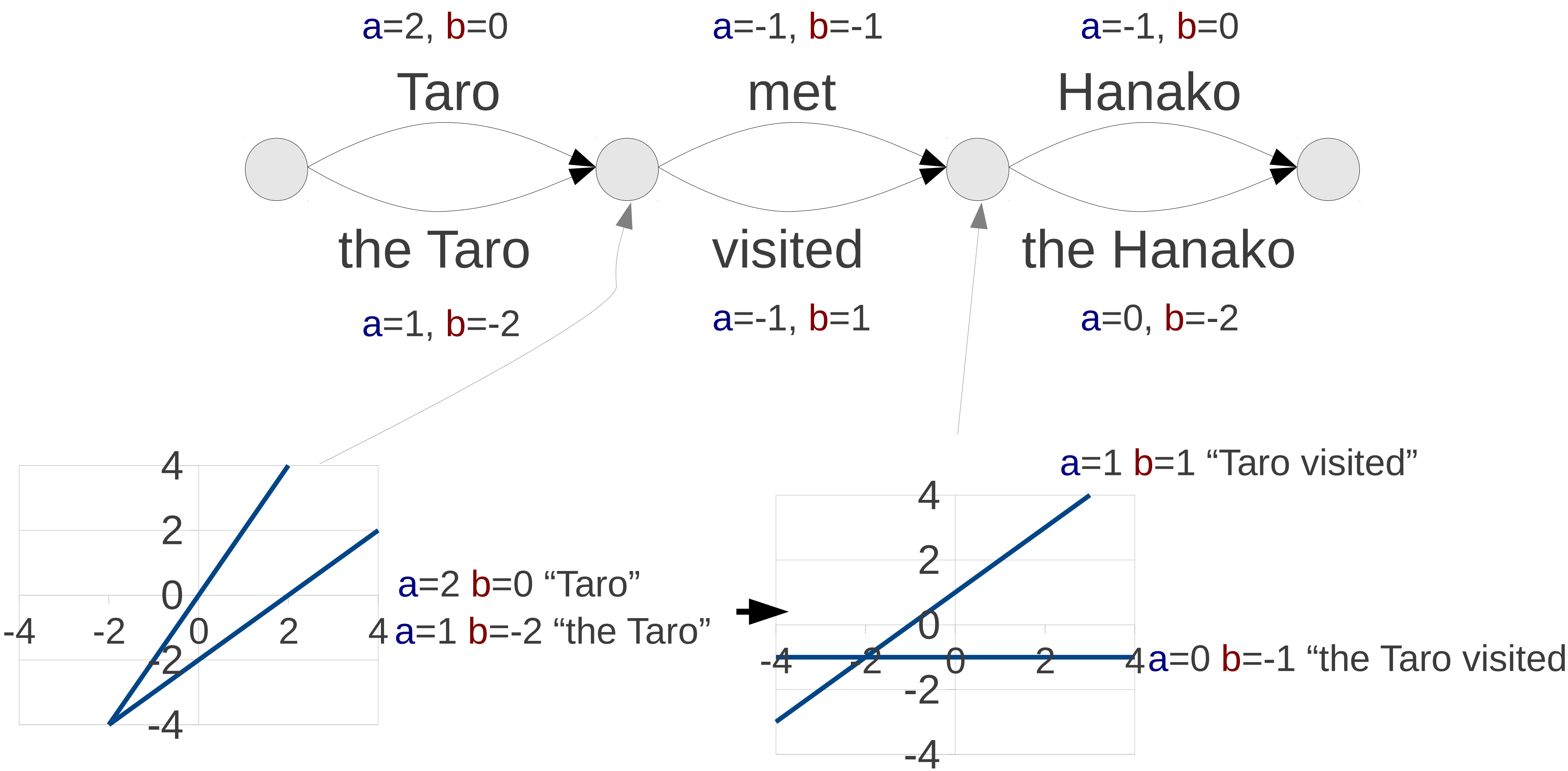
Add Second



$$y = a x + b$$

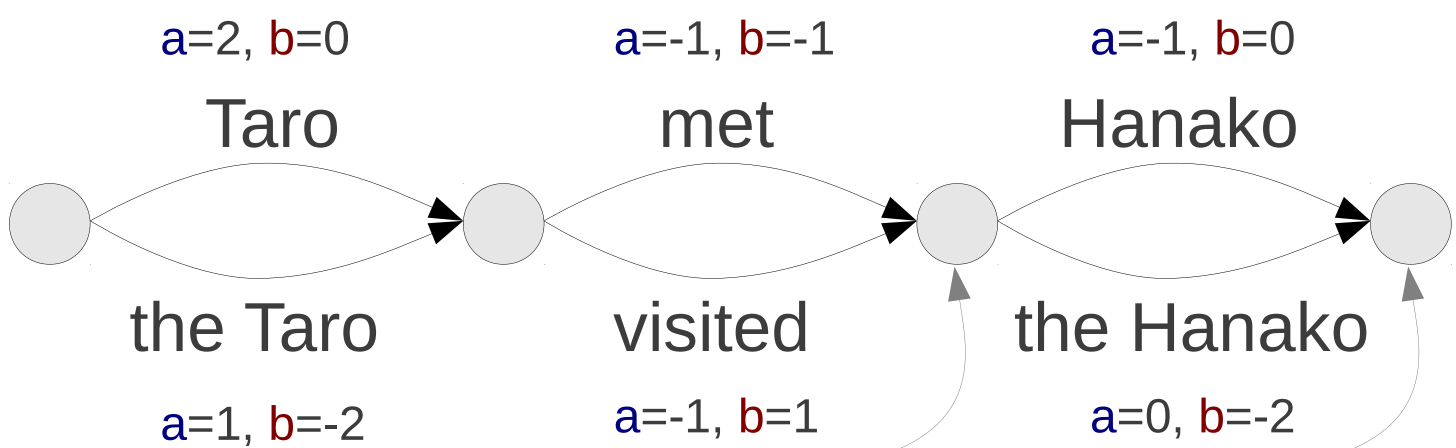
Delete all lines not in upper envelope

Add Second



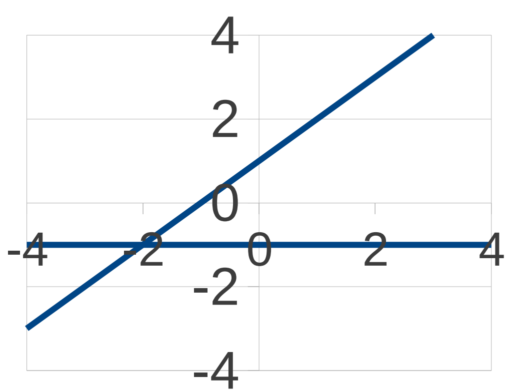
$$y = a\,x + b$$

Add Third

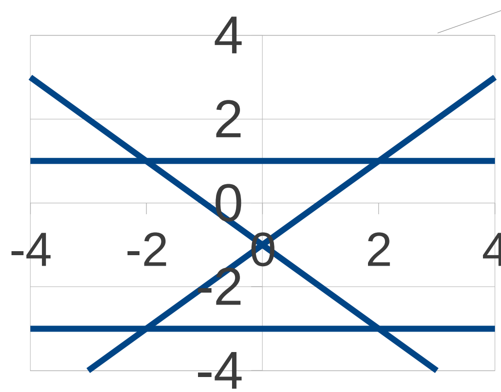


a=1 b=1 “Taro visited”

a=0 b=-1 “the Taro visited”



$$y = a\,x + b$$



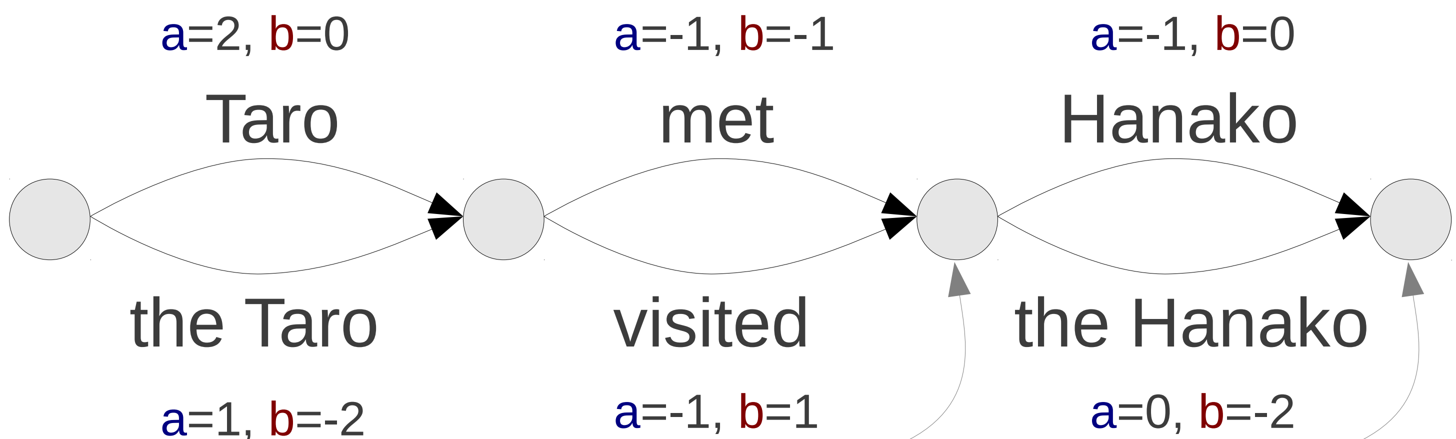
a=1 b=-1 “Taro visited the Hanako”

a=0 b=1 “Taro visited Hanako”

a=0 b=-3 “the Taro visited the Hanako”

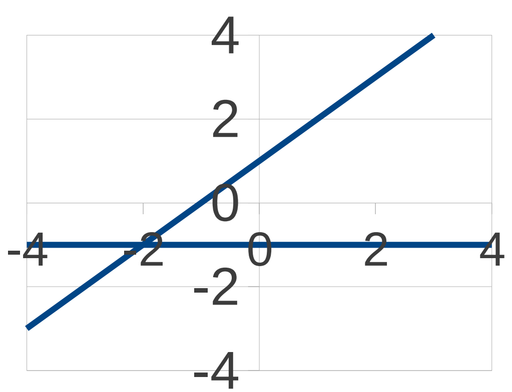
a=-1 b=-1 “the Taro visited Hanako”

Add Third

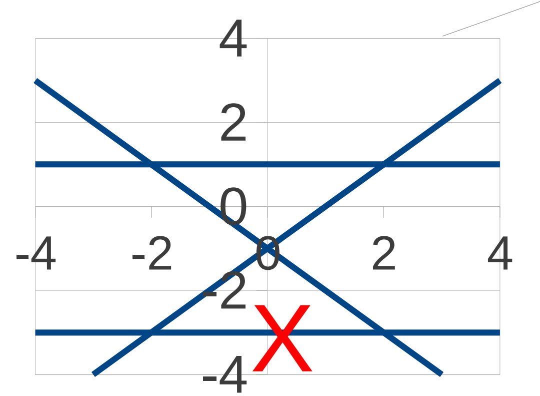


a=1 b=1 “Taro visited”

a=0 b=-1 “the Taro visited”



$$y = a x + b$$



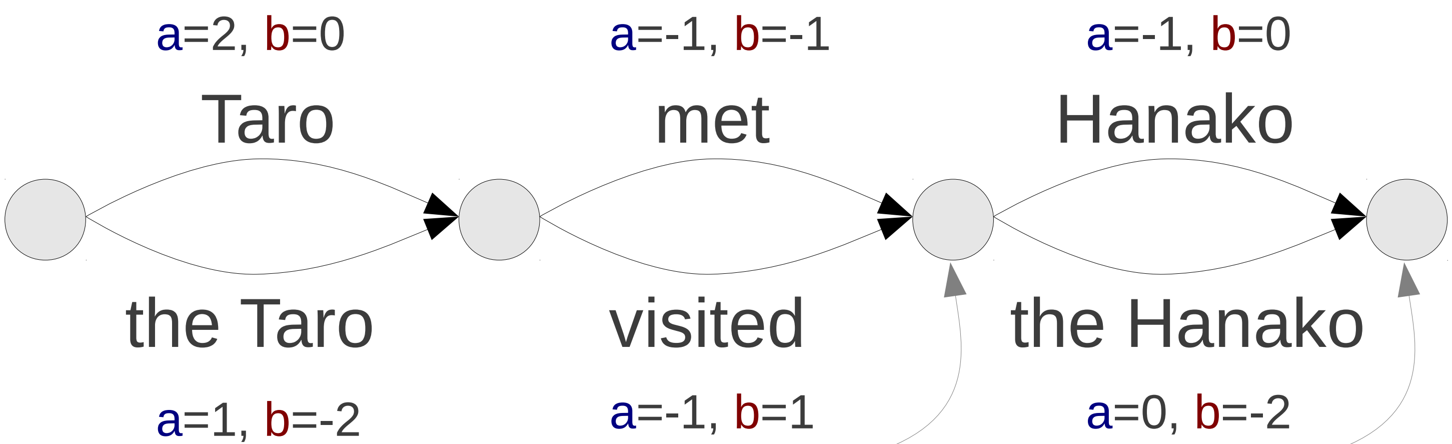
a=1 b=-1 “Taro visited the Hanako”

a=0 b=1 “Taro visited Hanako”

a=0 b=-3 “the Taro visited the Hanako” ✗

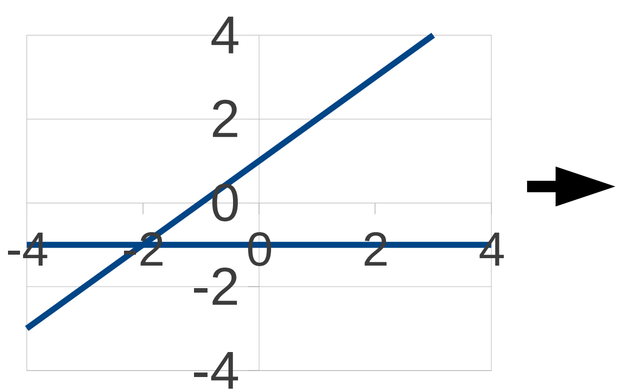
a=-1 b=-1 “the Taro visited Hanako”

Add Third

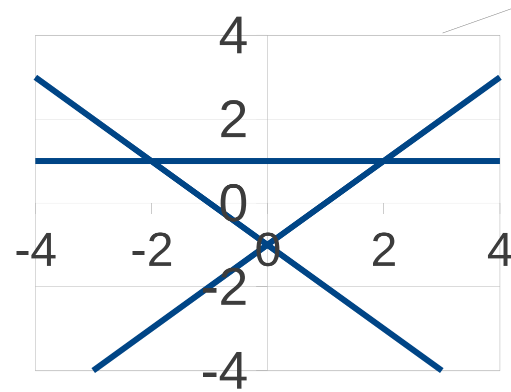


a=1 b=1 “Taro visited”

a=0 b=-1 “the Taro visited”



$$y = a\,x + b$$



a=1 b=-1 “Taro visited the Hanako”

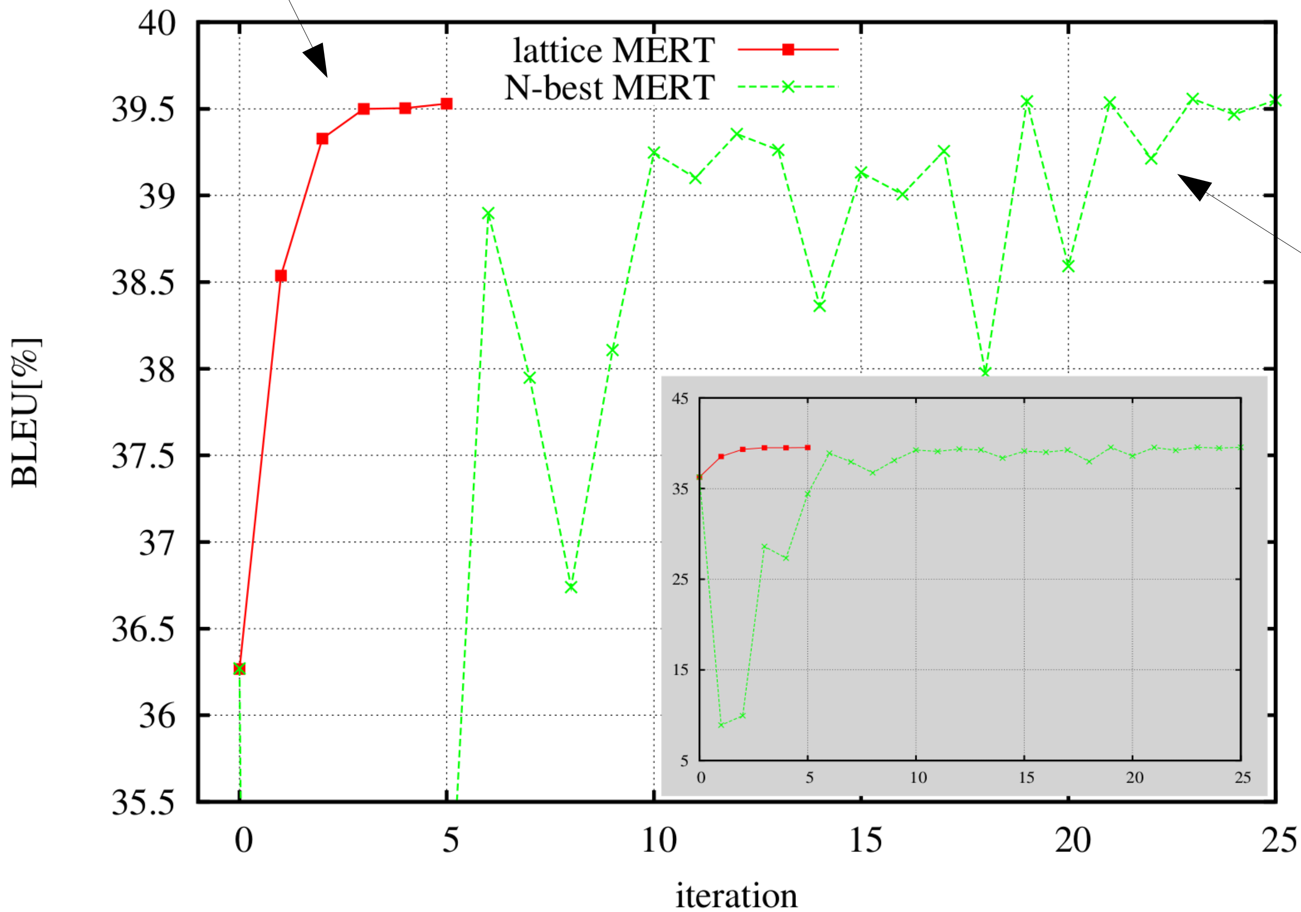
a=0 b=1 “Taro visited Hanako”

a=-1 b=-1 “the Taro visited Hanako”

Improved Stability

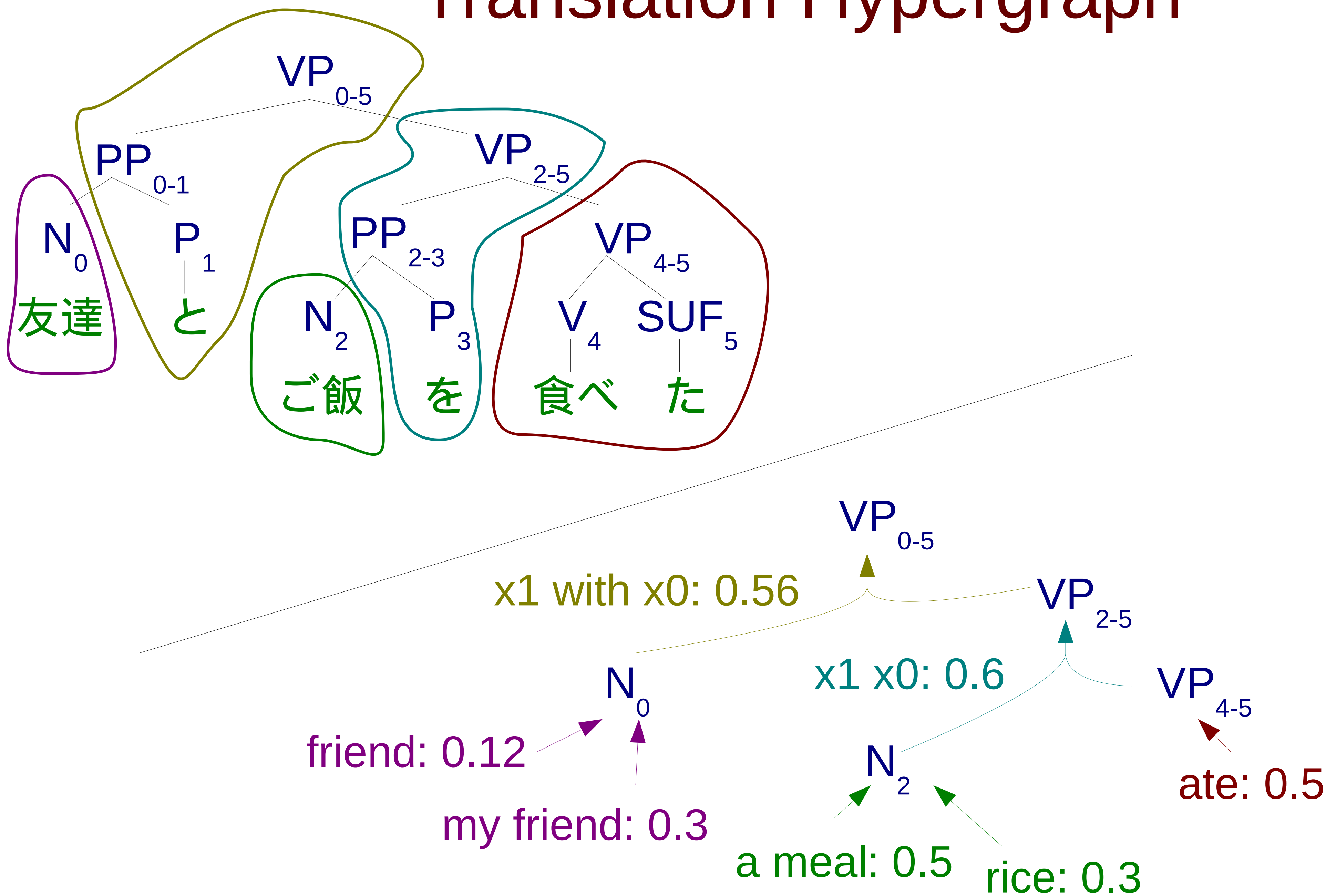
Lattice MERT
in Red

Traditional MERT
in Green



Hypergraph MERT

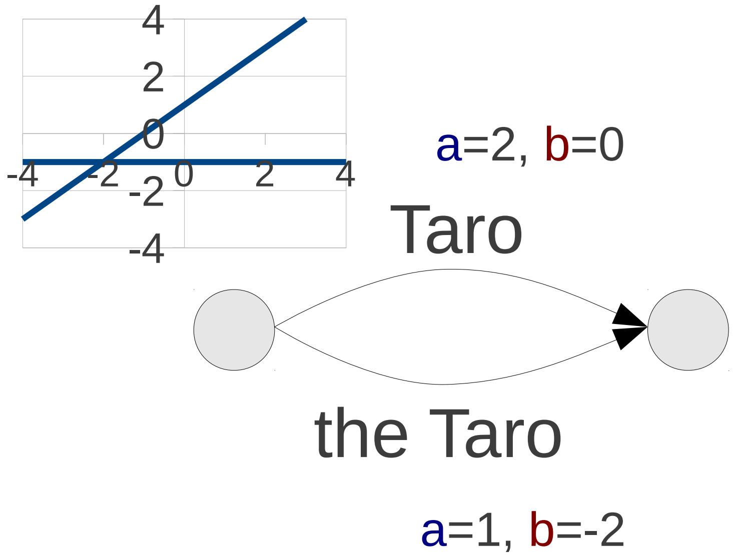
Translation Hypergraph



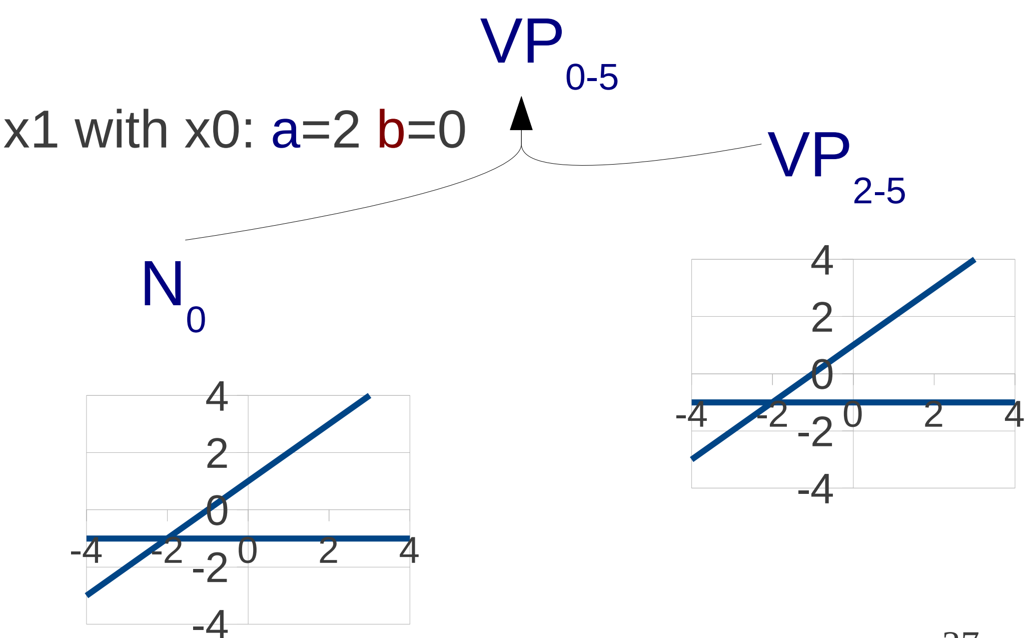
Hypergraph MERT

- Almost exactly the same as lattice MERT

Lattice MERT



Hypergraph MERT



Summary

Summary

- n-best MERT is unstable because of lack of diversity in the n-best list
- This problem can be solved by lattice or hypergraph MERT
- Algorithm finds the upper envelope for each sentence efficiently using dynamic programming