A Word-Complexity Lexicon and **A Neural Readability Ranking Model** for Lexical Simplification

Mounica Maddela and Wei Xu

THE OHIO STATE UNIVERSITY
Department of Computer Science and Engineering

INPUT: *Applesauce is a **puree** made of apples.*

OUTPUT: *Applesauce is a **soft paste**. It is made of apples.*

Text Simplification

INPUT: *Applesauce is a **puree** made of apples.*

OUTPUT: *Applesauce is a **soft paste**. It is made of apples.*

Applications

- Reading assistance for children, non-native speakers and disabled.
- Improve other NLP tasks (MT, summarization …)

Assessing **word complexity** is vital!

INPUT: *Applesauce is a puree made of apples.*

OUTPUT: *Applesauce is a soft paste. It is made of apples.*

Assessing **word complexity** is vital!

INPUT: *Applesauce is a **puree** made of apples.*

OUTPUT: *Applesauce is a soft paste. It is made of apples.*

Complex Word Identification

Assessing **word complexity** is vital!

INPUT: *Applesauce is a* ***puree*** *made of apples.*

OUTPUT: *Applesauce is a* ***soft paste.*** *It is made of apples.*

liquidized sauce

thick liquid

Complex Word Identification - Substitution Generation

Assessing **word complexity** is vital!

INPUT: *Applesauce is a **puree** made of apples.*

OUTPUT: *Applesauce is a **soft paste**. It is made of apples.*

thick liquid

liquidized sauce

complex

Complex Word Identification - Substitution Generation - Substitution Ranking

A Large Word-complexity Lexicon

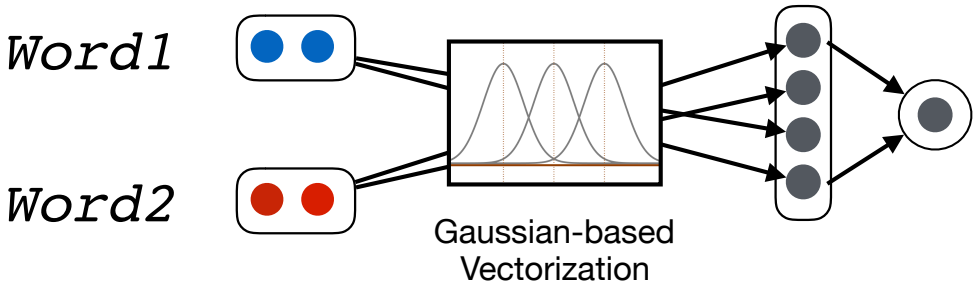
- 15,000 English words w/ human ratings

day	1.0
convenient	2.4
transmitted	3.2
cohort	4.3
assay	5.8

MIN 1 (simple)

MAX 6 (complex)

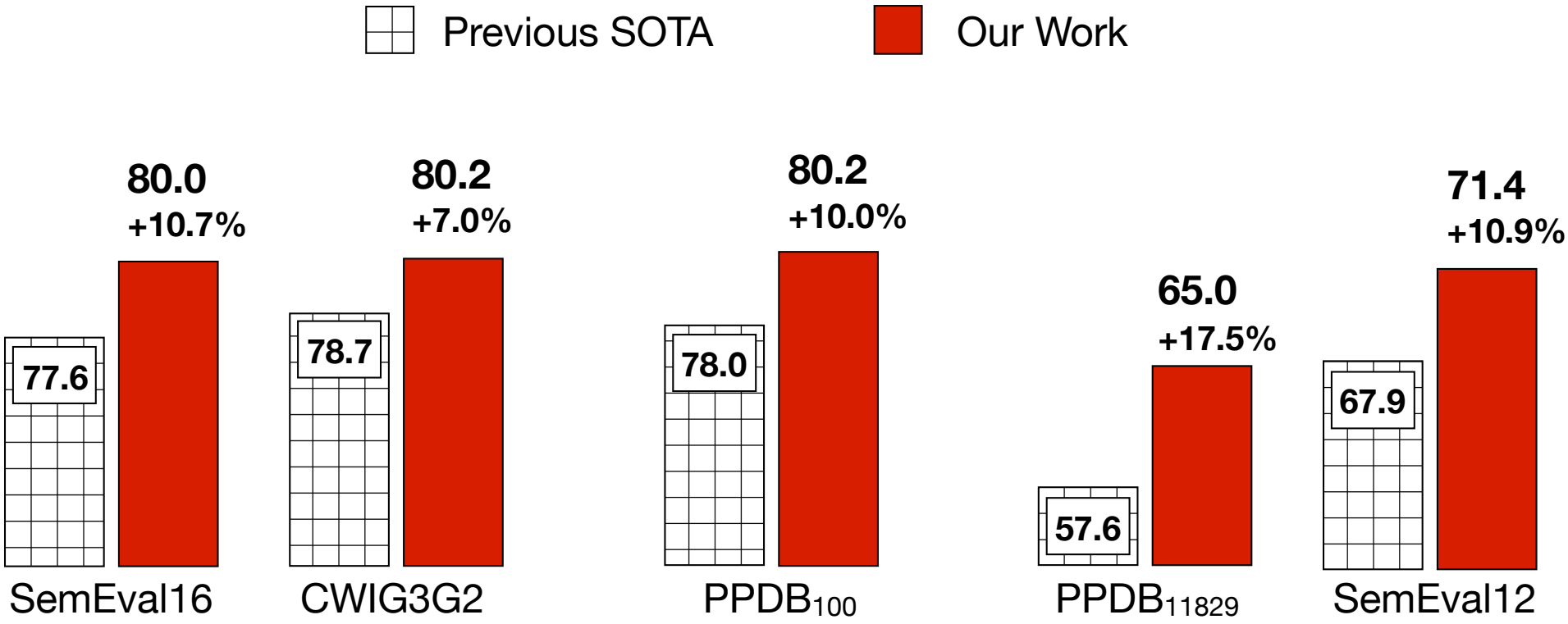
- predict relative complexity for any given words or phrases



R(*Word1* > *Word2*) = 0.91

A Pairwise Neural Ranking Model

- improve the state-of-the-art significantly for all lexical simplification tasks



Complex Word Identification - Substitution Generation - Substitution Ranking

(% is relative error reduction)

Previous Work

Rely on **heuristics and corpus level features** to measure word complexity

- Word length

  (Shardlow 2013, Biran et. al. 2011, and many others)

- Word frequency in corpus

  (Bott et. al. 2011, Kajiwara et. al. 2013, Horn et. al. 2014, and many others)

- Language model probability

  (Glavas & Stajner 2015, Paetzold & Special 2016/17, and many others)

Weakness of Previous Work

Assumption #1: shorter words are simpler → **Wrong! (21% of time*)**

duly > *thoroughly*

pundit > *professional*

alien > *stranger*

* based on 2272 lexical paraphrases sampled from PPDB

Weakness of Previous Work

Assumption #2: more frequent words are simpler → **Wrong! (14% of time*)**

folly > *foolishness*

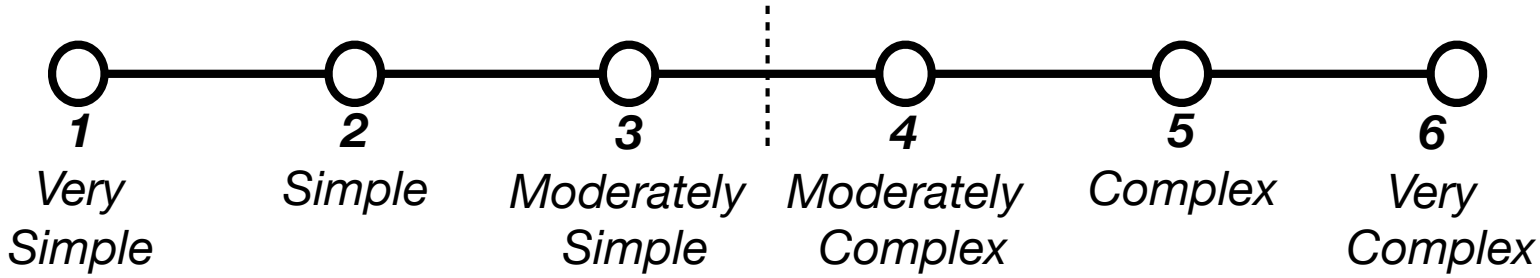
scheme > *outline*

distress > *discomfort*

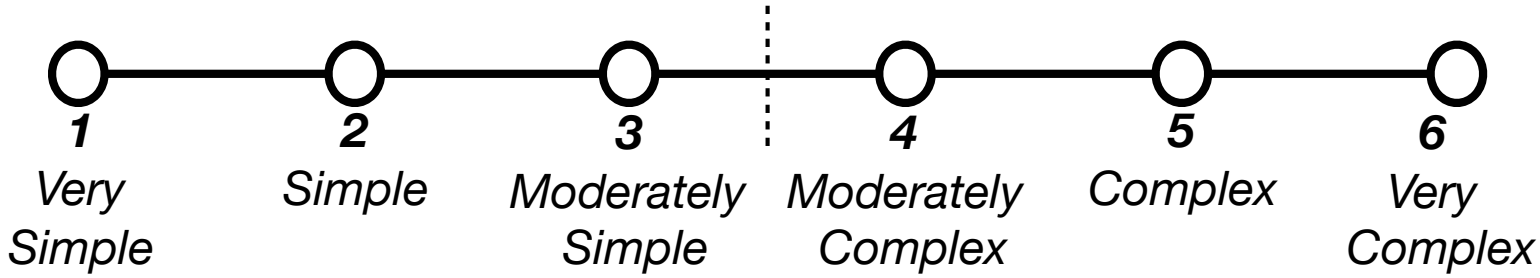
* based on 2272 lexical paraphrases sampled from PPDB

A Large Word-complexity Lexicon

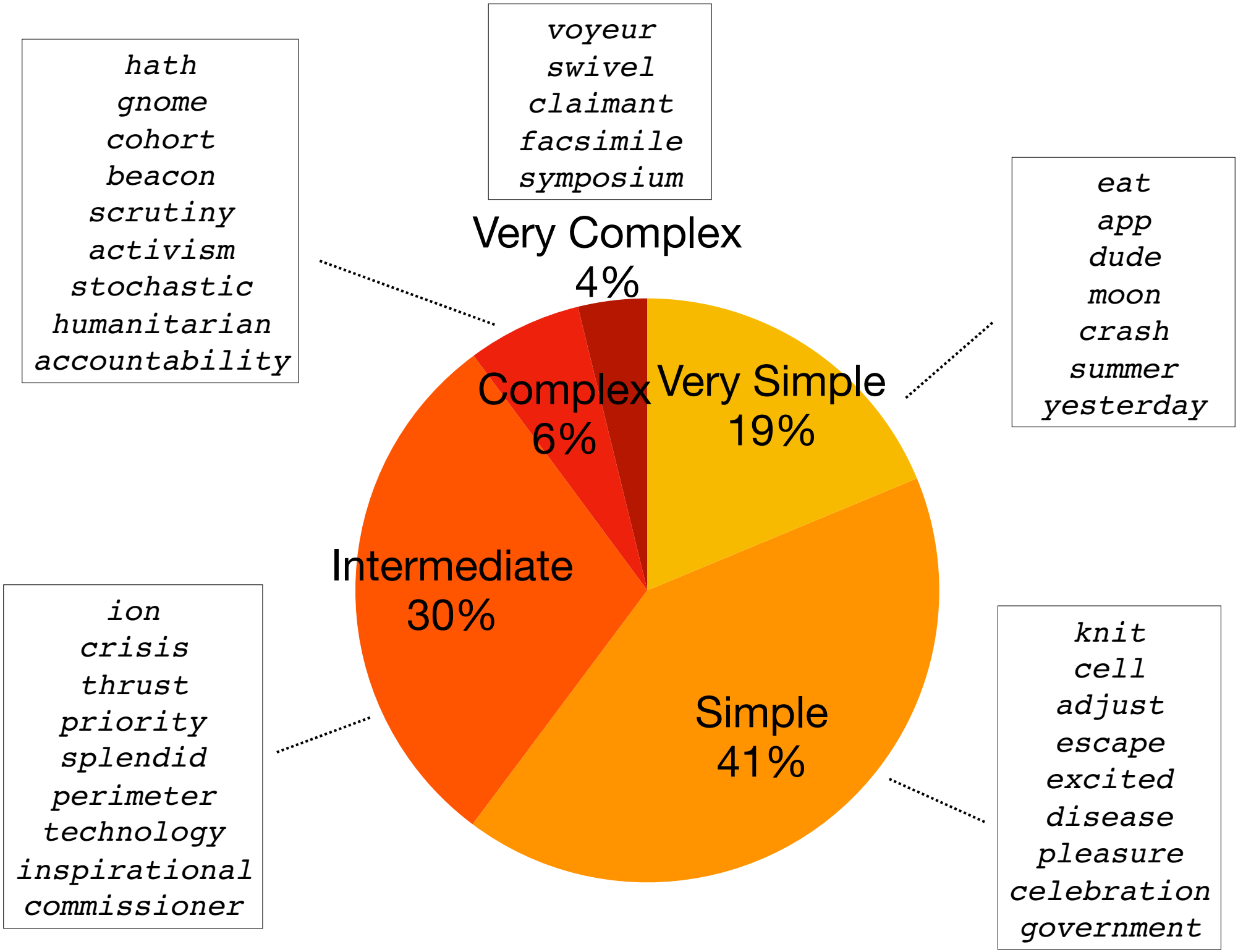
- 15,000 most frequent English words from Google 1T ngram corpus
- Rated on a 6-point Likert scale



- 15,000 most frequent English words from Google 1T ngram corpus
- Rated on a 6-point Likert scale



- 11 annotators (non-native speakers)
- 5 ~ 7 ratings for each word
- 2.5 hours to rate 1000 words


voyeur
swivel
claimant
facsimile
symposium
hath
gnome
cohort
beacon
scrutiny
activism
stochastic
humanitarian
accountability
Very Complex
4%
Complex
6%
Very Simple
19%
eat
app
dude
moon
crash
summer
yesterday
Intermediate
30%
Simple
41%
ion
crisis
thrust
priority
splendid
perimeter
technology
inspirational
commissioner
knit
cell
adjust
escape
excited
disease
pleasure
celebration
government

- Inter-annotator agreement is 0.64 (Pearson correlation)
- One annotator rating vs. mean of the rest

Word	Score	A1	A2	A3	A4	A5
muscles	1.6	2	1	2	2	1
pattern	2.4	2	3	1	1	3
educational	3.2	3	3	3	3	4
cortex	4.2	4	4	4	4	5
assay	5.8	6	6	6	5	6

difference (one vs. rest)

< **0.5** for **47%** of annotations

< **1.0** for **78%** of annotations

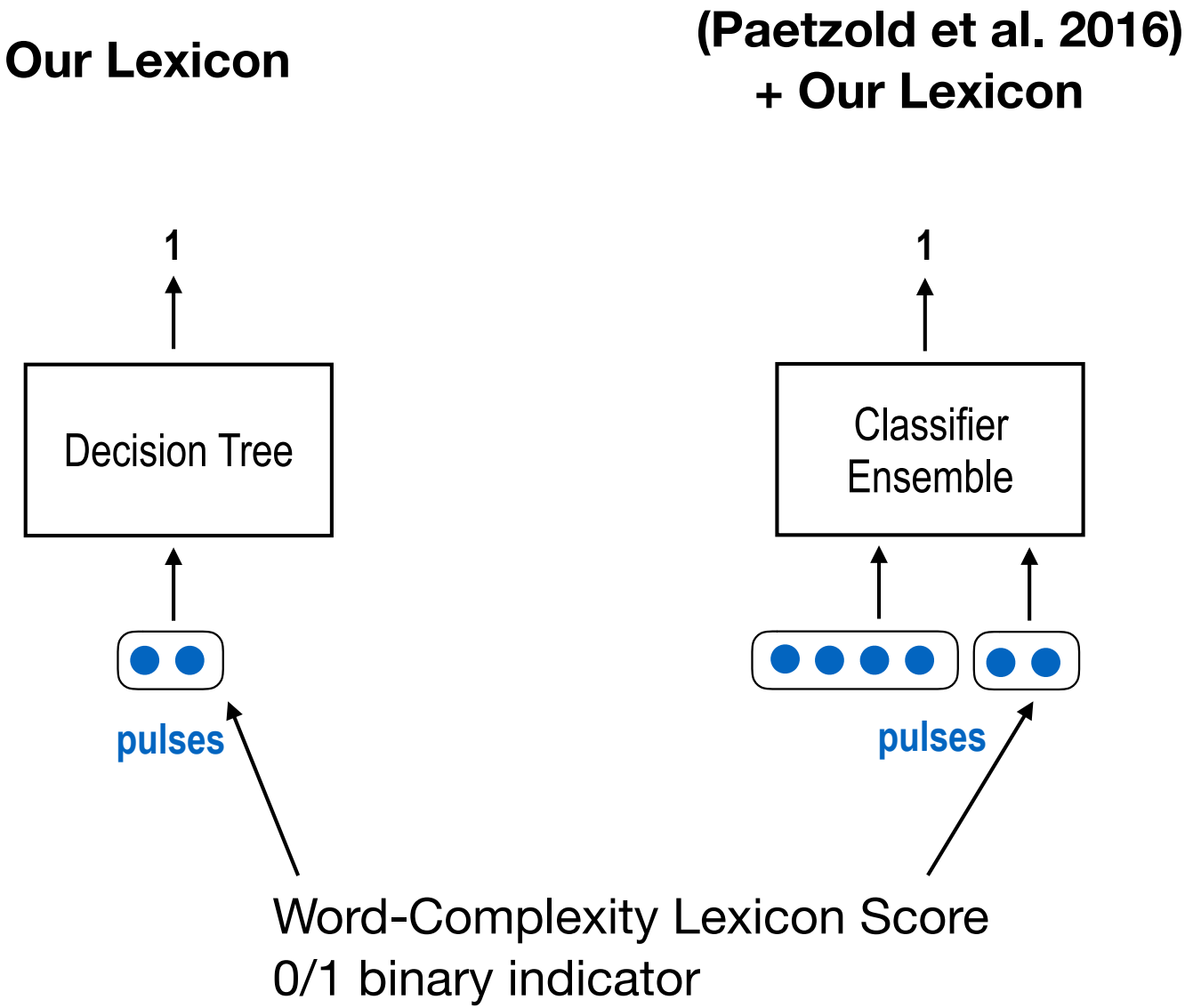
< **1.5** for **93%** of annotations

Evaluation - Complex Word Identification

- Complex Word Identification Shared Task - BEA@NAACL'18
- 34879 sentences from Wikipedia and news articles
- 27299 training, 3328 development, 4252 test instances

Input	*The whale was sensing him with sound* ***pulses*****.**
Output	***[Complex,*** *simple]*

Evaluation


Our Lexicon
(Paetzold et al. 2016)
+ Our Lexicon
1
1
Decision Tree
Classifier
Ensemble
pulses
pulses
Word-Complexity Lexicon Score
0/1 binary indicator

Evaluation

- Complex Word Identification Shared Task 2018
- 27299 training, 3328 development, 4252 test instances

	F-score	**Accuracy**
Senses	62.3	54.1
SimpleWiki Frequency	63.3	61.6
Length	65.9	67.7
(Paetzold et al. 2016)	73.8	78.7

Evaluation

- Complex Word Identification Shared Task 2018
- 27299 training, 3328 development, 4252 test instances

	F-score	**Accuracy**
Senses	62.3	54.1
SimpleWiki Frequency	63.3	61.6
Length	65.9	67.7
(Paetzold et al. 2016)	73.8	78.7
Our Lexicon	67.5	69.8

Evaluation

- Complex Word Identification Shared Task 2018
- 27299 training, 3328 development, 4252 test instances

	F-score	Accuracy
Senses	62.3	54.1
SimpleWiki Frequency	63.3	61.6
Length	65.9	67.7
(Paetzold et al. 2016)	73.8	78.7
Our Lexicon	67.5	69.8
(Paetzold et al. 2016) + Our Lexicon	*74.8	*80.2

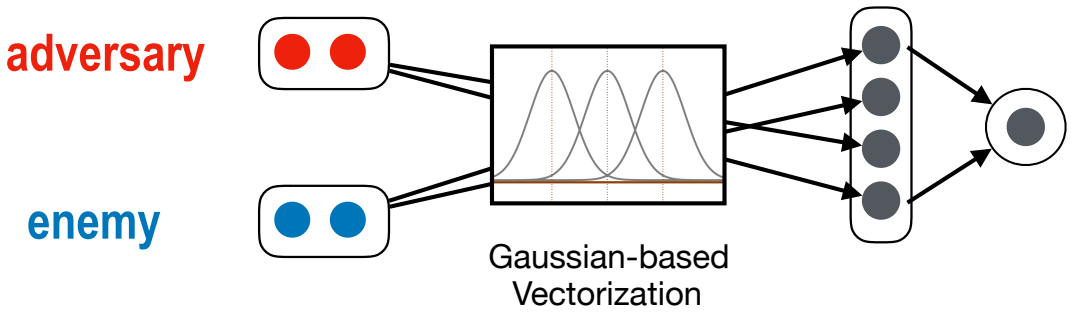
+1.0

+1.5

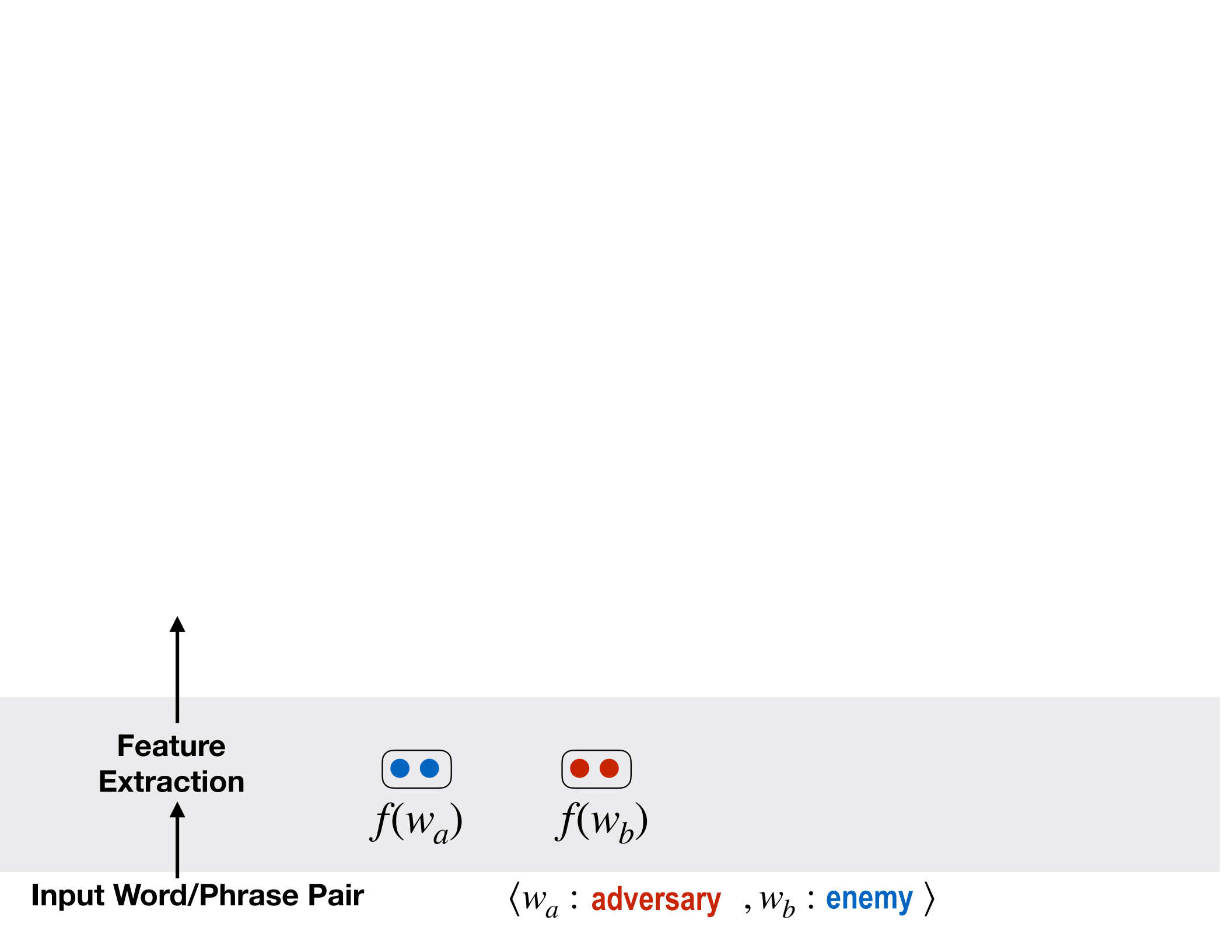
* statistically significant ($p < 0.01$) based on the paired bootstrap test

A Pairwise Neural Ranking Model

- predict relative complexity for any given words or phrases

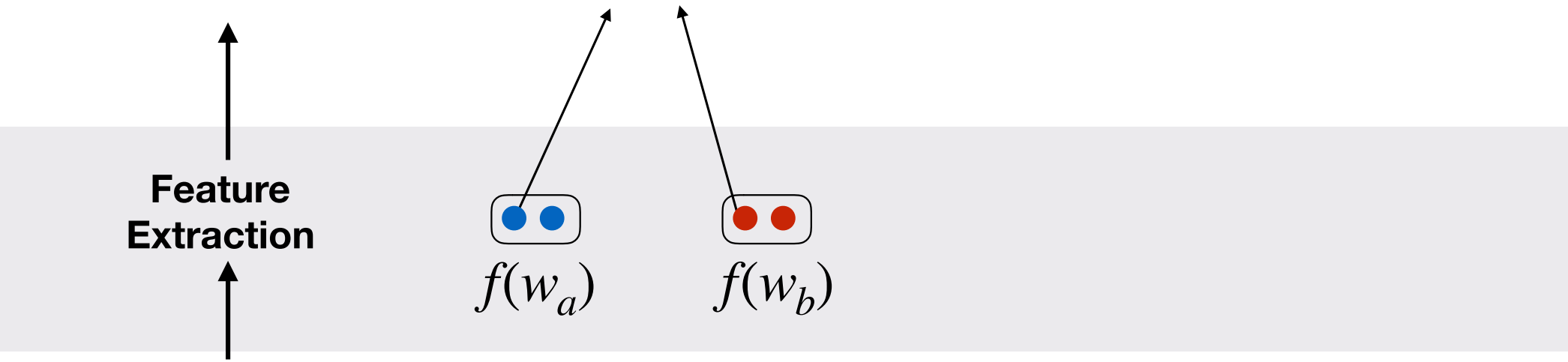


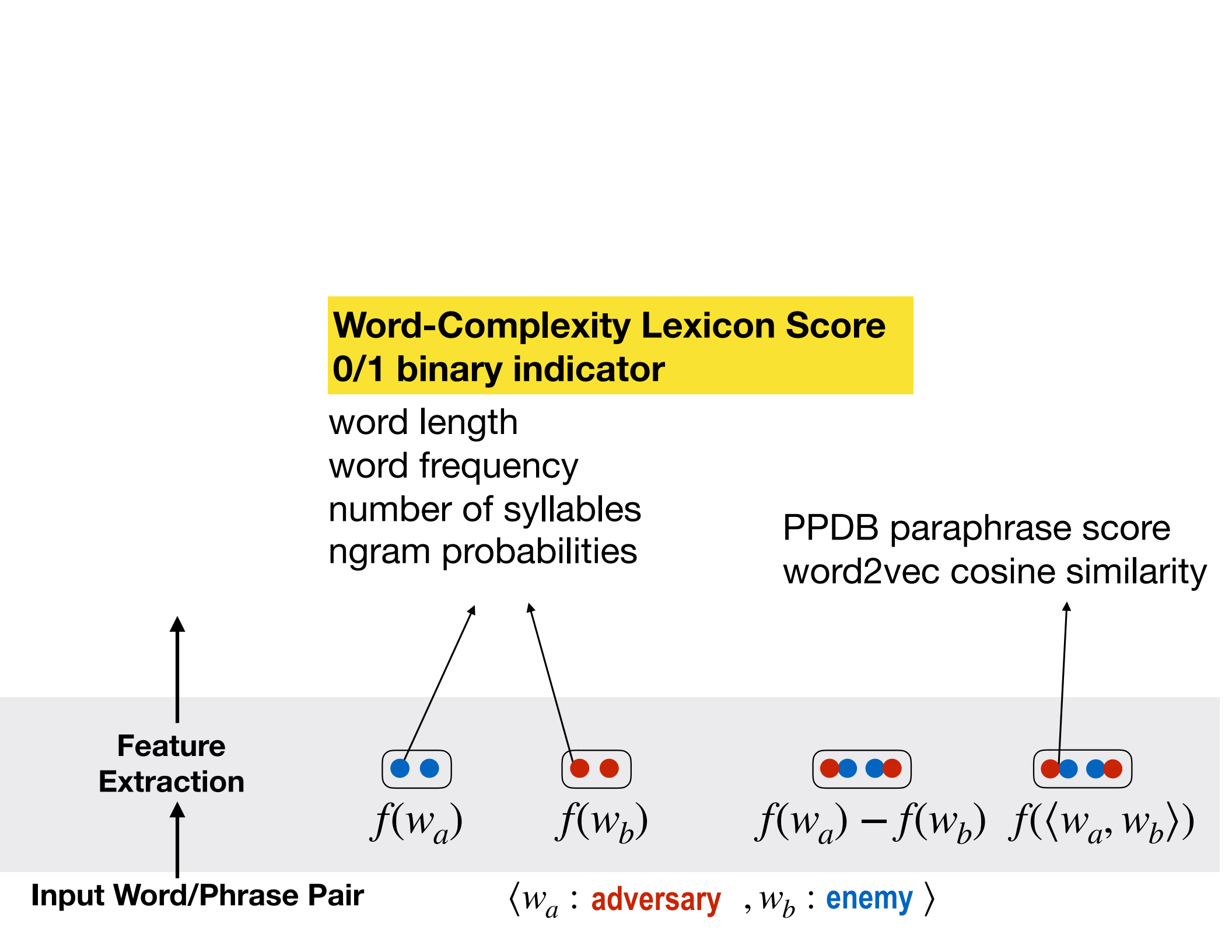
R(adversary > enemy) = 0.91

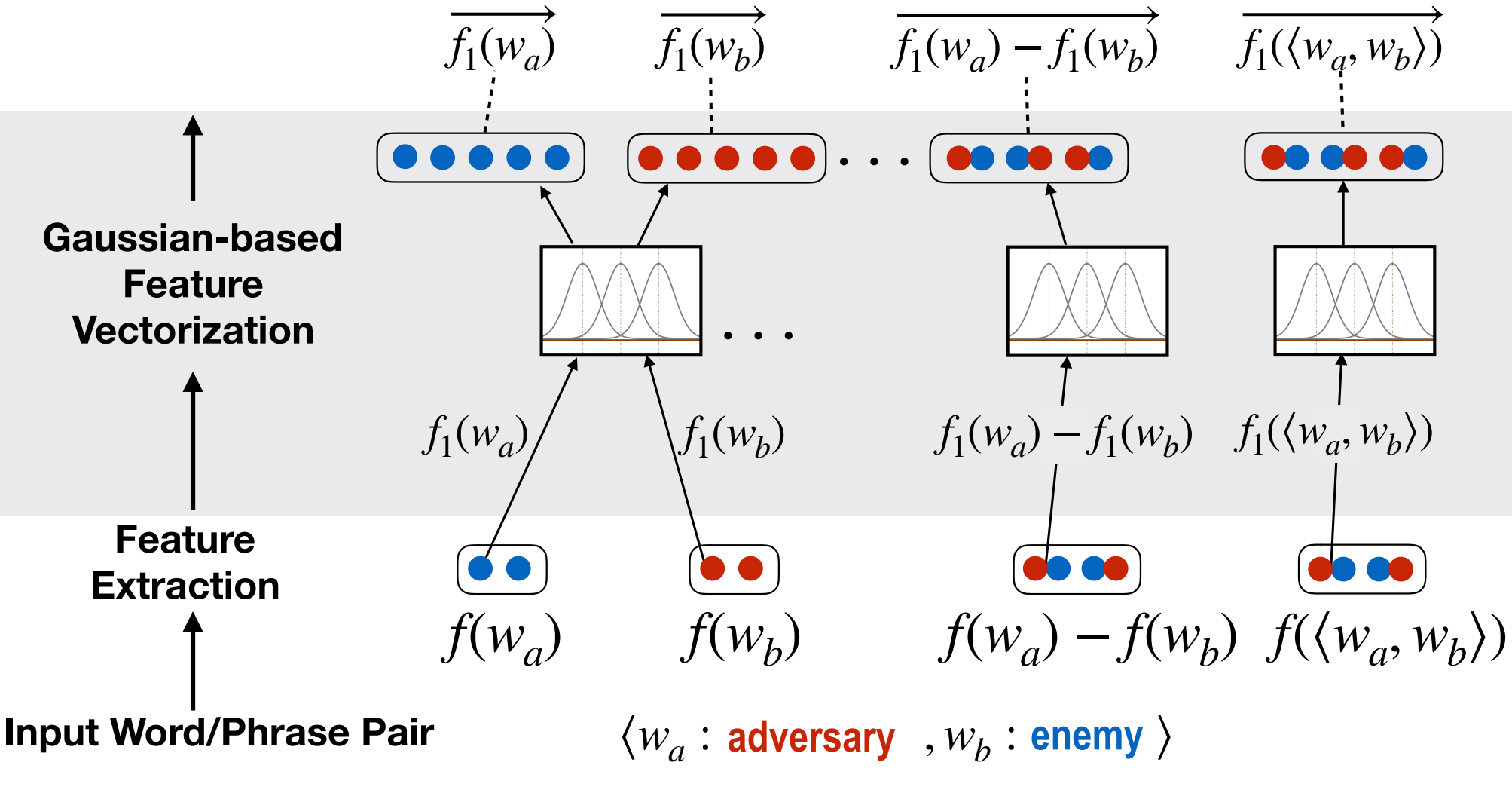

Feature Extraction
$f(w_a)$
$f(w_b)$
Input Word/Phrase Pair
$\langle w_a$: adversary , w_b : enemy $\rangle$

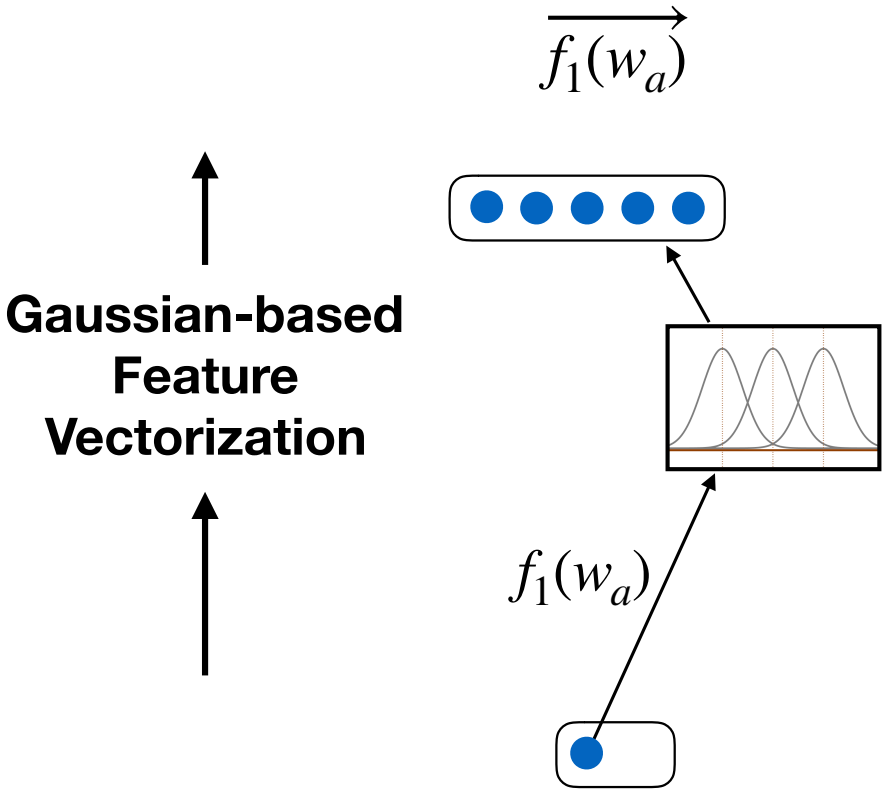
Word-Complexity Lexicon Score
0/1 binary indicator

word length
word frequency
number of syllables
ngram probabilities



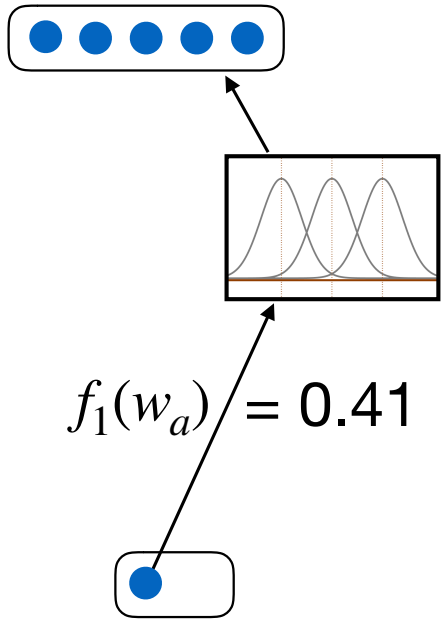

Word-Complexity Lexicon Score
0/1 binary indicator
word length
word frequency
number of syllables
ngram probabilities
PPDB paraphrase score
word2vec cosine similarity
Feature Extraction
$f(w_a)$
$f(w_b)$
$f(w_a) - f(w_b)$
$f(\langle w_a, w_b \rangle)$
Input Word/Phrase Pair
$\langle w_a$: adversary , w_b : enemy $\rangle$


$\overrightarrow{f_1(w_a)}$
$\overrightarrow{f_1(w_b)}$
$\overrightarrow{f_1(w_a) - f_1(w_b)}$
$\overrightarrow{f_1(\langle w_a, w_b \rangle)}$
Gaussian-based Feature Vectorization
$f_1(w_a)$
$f_1(w_b)$
$f_1(w_a) - f_1(w_b)$
$f_1(\langle w_a, w_b \rangle)$
Feature Extraction
$f(w_a)$
$f(w_b)$
$f(w_a) - f(w_b)$
$f(\langle w_a, w_b \rangle)$
Input Word/Phrase Pair
$\langle w_a$: adversary , w_b : enemy $\rangle$


$\overrightarrow{f_1(w_a)}$
Gaussian-based
Feature
Vectorization
$f_1(w_a)$

$\overrightarrow{f_1(w_a)}$ = [~0.0, **0.44**, **0.54**, ~0.02, ~0.0]

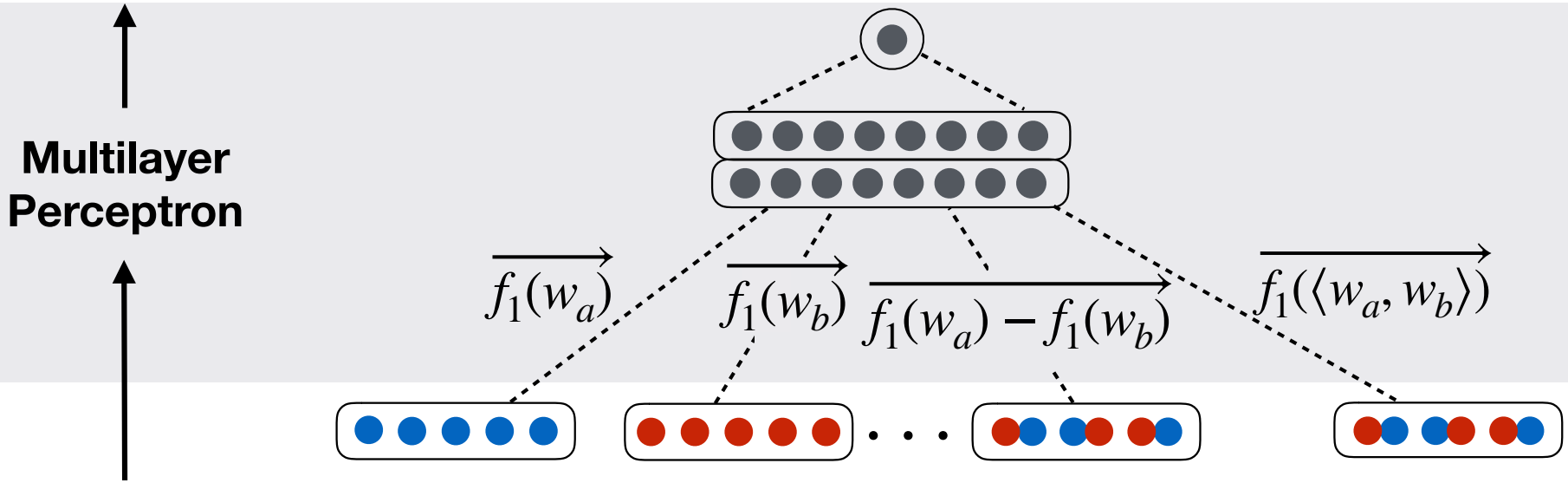
Gaussian-based Feature Vectorization



$f_1(w_a) = 0.41$

$$d_j(f) = e^{-\frac{(f-\mu_j)^2}{2\sigma^2}}$$

0.91 = R(more_complex | **adversary** - **enemy**)



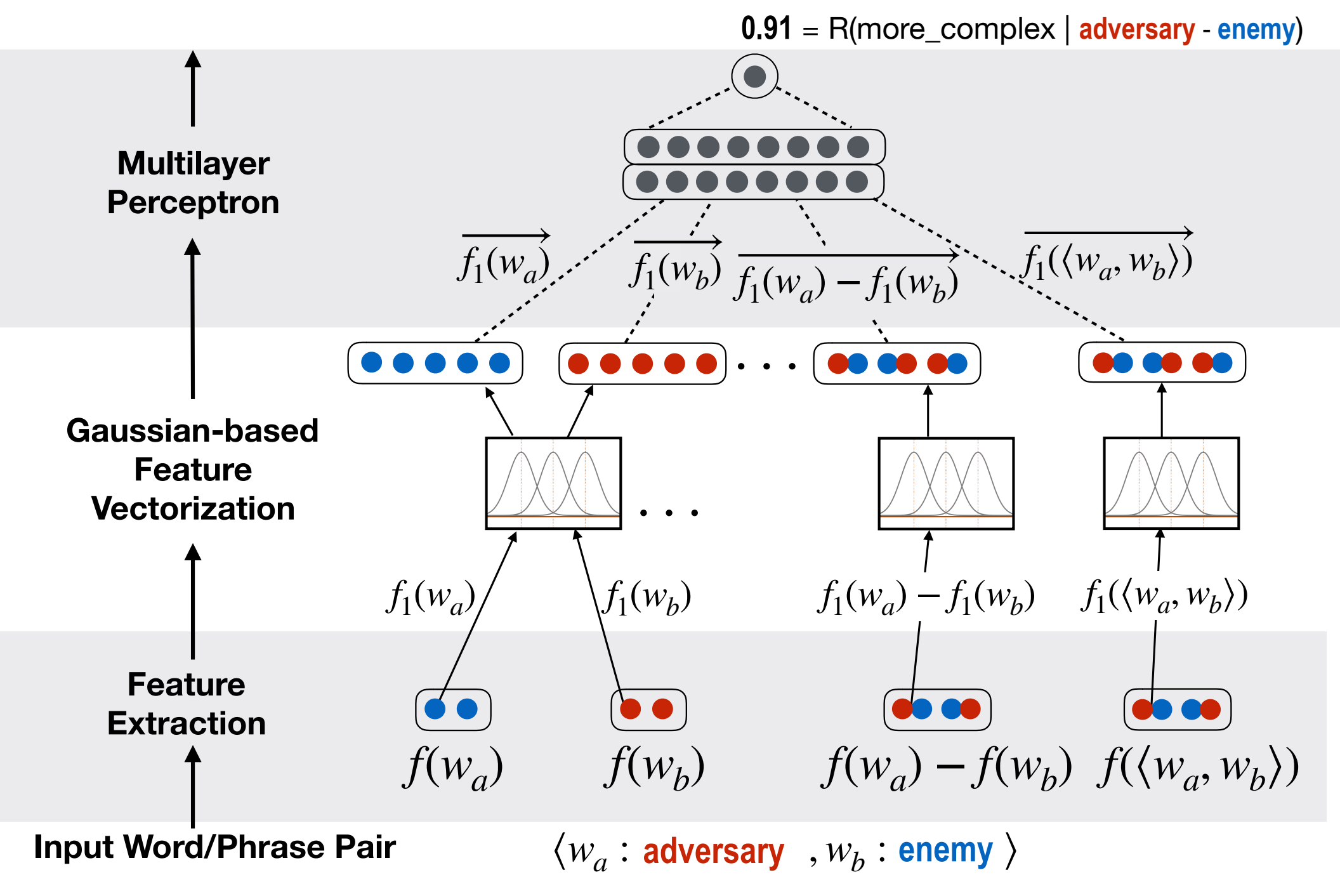
R > 0 $\Rightarrow$ w_a is more complex than w_b

R < 0 $\Rightarrow$ w_a is simpler than w_b

| R | indicates complexity difference

$\langle w_a$: **adversary** , w_b : **enemy** $\rangle$

Neural Readability Ranking Model



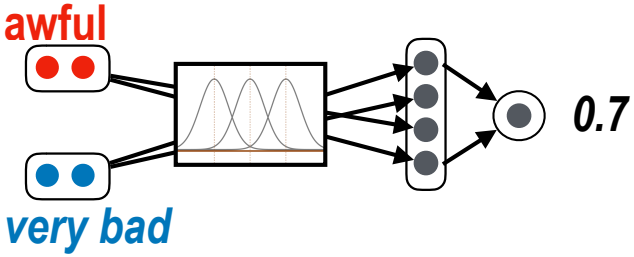
Evaluation

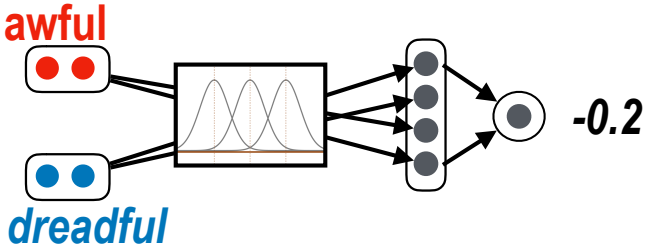
- English Lexical Simplification Shared Task - SemEval 2012
- 300 training sentences, 1710 test sentences

Input	*There were also pieces that would have been* ***terrible*** *in any environment.* *{awful, very bad, dreadful}*
Gold truth	***very bad < awful < dreadful***

Candidates: ***awful, very bad, dreadful***

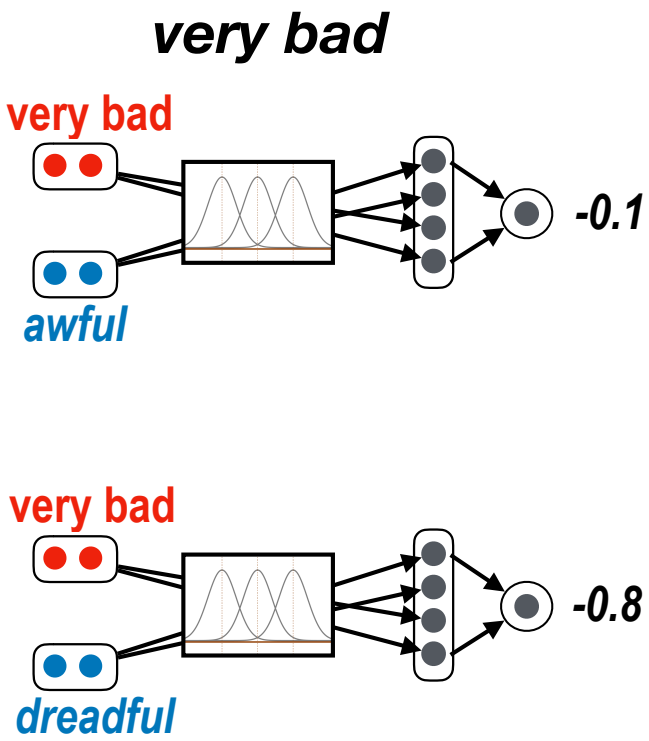
awful





0.7 - 0.2 = 0.5

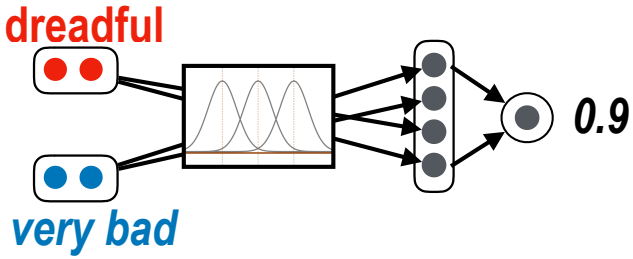
Candidates: ***awful, very bad, dreadful***

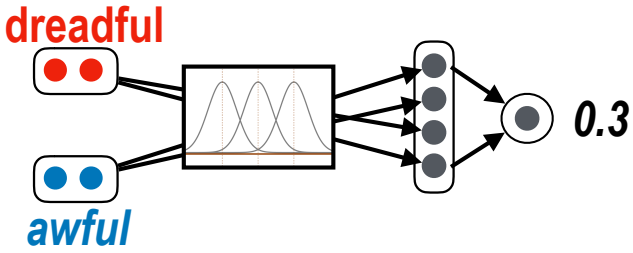


-0.1 - 0.8 = -0.9

Candidates: ***awful, very bad, dreadful***

dreadful

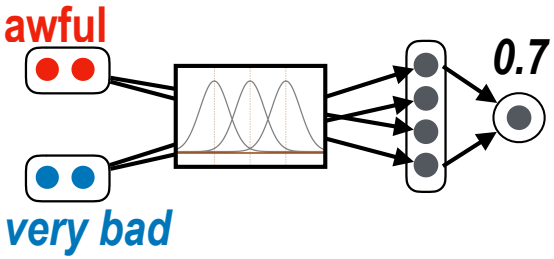


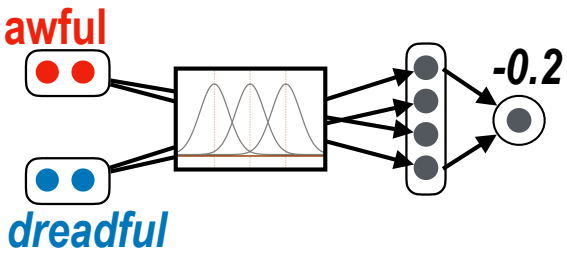


0.9 + 0.3 = 1.2

Candidates: ***awful, very bad, dreadful***

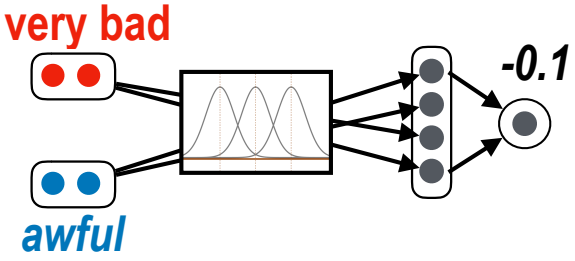
awful

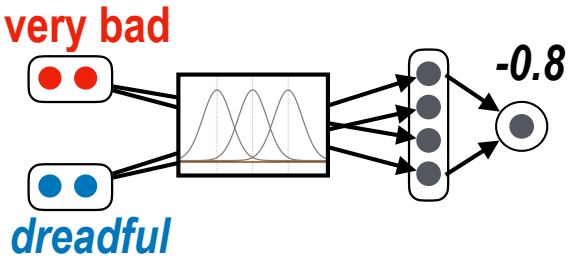




0.7 - 0.2 = 0.5

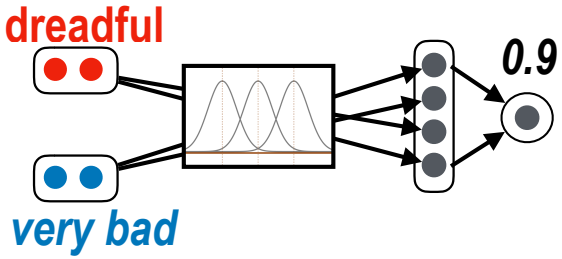
very bad

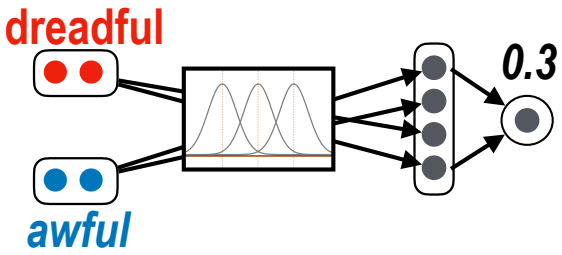




-0.1 - 0.8 = -0.9

dreadful





0.9 - 0.3 = 1.2

very bad < awful < dreadful

- English Lexical Simplification Shared Task - SemEval 2012
- 300 training sentences, 1710 test sentences

		Precision@1	Pearson
heuristics	(Biran et al. 2011)	51.3	0.505
SVM	(Jauhar & Specia 2012)	60.2	0.575
heuristics	(Kajiwara et al. 2013)	60.4	0.649
SVM	(Horn et al. 2014)	63.9	0.673
heuristics	(Glavaš & Štajner 2015)	63.2	0.644
SVM	(Paetzold & Specia 2015)	65.3	0.677
neural	(Paetzold & Specia 2017)	65.6	0.679
neural	Our Model + Lexicon + Gaussian	67.3*	0.714*

+0.2 +0.002 +1.7 +0.035

* statistically significant ($p < 0.05$) based on the paired bootstrap test

- English Lexical Simplification Shared Task - SemEval 2012
- 300 training sentences, 1710 test sentences

		Precision@1	Pearson
heuristics	(Biran et al. 2011)	51.3	0.505
SVM	(Jauhar & Specia 2012)	60.2	0.575
heuristics	(Kajiwara et al. 2013)	60.4	0.649
SVM	(Horn et al. 2014)	63.9	0.673
heuristics	(Glavaš & Štajner 2015)	63.2	0.644
SVM	(Paetzold & Specia 2015)	65.3	0.677
neural	(Paetzold & Specia 2017)	65.6	0.679
neural	Our Model + Gaussian	66.6	0.702*
neural	Our Model + Lexicon + Gaussian	67.3*	0.714*

+0.2 +0.002 +1.7 +0.035

* statistically significant ($p < 0.05$) based on the paired bootstrap test

- English Lexical Simplification Shared Task - SemEval 2012
- 300 training sentences, 1710 test sentences

		Precision@1	Pearson
heuristics	(Biran et al. 2011)	51.3	0.505
SVM	(Jauhar & Specia 2012)	60.2	0.575
heuristics	(Kajiwara et al. 2013)	60.4	0.649
SVM	(Horn et al. 2014)	63.9	0.673
heuristics	(Glavaš & Štajner 2015)	63.2	0.644
SVM	(Paetzold & Specia 2015)	65.3	0.677
neural	(Paetzold & Specia 2017)	65.6	0.679
neural	Our Model	65.4	0.682
neural	Our Model + Gaussian	66.6	0.702*
neural	Our Model + Lexicon + Gaussian	67.3*	0.714*

+0.2 +0.002 +1.7 +0.035

* statistically significant ($p < 0.05$) based on the paired bootstrap test

Evaluation - Correct Examples

- Our Model predicts the correct output

Input	*The **concept** of a "picture element" dates to the earliest days of television.*
(Paetzold & Specia 2017)	***theory, thought, idea***
Our Model + Our Lexicon	***idea, thought, theory***
Gold truth	***idea, thought, theory***

- Our Model handles phrases better than previous SOTA.

Input	*There were also pieces that would have been **terrible** in any environment.*
(Paetzold & Specia 2017)	***awful, very bad, dreadful***
Our Model + Our Lexicon	***very bad, awful, dreadful***
Gold truth	***very bad, awful, dreadful***

Evaluation - Error Analysis

Input	*The colonies of one **strain** appeared smooth.*
(Paetzold & Specia 2017)	***sort, type, breed, variety***
Our Model + Our Lexicon	***type, sort, breed, variety***
Gold truth	***type, sort, variety, breed***

Input	*No damage or **casualties** were reported.*
(Paetzold & Specia 2017)	***injuries, accidents, deaths, fatalities***
Our Model + Our Lexicon	***injuries, deaths, accidents, fatalities***
Gold truth	***deaths, injuries, accidents, fatalities***

SimplePPDB++

- 14.1 million paraphrase rules w/ improved complexity ranking scores

Paraphrase Rule		Score
	→ *self-supporting*	0.93
self-reliant	→ *self-sufficient*	0.48
	→ *self-sustainable*	-0.60
	→ *possible*	0.94
viable	→ *realistic*	0.15
	→ *plausible*	-0.91
	→ *in-depth review*	0.89
detailed assessement	→ *careful examination*	0.28
	→ *comprehensive evaluation*	-0.87

Thanks

- **Word-Complexity Lexicon** & **SimplePPDB++** are available!

day	1.0
convenient	2.4
transmitted	3.2
cohort	4.3
assay	5.8

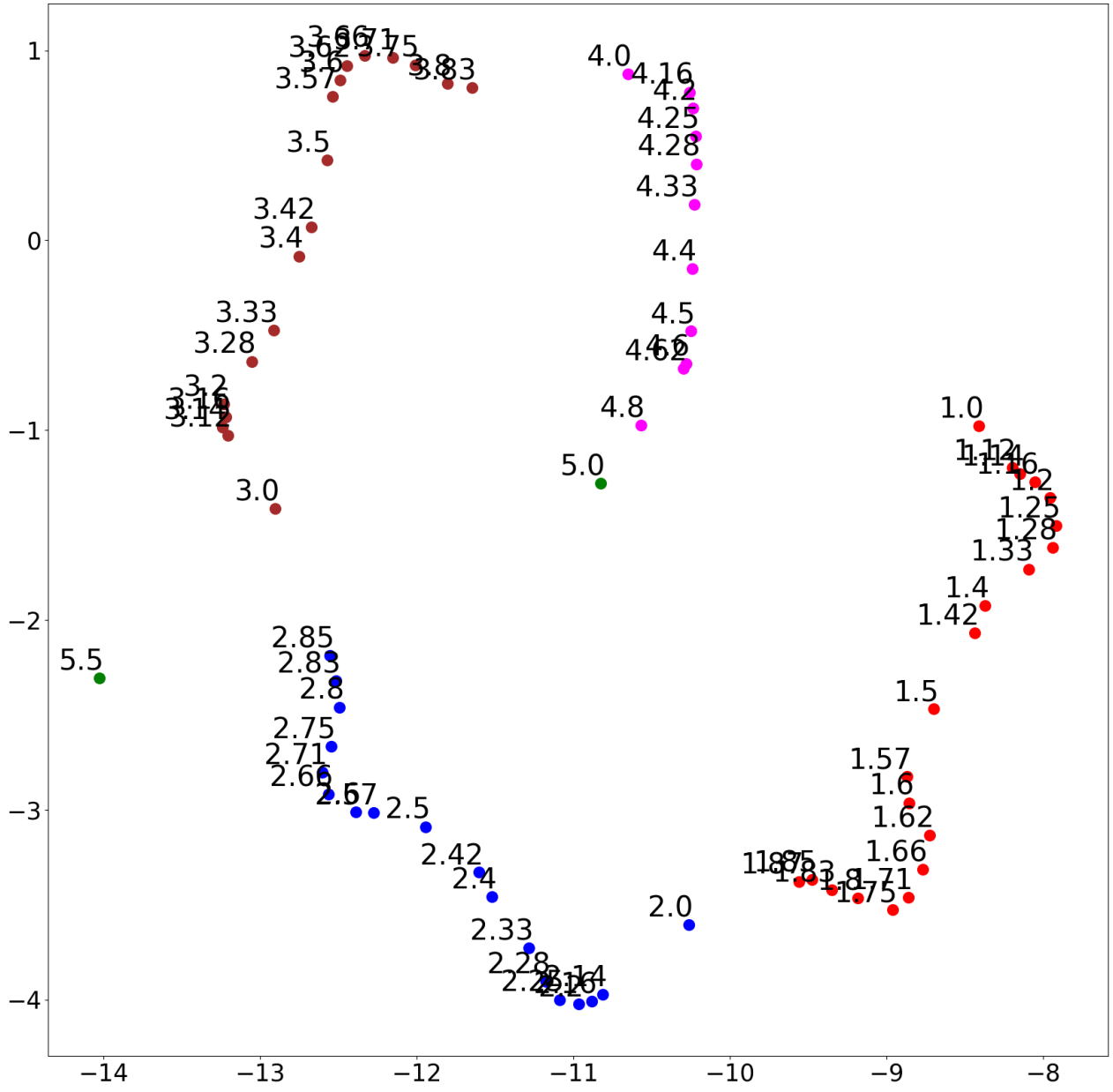
MIN 1 (simple)

MAX 6 (complex)

- PyTorch Code for the **Neural Ranking model** is also available!

  https://github.com/mounicam/lexical_simplification

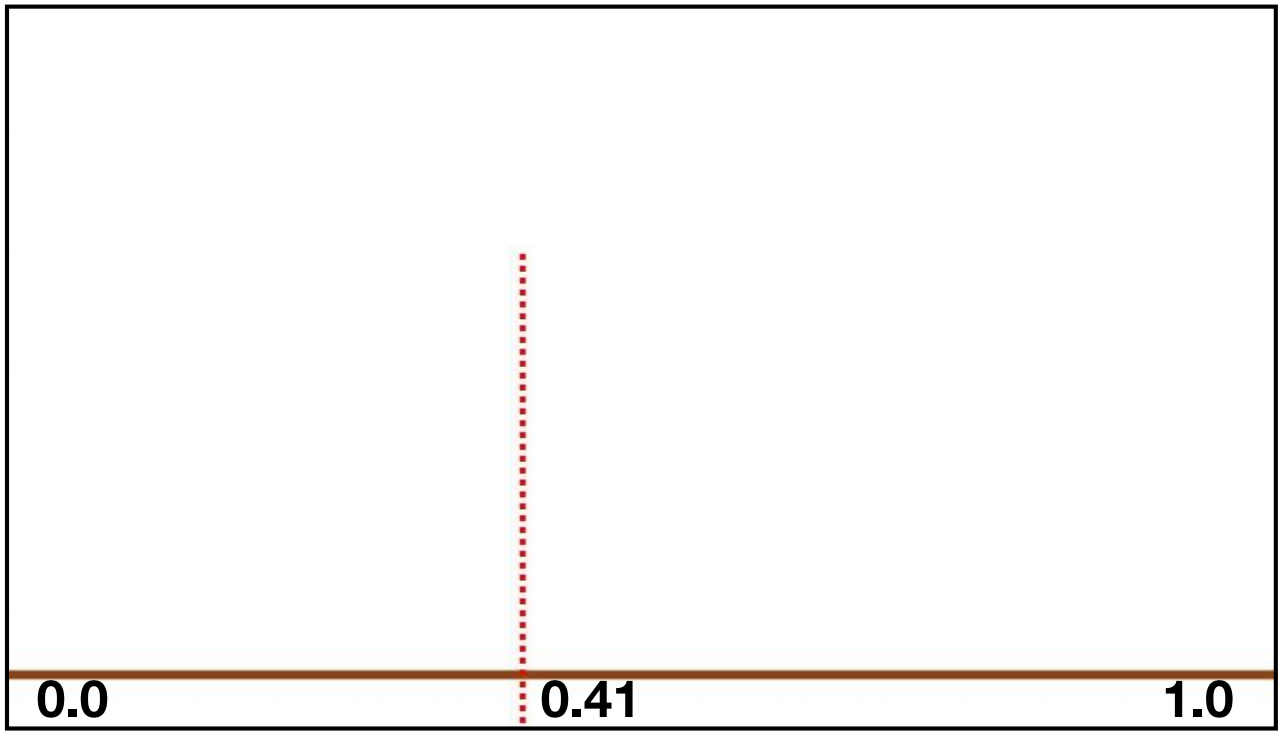
- Contacts: Mounica Maddela (maddela.4@osu.edu)



t-SNE visualization of the complexity scores, ranging between 1.0 and 6.0

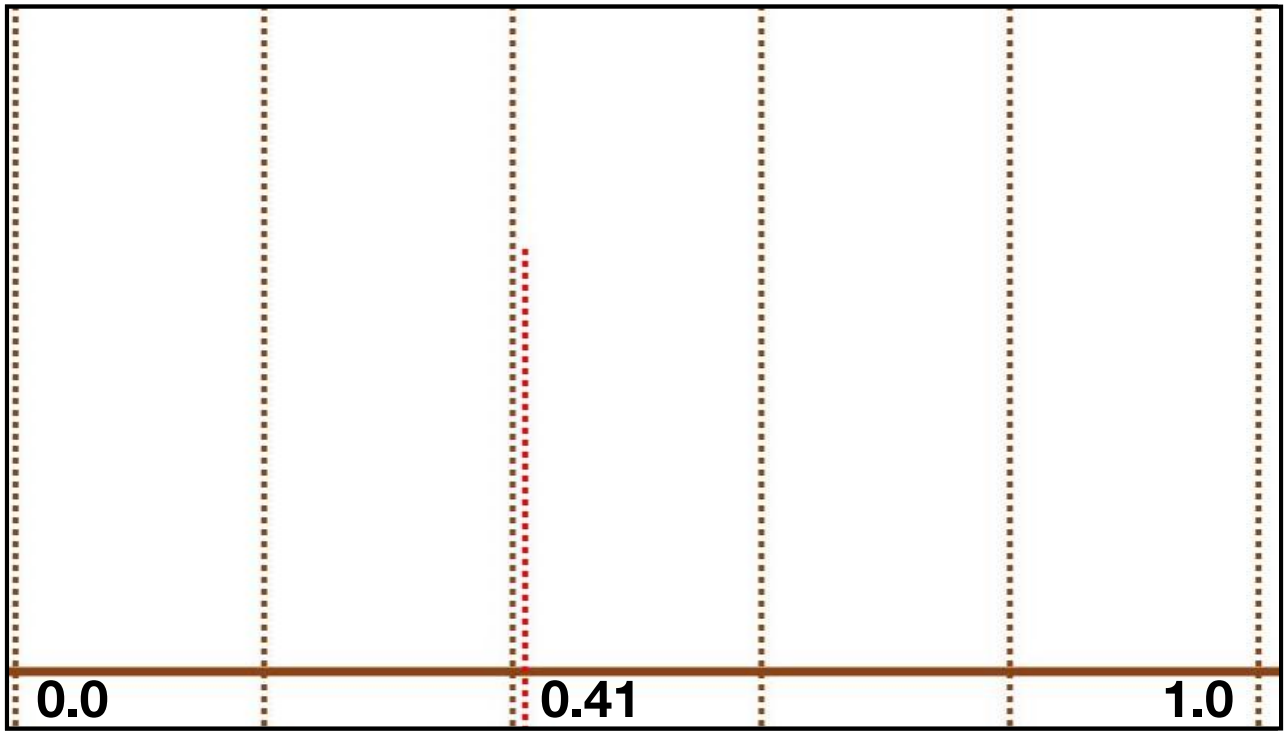
Single feature value : $f(w) = 0.41, \quad f(w) \in [0,1]$

Vectorized feature : $f(w)$ = [~0.0, 0.44, 0.54, ~0.02, ~0.0]



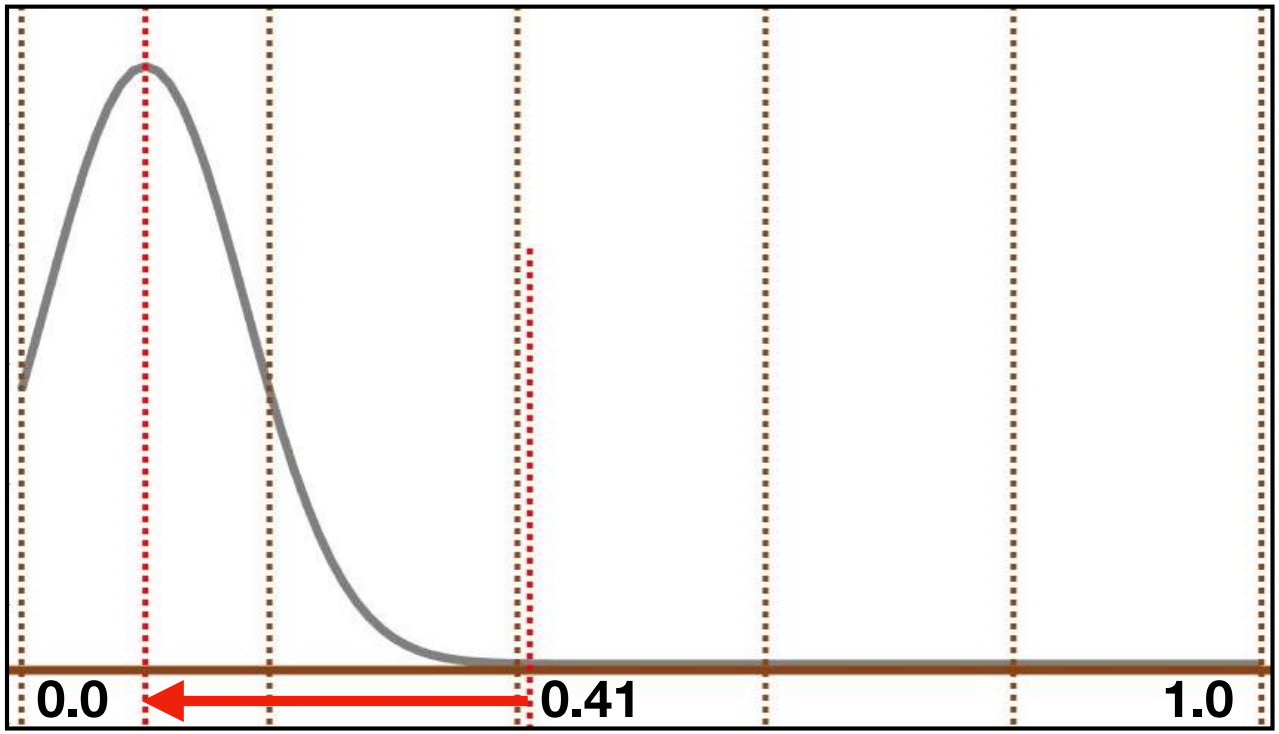
Single feature value : $f(w) = 0.41, \quad f(w) \in [0,1]$

Vectorized feature : $f(w)$ = [~0.0, 0.44, 0.54, ~0.02, ~0.0]



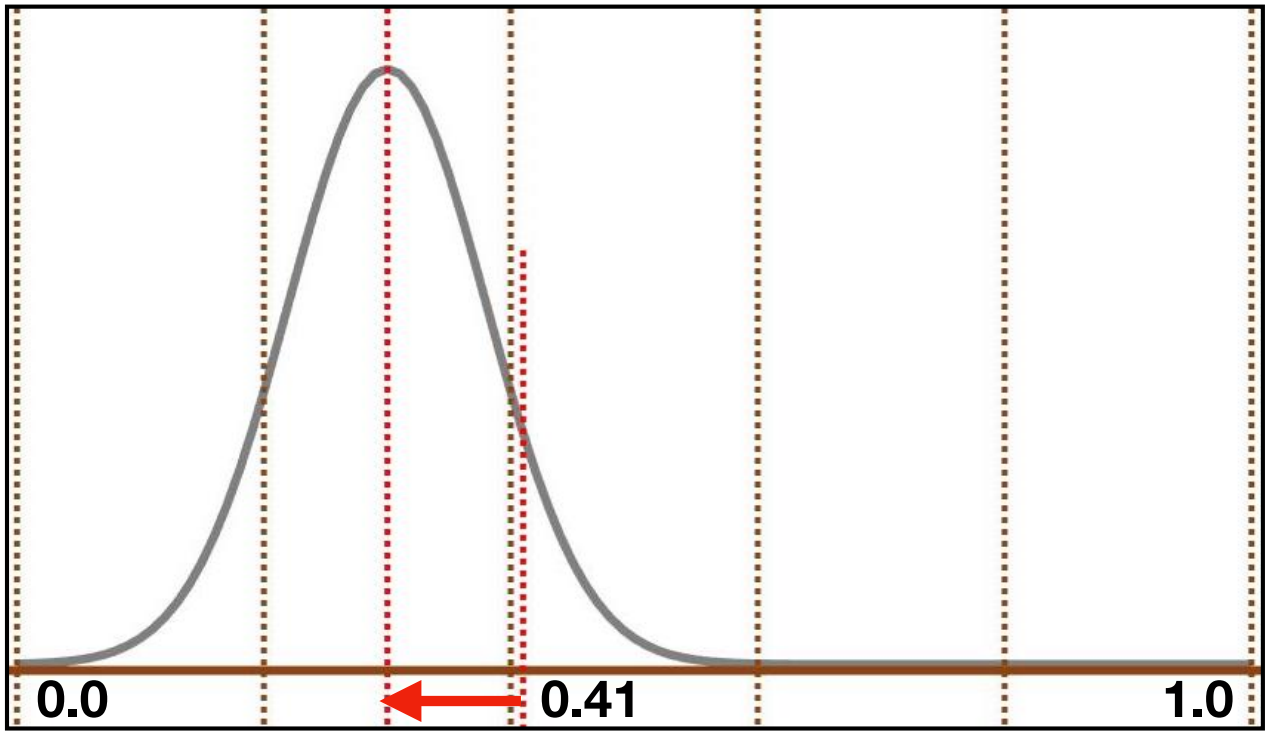
Single feature value : $f(w) = 0.41, \quad f(w) \in [0,1]$

Vectorized feature : $f(w)$ = [**~0.0**,]



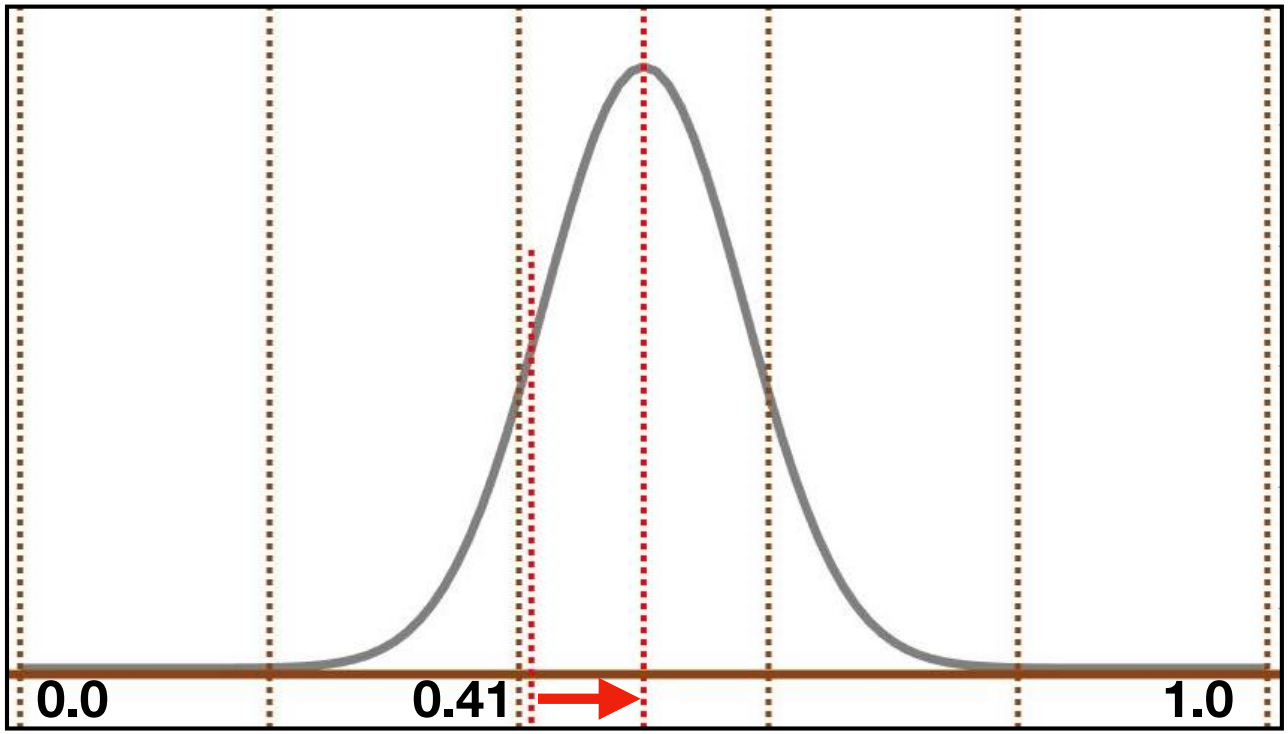
Single feature value : $f(w) = 0.41, \quad f(w) \in [0,1]$

Vectorized feature : $f(w)$ = [~0.0, **0.44**,]



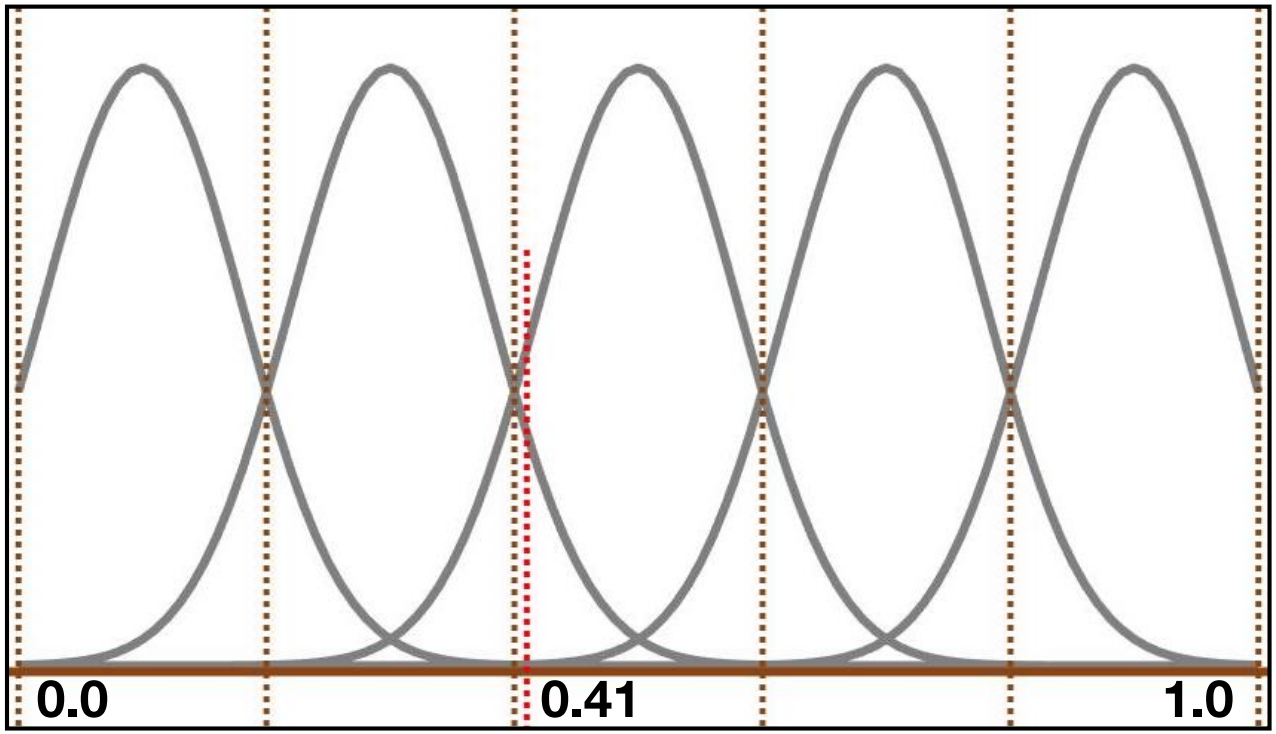
Single feature value : $f(w) = 0.41, \quad f(w) \in [0,1]$

Vectorized feature : $f(w)$ = [~0.0, 0.44, **0.54**,]



Single feature value : $f(w) = 0.41, \quad f(w) \in [0,1]$

Vectorized feature : $f(w)$ = [~0.0, **0.44**, **0.54**, ~0.02, ~0.0]



Word-Complexity Lexicon

Coverage over Penn Treebank (~1.1 million words)