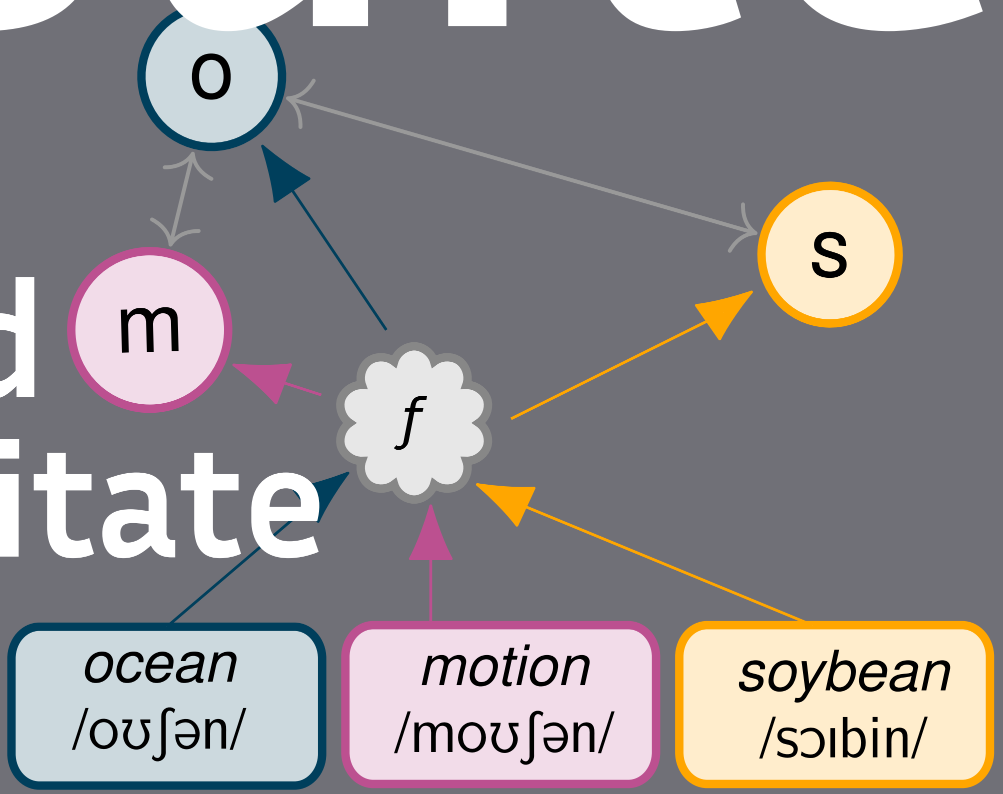
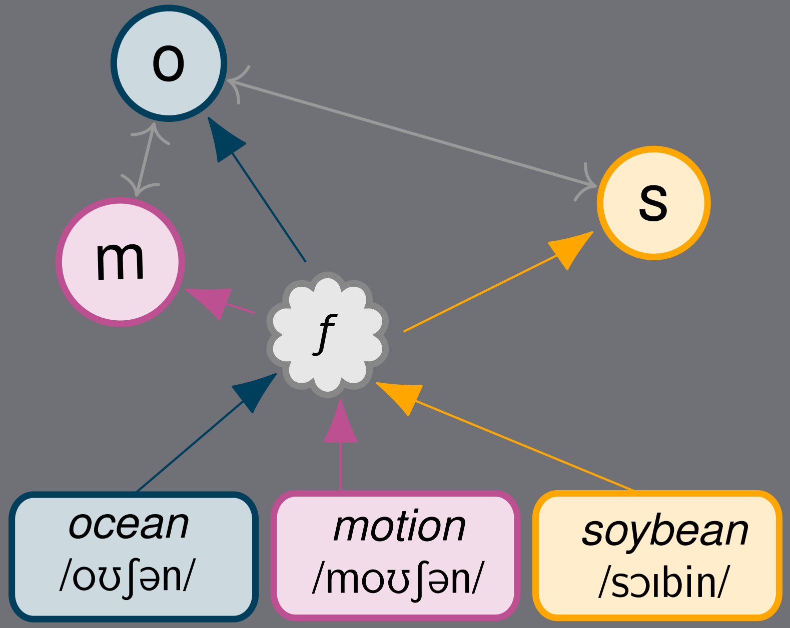
PWESuite

Phonetic Word Embeddings and Tasks They Facilitate

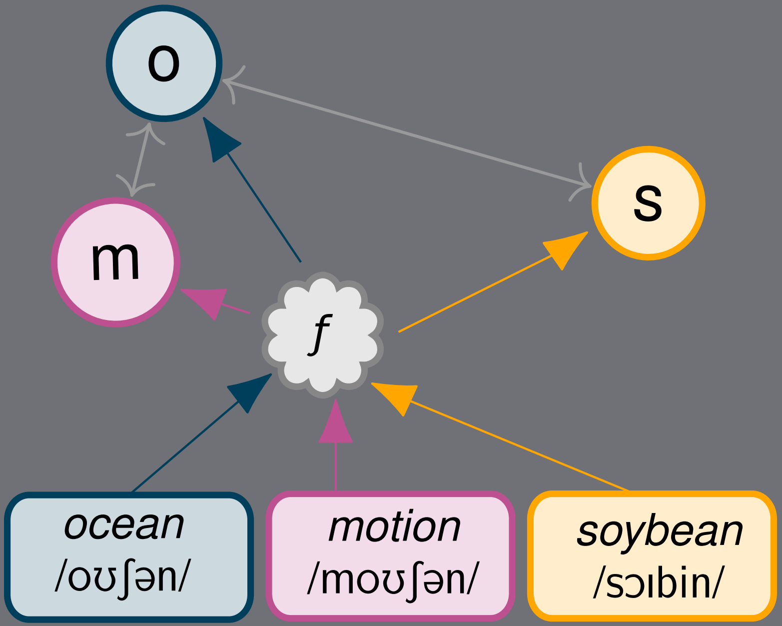


Vilém Zouhar[=] Kalvin Chang[=] Chenxuan Cui
Nathaniel Carlson Nathaniel Robinson
Mrinmaya Sachan David Mortensen

Phonetic Word Embeddings

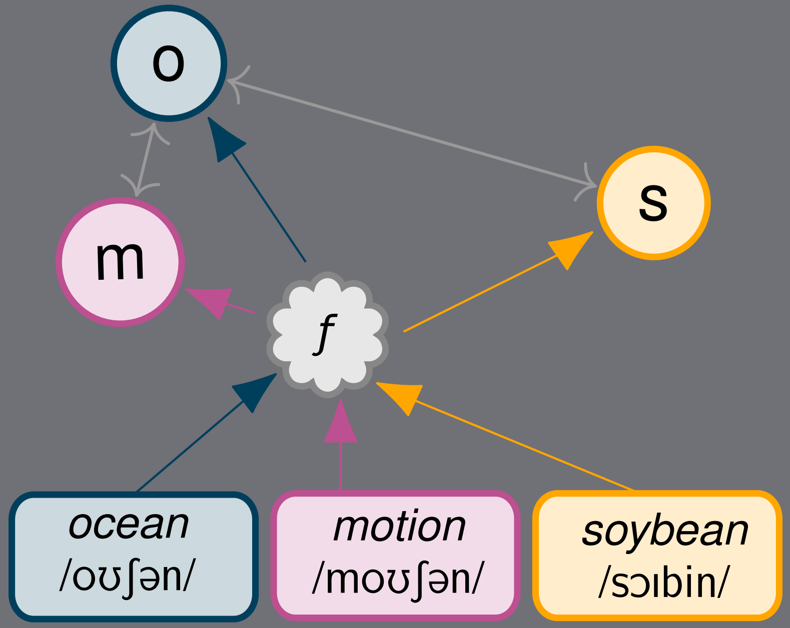


Phonetic Word Embeddings



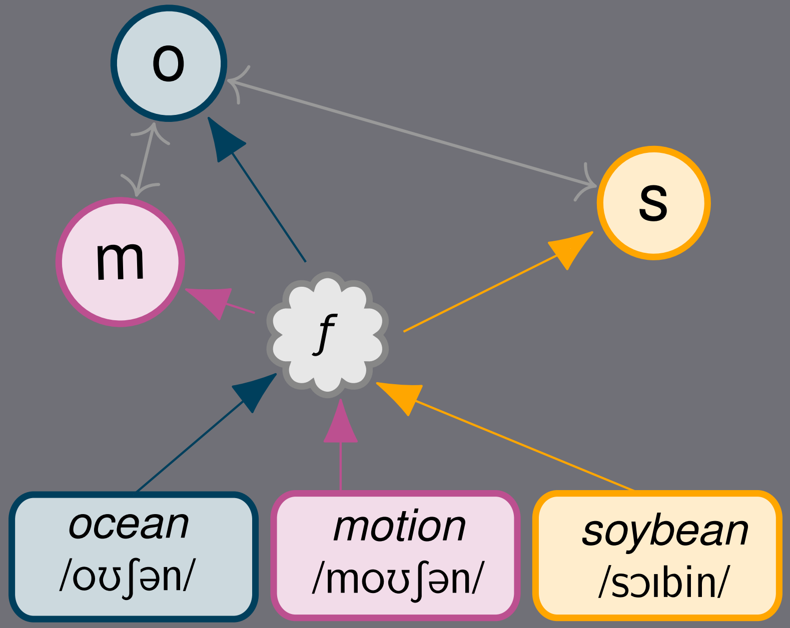
Intuition

Phonetic Word Embeddings



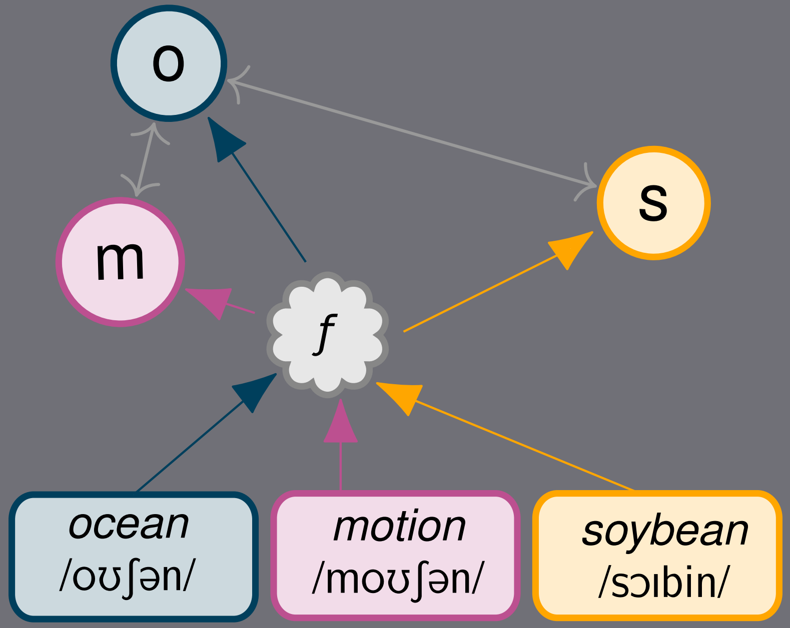
Intuition · Tasks

Phonetic Word Embeddings



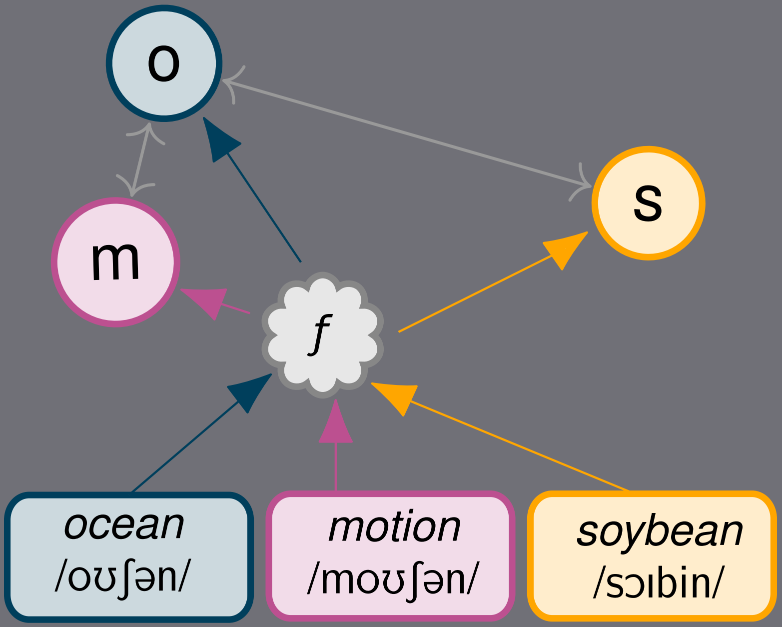
Intuition · Tasks · Application

Phonetic Word Embeddings



Intuition · Tasks · Application
Models

Phonetic Word Embeddings



Intuition · Tasks · Application
Models · Model from scratch

ocean wave motion

Word Embeddings

$$\begin{array}{c} \text{ocean} \\ \downarrow f \\ \underbrace{6\ \ 4\ \ 8\ \ 7\ \ \dots\ \ 1\ \ 2\ \ 7\ \ 0\ \ 7} \end{array} \qquad \begin{array}{c} \text{wave} \\ \downarrow f \\ \underbrace{7\ \ 0\ \ 0\ \ 1\ \ 2\ \ 8\ \ 2\ \ 1\ \ 7\ \ 7} \end{array} \qquad \begin{array}{c} \text{motion} \\ \downarrow f \\ \underbrace{2\ \ 4\ \ 5\ \ 3\ \ 8\ \ 5\ \ 0\ \ 9\ \ 0\ \ 5} \end{array}$$

ocean *wave* *motion*

Word Embeddings

Phonetic Word Embeddings

ocean wave motion

Phonetic Word Embeddings

$$\text{ocean} \xrightarrow{f} \underbrace{6\ 4\ 8\ 7\ 2\ \ldots\ 2\ 7\ 0\ 7}_{\textit{ocean}}$$

$$\text{wave} \xrightarrow{f} \underbrace{7\ 0\ 0\ 1\ 2\ \ldots\ 2\ 1\ 7\ 7}_{\textit{wave}}$$

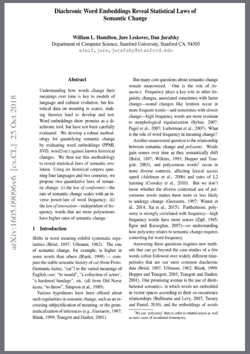
$$\text{motion} \xrightarrow{f} \underbrace{2\ 4\ 5\ 3\ 8\ \ldots\ 0\ 9\ 0\ 5}_{\textit{motion}}$$

Word Embeddings

Phonetic Word Embeddings

Applications

- Linguistic analysis



Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

William L. Hamilton, Jure Leskovec, Dan Jurafsky
Department of Computer Science, Stanford University, Stanford CA, 94305
wleif,jure,jurafsky@stanford.edu

arXiv:1605.09096v6 [cs.CL] 25 Oct 2018

Abstract

Understanding how words change their meanings over time is key to models of language and cultural evolution, but historical data on meaning is scarce, making theories hard to develop and test. Word embeddings show promise as a diachronic tool, but have not been carefully evaluated. We develop a robust methodology for quantifying semantic change by evaluating word embeddings (PPMI, SVD, word2vec) against known historical changes. We then use this methodology to reveal statistical laws of semantic evolution. Using six historical corpora spanning four languages and two centuries, we propose two quantitative laws of semantic change: (i) the *law of conformity*—the rate of semantic change scales with an inverse power-law of word frequency; (ii) the *law of innovation*—independent of frequency, words that are more polysemous have higher rates of semantic change.

1 Introduction

Shifts in word meaning exhibit systematic regularities (Bréal, 1897; Ullmann, 1962). The rate of semantic change, for example, is higher in some words than others (Blank, 1999) — compare the stable semantic history of *cat* (from Proto-Germanic *kattuz*, "cat") to the varied meanings of English *cast*: "to mould", "a collection of actors', "a hardened bandage", etc. (all from Old Norse *kasta*, "to throw", Simpson et al., 1989).

Various hypotheses have been offered about such regularities in semantic change, such as an increasing subjectification of meaning, or the grammaticalization of inferences (e.g., Geeraerts, 1997; Blank, 1999; Traugott and Dasher, 2001).

But many core questions about semantic change remain unanswered. One is the role of *frequency*. Frequency plays a key role in other linguistic changes, associated sometimes with faster change—sound changes like lenition occur in more frequent words—and sometimes with slower change—high frequency words are more resistant to morphological regularization (Bybee, 2007; Pagel et al., 2007; Lieberman et al., 2007). What is the role of word frequency in meaning change?

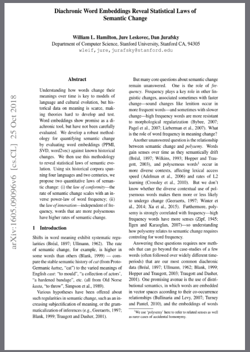
Another unanswered question is the relationship between semantic change and *polysemy*. Words gain senses over time as they semantically drift (Bréal, 1897; Wilkins, 1993; Hopper and Traugott, 2003), and polysemous words[1] occur in more diverse contexts, affecting lexical access speed (Adelman et al., 2006) and rates of L2 learning (Crossley et al., 2010). But we don't know whether the diverse contextual use of polysemous words makes them more or less likely to undergo change (Geeraerts, 1997; Winter et al., 2014; Xu et al., 2015). Furthermore, polysemy is strongly correlated with frequency—high frequency words have more senses (Zipf, 1945; İlgen and Karaoglan, 2007)—so understanding how polysemy relates to semantic change requires controling for word frequency.

Answering these questions requires new methods that can go beyond the case-studies of a few words (often followed over widely different time-periods) that are our most common diachronic data (Bréal, 1897; Ullmann, 1962; Blank, 1999; Hopper and Traugott, 2003; Traugott and Dasher, 2001). One promising avenue is the use of distributional semantics, in which words are embedded in vector spaces according to their co-occurrence relationships (Bullinaria and Levy, 2007; Turney and Pantel, 2010), and the embeddings of words

[1] We use 'polysemy' here to refer to related senses as well as rarer cases of accidental homonymy.

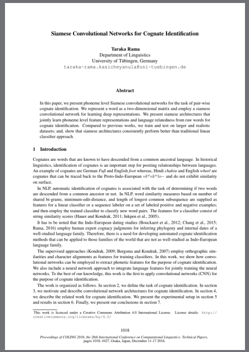
Applications

- Linguistic analysis
- Cognate/loanword detection



Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

William L. Hamilton, Jure Leskovec, Dan Jurafsky
Department of Computer Science, Stanford University, Stanford CA, 94305
wleif,jure,jurafsky@stanford.edu



Siamese Convolutional Networks for Cognate Identification

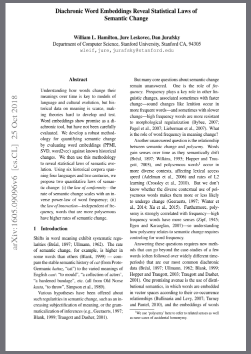
Taraka Rama
Department of Linguistics
University of Tübingen, Germany
taraka-rama.kasicheyanula@uni-tuebingen.de

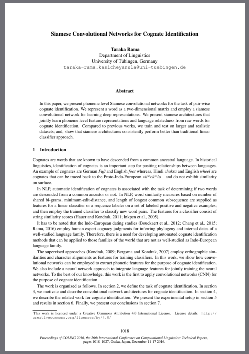
Abstract

In this paper, we present phoneme level Siamese convolutional networks for the task of pair-wise cognate identification. We represent a word as a two-dimensional matrix and employ a siamese convolutional network for learning deep representations. We present siamese architectures that jointly learn phoneme level feature representations and language relatedness from raw words for cognate identification. Compared to previous works, we train and test on larger and realistic datasets; and, show that siamese architectures consistently perform better than traditional linear classifier approach.

Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition



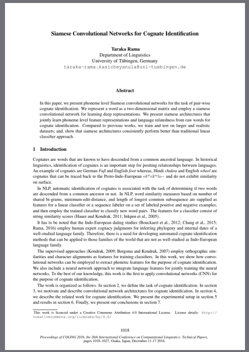


Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction

Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

William L. Hamilton, Jure Leskovec, Dan Jurafsky
Department of Computer Science, Stanford University, Stanford CA, 94305



Siamese Convolutional Networks for Cognate Identification

Taraka Rama
Department of Linguistics
University of Tübingen, Germany

Phonologically Aware Neural Model for Named Entity Recognition in Low Resource Transfer Settings

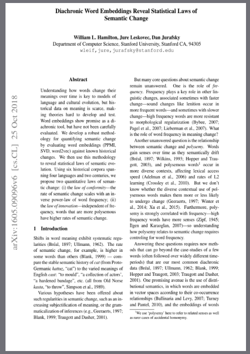
Akash Bharadwaj, David Mortensen, Chris Dyer, Jaime G. Carbonell
Language Technologies Institute
Carnegie Mellon University

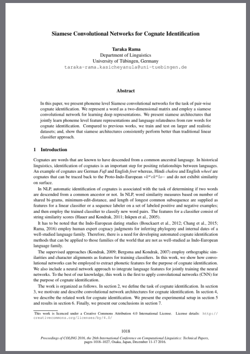
Keyphrase Extraction from Disaster-related Tweets

Jishnu Ray Chowdhury, Kansas State University
Cornelia Caragea, University of Illinois at Chicago
Doina Caragea, Kansas State University

Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction





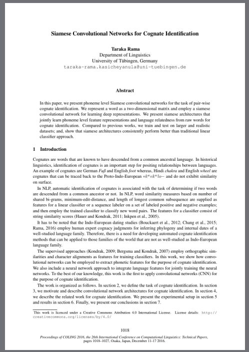
Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction
- Phonotactic learning


Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

William L. Hamilton, Jure Leskovec, Dan Jurafsky

Department of Computer Science, Stanford University, Stanford CA, 94305


Siamese Convolutional Networks for Cognate Identification

Taraka Rama

Department of Linguistics

University of Tübingen, Germany


Phonologically Aware Neural Model for Named Entity Recognition in Low Resource Transfer Settings

Akash Bharadwaj, David Mortensen, Chris Dyer, Jaime G. Carbonell

Language Technologies Institute

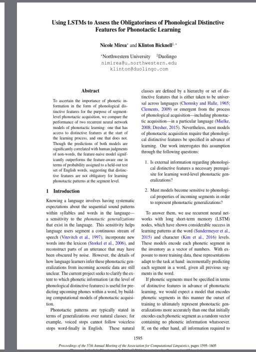
Carnegie Mellon University


Keyphrase Extraction from Disaster-related Tweets

Jishnu Ray Chowdhury, Cornelia Caragea, Doina Caragea


On the Use of Phonotactic Vector Representations with FastText for Language Identification

David Romero and Christian Salamea

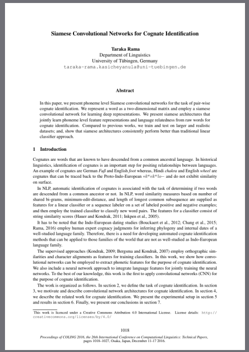

Using LSTMs to Assess the Obligatoriness of Phonological Distinctive Features for Phonotactic Learning

Nicole Mirea and Klinton Bicknell

Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction
- Phonotactic learning
- Multimodal word embeddings

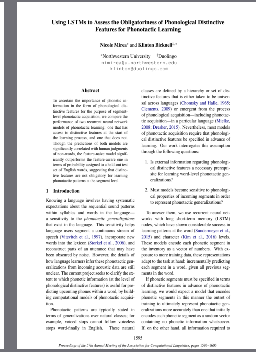

Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

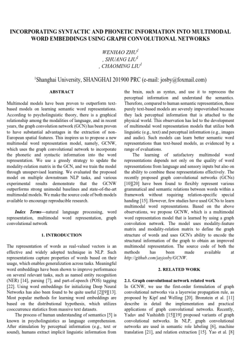

Siamese Convolutional Networks for Cognate Identification


Phonologically Aware Neural Model for Named Entity Recognition in Low Resource Transfer Settings


Keyphrase Extraction from Disaster-related Tweets


On the Use of Phonotactic Vector Representations with FastText for Language Identification


Using LSTMs to Assess the Obligatoriness of Phonological Distinctive Features for Phonotactic Learning

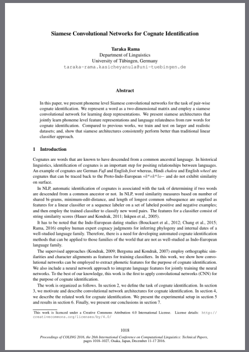

INCORPORATING SYNTACTIC AND PHONETIC INFORMATION INTO MULTIMODAL WORD EMBEDDINGS USING GRAPH CONVOLUTIONAL NETWORKS

Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction
- Phonotactic learning
- Multimodal word embeddings
- Spoken language understanding


Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

William L. Hamilton, Jure Leskovec, Dan Jurafsky


Siamese Convolutional Networks for Cognate Identification

Taraka Rama


Phonologically Aware Neural Model for Named Entity Recognition in Low Resource Transfer Settings

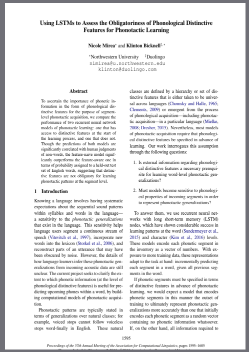
Akash Bharadwaj, David Mortensen, Chris Dyer, Jaime G. Carbonell


Keyphrase Extraction from Disaster-related Tweets

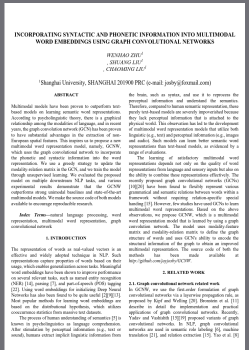
Jishnu Ray Chowdhury, Cornelia Caragea, Doina Caragea


On the Use of Phonotactic Vector Representations with FastText for Language Identification

David Romero and Christian Salamea


Using LSTMs to Assess the Obligatoriness of Phonological Distinctive Features for Phonotactic Learning

Nicole Mirea and Klinton Bicknell


INCORPORATING SYNTACTIC AND PHONETIC INFORMATION INTO MULTIMODAL WORD EMBEDDINGS USING GRAPH CONVOLUTIONAL NETWORKS

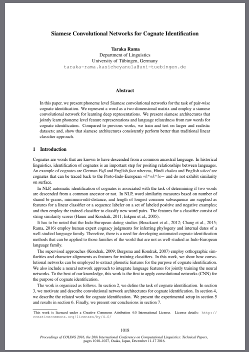

PHONETIC-AND-SEMANTIC EMBEDDING OF SPOKEN WORDS WITH APPLICATIONS IN SPOKEN CONTENT RETRIEVAL

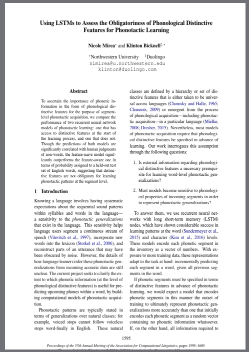
Yi-Chen Chen, Sung-Feng Huang, Chia-Hao Shen, Hung-yi Lee, Lin-shan Lee

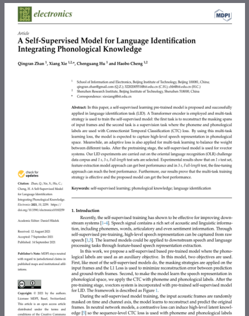
National Taiwan University, Taiwan

Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction
- Phonotactic learning
- Multimodal word embeddings
- Spoken language understanding
- Language identification

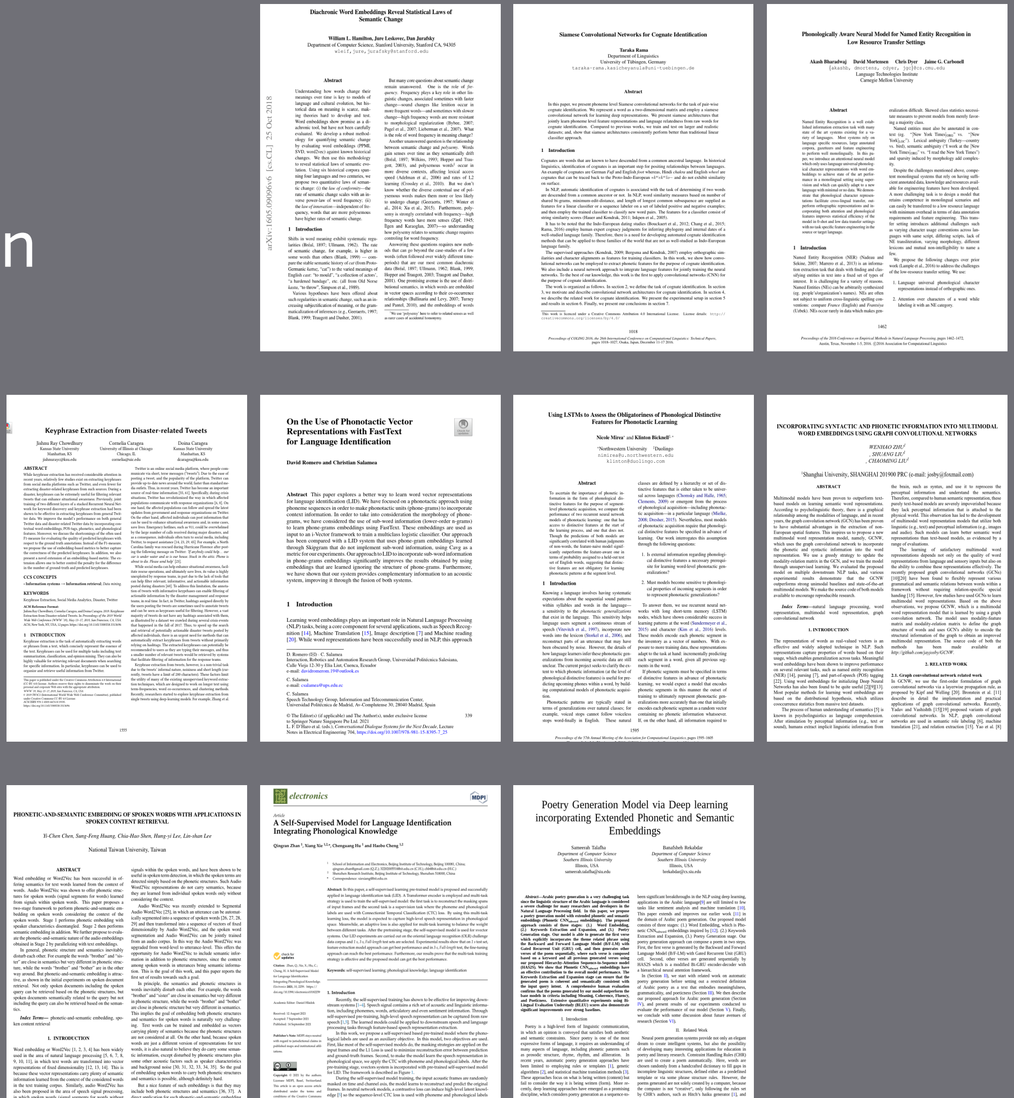






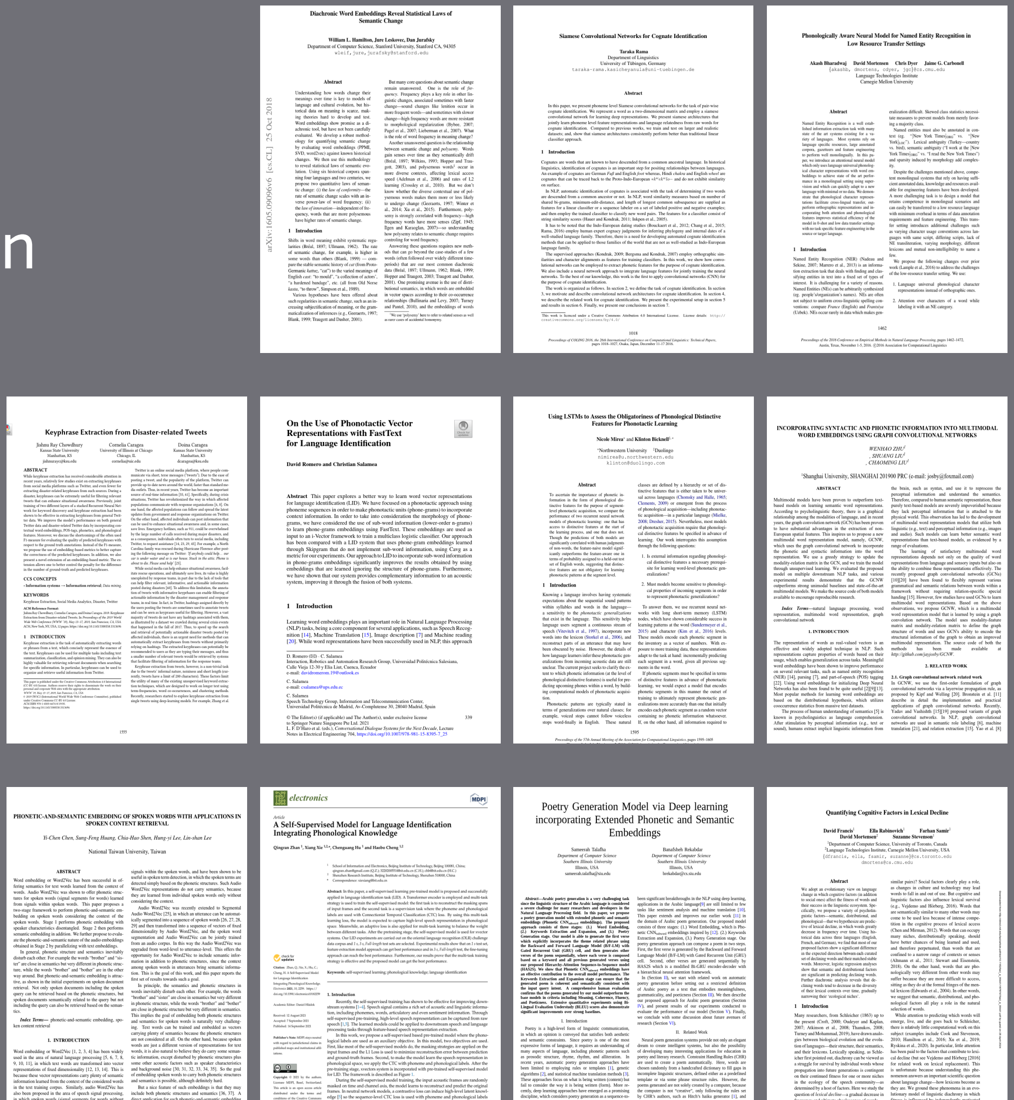
Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction
- Phonotactic learning
- Multimodal word embeddings
- Spoken language understanding
- Language identification
- Poetry generation



Applications

- Linguistic analysis
- Cognate/loanword detection
- Multilingual named entity recognition
- Keyphrase extraction
- Spelling correction
- Phonotactic learning
- Multimodal word embeddings
- Spoken language understanding
- Language identification
- Poetry generation


Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

Siamese Convolutional Networks for Cognate Identification

Phonologically Aware Neural Model for Named Entity Recognition in Low Resource Transfer Settings

Keyphrase Extraction from Disaster-related Tweets

On the Use of Phonotactic Vector Representations with FastText for Language Identification

Using LSTMs to Assess the Obligatoriness of Phonological Distinctive Features for Phonotactic Learning

INCORPORATING SYNTACTIC AND PHONETIC INFORMATION INTO MULTIMODAL WORD EMBEDDINGS USING GRAPH CONVOLUTIONAL NETWORKS

PHONETIC-AND-SEMANTIC EMBEDDING OF SPOKEN WORDS WITH APPLICATIONS IN SPOKEN CONTENT RETRIEVAL

A Self-Supervised Model for Language Identification Integrating Phonological Knowledge

Poetry Generation Model via Deep learning incorporating Extended Phonetic and Semantic Embeddings

Quantifying Cognitive Factors in Lexical Decline

Phonetic Word Embedding Tasks

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion*

ocean sounds 2% similar to *soybean*

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

sick sounds 75% similar to *sit* $S_H(\text{sick}, \text{sit}) = 0.75$

...

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

sick sounds 75% similar to *sit* $S_H(\text{sick}, \text{sit}) = 0.75$

...

$$f(\text{ocean}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline 6 & 4 & 8 & 7 & 2 & \ldots & 2 & 7 & 0 & 7 \\ \hline \end{array}$$

$$f(\text{motion}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline 5 & 1 & 5 & 9 & 1 & \ldots & 4 & 8 & 7 & 2 \\ \hline \end{array}$$

$$f(\text{soybean}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline 5 & 3 & 8 & 5 & 0 & \ldots & 0 & 5 & 5 & 1 \\ \hline \end{array}$$

$$\ldots$$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

sick sounds 75% similar to *sit* $S_H(\text{sick}, \text{sit}) = 0.75$

...

$f(\text{ocean}) =$ 6 4 8 7 2 ... 2 7 0 7

$f(\text{motion}) =$ 5 1 5 9 1 ... 4 8 7 2

$f(\text{soybean}) =$ 5 3 8 5 0 ... 0 5 5 1

$\ldots$

$S_V(f(\text{ocean}), f(\text{motion})) = 2$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

sick sounds 75% similar to *sit* $S_H(\text{sick}, \text{sit}) = 0.75$

...

$$f(\text{ocean}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline 6 & 4 & 8 & 7 & 2 & \ldots & 2 & 7 & 0 & 7 \\ \hline\end{array}$$

$$f(\text{motion}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline 5 & 1 & 5 & 9 & 1 & \ldots & 4 & 8 & 7 & 2 \\ \hline\end{array}$$

$$f(\text{soybean}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline 5 & 3 & 8 & 5 & 0 & \ldots & 0 & 5 & 5 & 1 \\ \hline\end{array}$$

$\ldots$

$$S_V(f(\text{ocean}), f(\text{motion})) = 2$$

$$S_V(f(\text{ocean}), f(\text{soybean})) = -0.1$$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $$S_H(\text{ocean}, \text{motion}) = 0.7$$

ocean sounds 2% similar to *soybean* $$S_H(\text{ocean}, \text{soybean}) = 0.02$$

screech sounds 11% similar to *plant* $$S_H(\text{screech}, \text{plant}) = 0.11$$

sick sounds 75% similar to *sit* $$S_H(\text{sick}, \text{sit}) = 0.75$$

...

$$f(\text{ocean}) = \begin{bmatrix} 6 & 4 & 8 & 7 & 2 & \ldots & 2 & 7 & 0 & 7 \end{bmatrix}$$

$$f(\text{motion}) = \begin{bmatrix} 5 & 1 & 5 & 9 & 1 & \ldots & 4 & 8 & 7 & 2 \end{bmatrix}$$

$$f(\text{soybean}) = \begin{bmatrix} 5 & 3 & 8 & 5 & 0 & \ldots & 0 & 5 & 5 & 1 \end{bmatrix}$$

$$\ldots$$

$$S_V(f(\text{ocean}), f(\text{motion})) = 2$$

$$S_V(f(\text{ocean}), f(\text{soybean})) = -0.1$$

$$S_V(f(\text{screech}), f(\text{plant})) = 0.3$$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

sick 0.75

Correlation between S_H and S_V with f

$f(\text{ocean}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} 6 & 4 & 8 & 7 & 2 & \ldots & 2 & 7 & 0 & 7 \end{array}$ $S_V(f(\text{ocean}), f(\text{motion})) = 2$

$f(\text{motion}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} 5 & 1 & 5 & 9 & 1 & \ldots & 4 & 8 & 7 & 2 \end{array}$ $S_V(f(\text{ocean}), f(\text{soybean})) = -0.1$

$f(\text{soybean}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} 5 & 3 & 8 & 5 & 0 & \ldots & 0 & 5 & 5 & 1 \end{array}$ $S_V(f(\text{screech}), f(\text{plant})) = 0.3$

$\ldots$

Task 1: Sound Similarity Correlation

ocean sounds 70% similar to *motion* $S_H(\text{ocean}, \text{motion}) = 0.7$

ocean sounds 2% similar to *soybean* $S_H(\text{ocean}, \text{soybean}) = 0.02$

screech sounds 11% similar to *plant* $S_H(\text{screech}, \text{plant}) = 0.11$

sick 0.75

Correlation between S_H and S_V with f

S_H is either human or articulatory feature distance

$f($ $= 2$

$f(\text{motion}) =$ | 5 | 1 | 5 | 9 | 1 | ... | 4 | 8 | 7 | 2 |

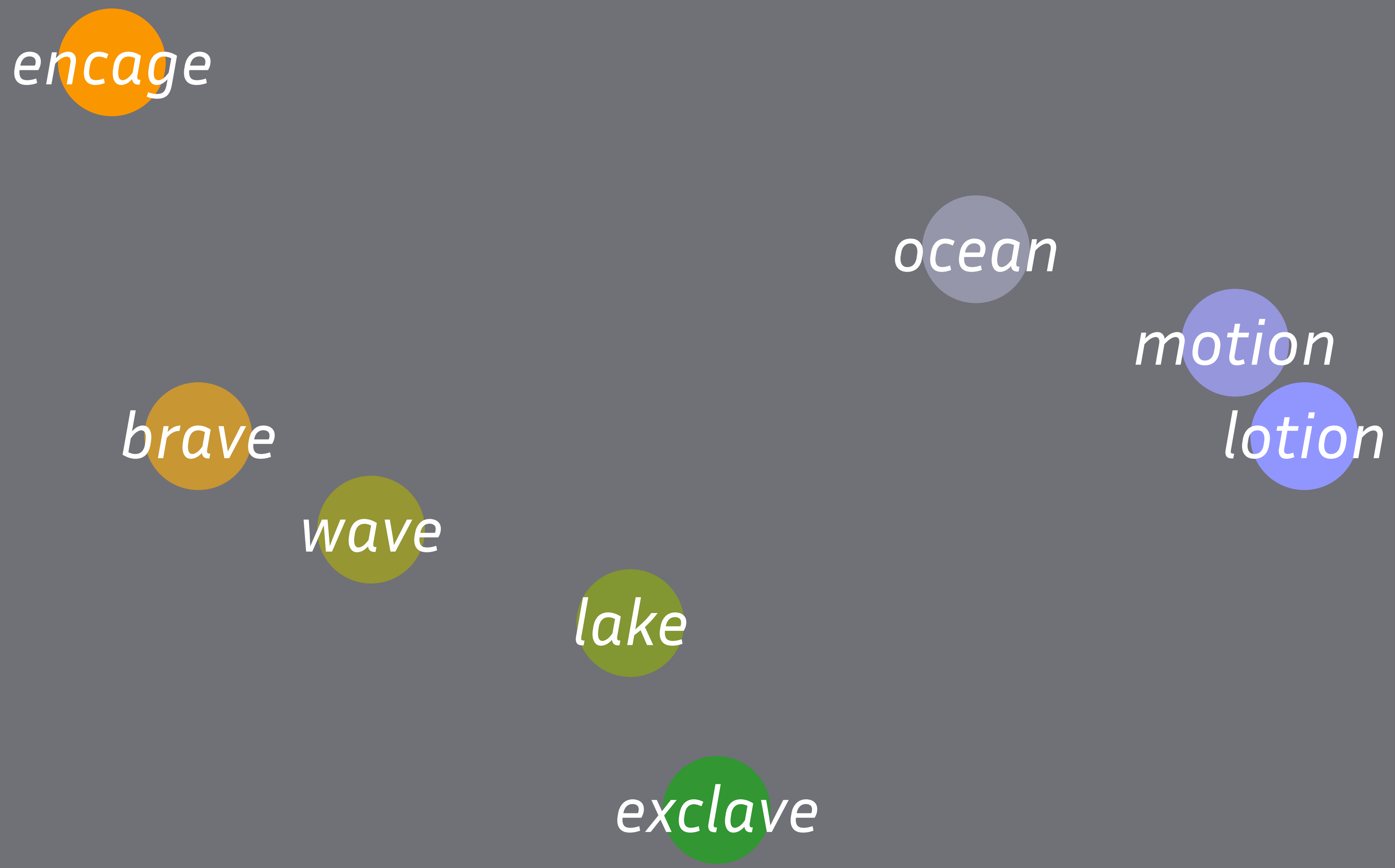
$S_V(f(\text{ocean}), f(\text{soybean})) = -0.1$

$f(\text{soybean}) =$ | 5 | 3 | 8 | 5 | 0 | ... | 0 | 5 | 5 | 1 |

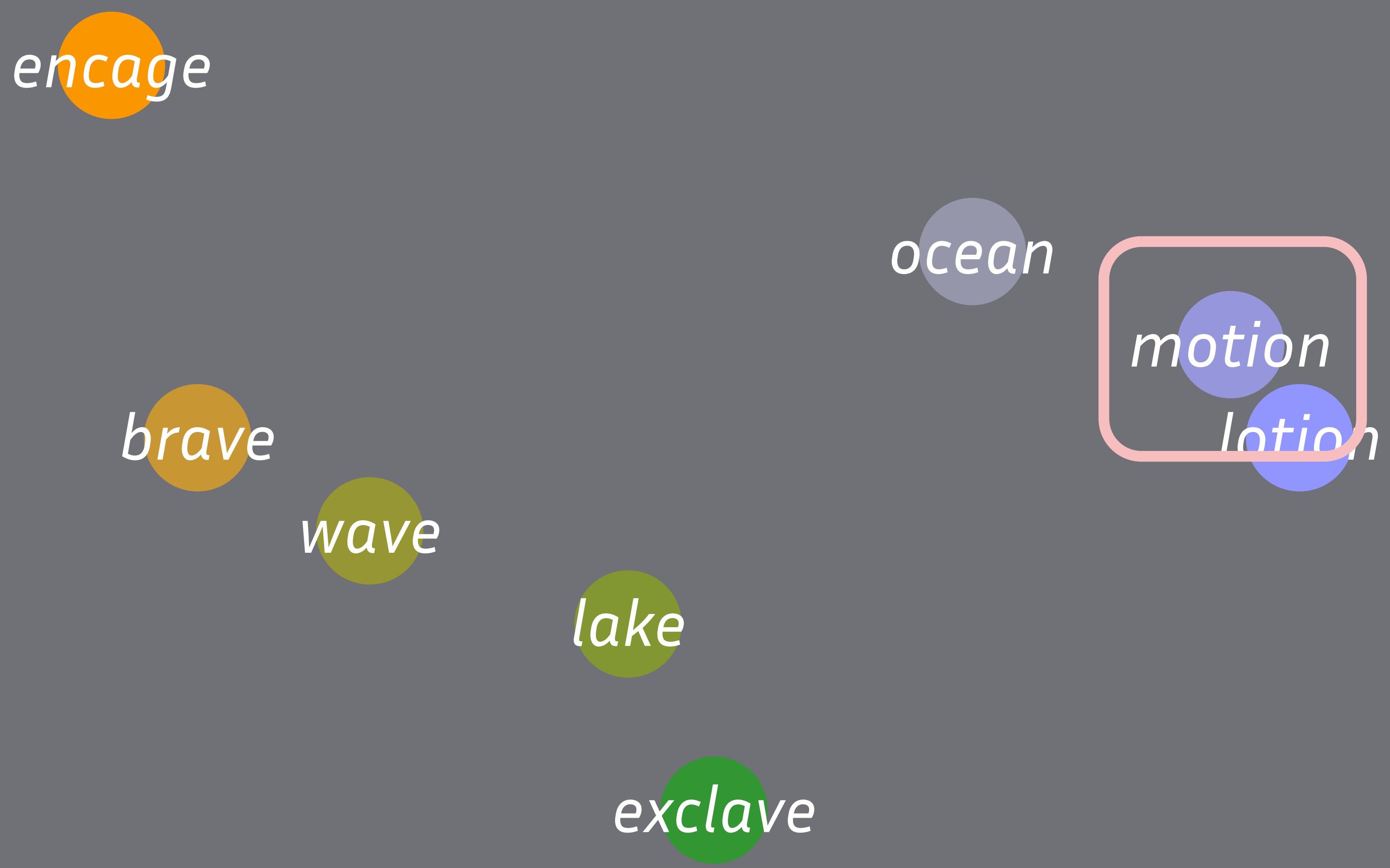
$S_V(f(\text{screech}), f(\text{plant})) = 0.3$

$\dots$

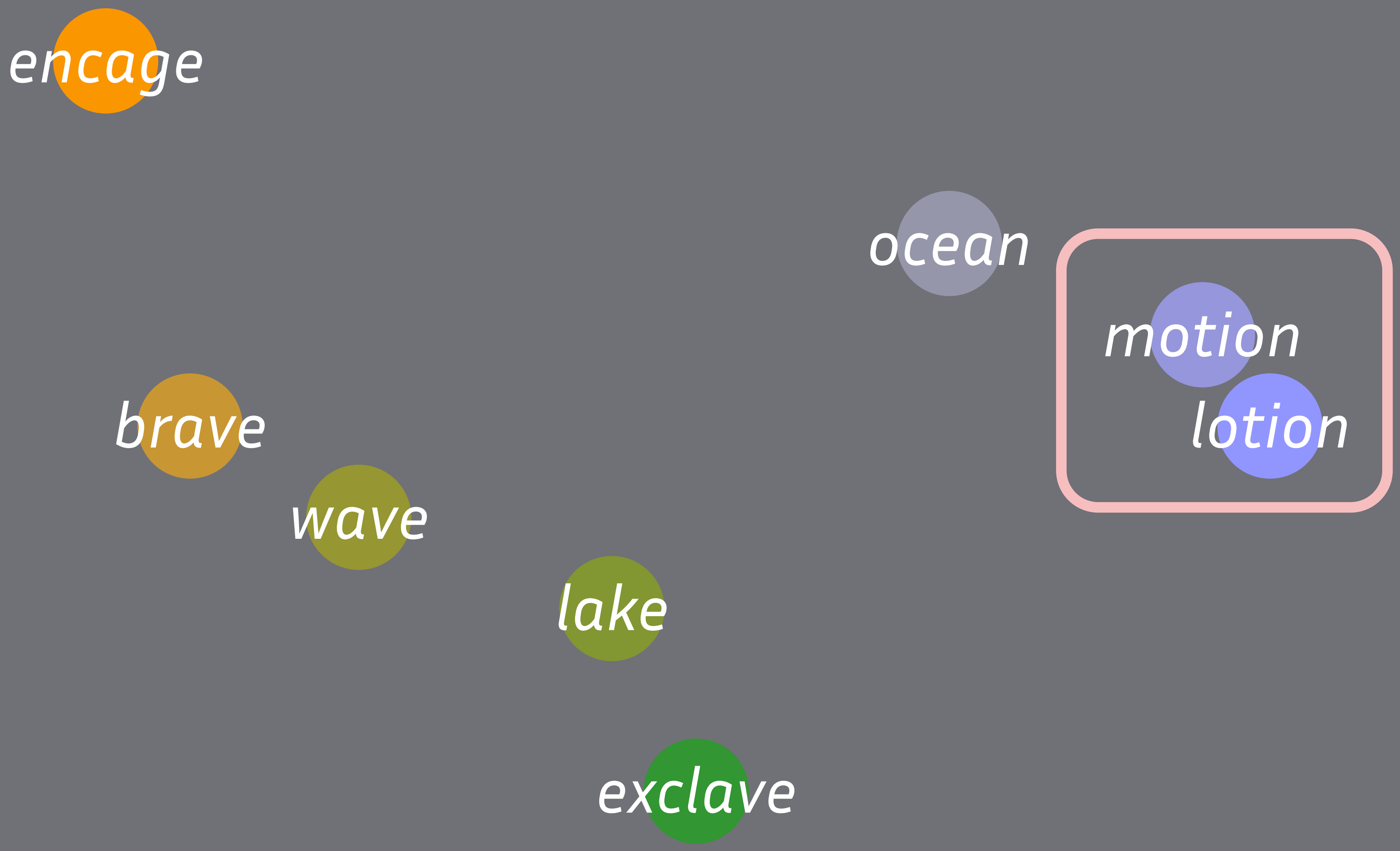
Task 2: Sound Similarity Retrieval



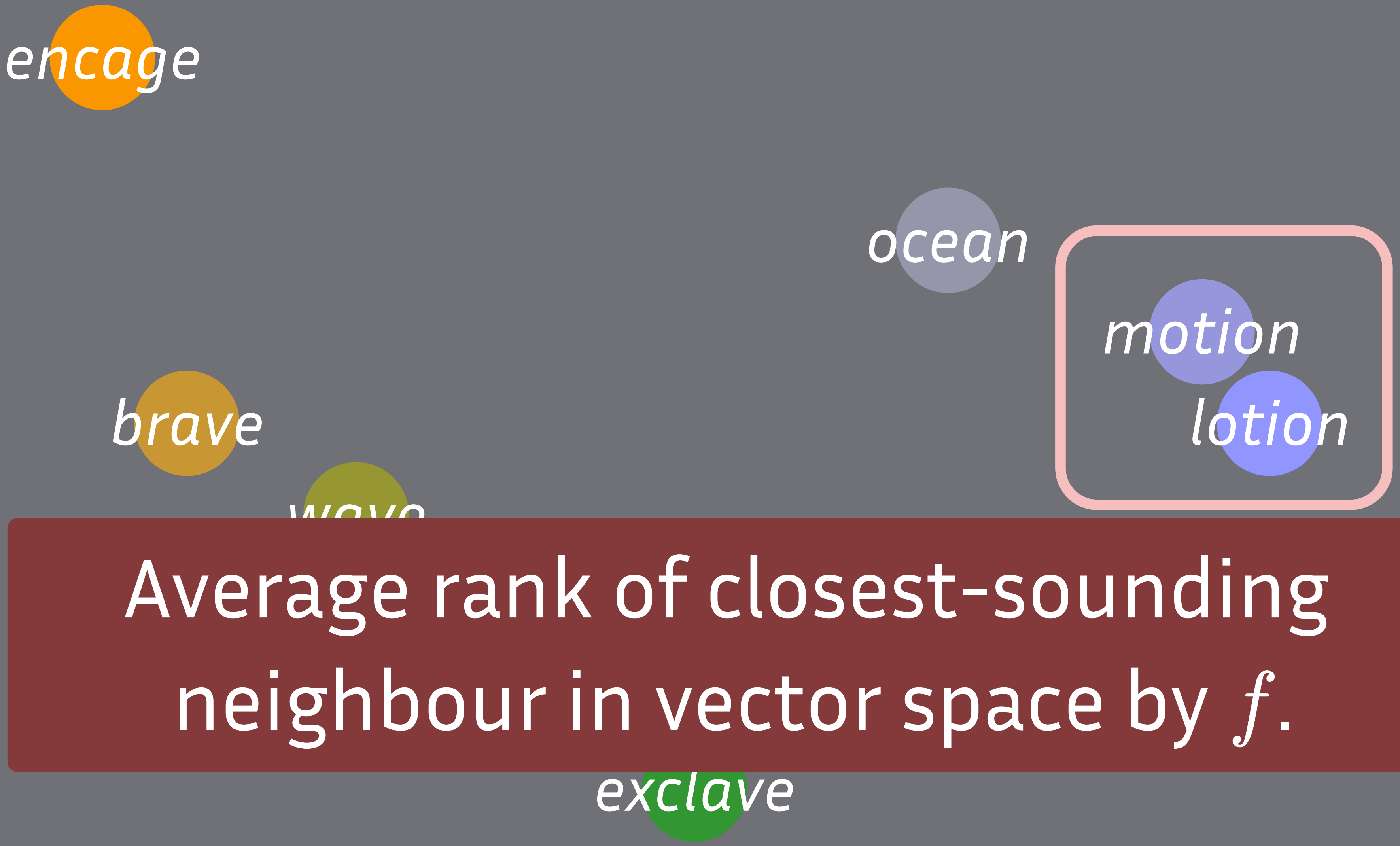
Task 2: Sound Similarity Retrieval



Task 2: Sound Similarity Retrieval



Task 2: Sound Similarity Retrieval



Average rank of closest-sounding neighbour in vector space by f.

Task 2: Sound Similarity Retrieval

$w \in \mathcal{W} \quad w' = \mathrm{ArgMax}_{w' \in \mathcal{W} \setminus \{w\}} S_H(w, w').$

In S_V with f, w' is r-th neighbour of w.

Metric: $\frac{1000 - r}{1000}$.

ocean

motion

lotion

brave

Average rank of closest-sounding neighbour in vector space by f.

exclave

Task 3: Rhyme Detection

ocean & motion	rhyme
bean & queen	rhyme
ocean & soybean	no rhyme

…

Task 3: Rhyme Detection

ocean & motion	rhyme
bean & queen	rhyme
ocean & soybean	no rhyme

...

$$\mathrm{MLP}(f(\text{ocean}), f(\text{motion})) \to \texttt{rhyme}$$

$$\mathrm{MLP}(f(\text{bean}), f(\text{queen})) \to \texttt{rhyme}$$

$$\mathrm{MLP}(f(\text{ocean}), f(\text{soybean})) \to \texttt{no-rhyme}$$

Task 3: Rhyme Detection

ocean & motion	rhyme
bean & queen	rhyme
ocean & soybean	no rhyme

...

$$\mathrm{MLP}(f(\mathrm{ocean}), f(\mathrm{motion})) \to \texttt{rhyme}$$

$$\mathrm{MLP}(f(\mathrm{bean}), f(\mathrm{queen})) \to \texttt{rhyme}$$

$$\mathrm{MLP}(f(\mathrm{ocean}), f(\mathrm{soybean})) \to \texttt{no-rhyme}$$

Accuracy of MLP on embeddings by f.

Task 4: Cognate Detection

plant$_{EN}$ & plante$_{FR}$	cognate
plane$_{EN}$ & plante$_{FR}$	no cognate
much$_{EN}$ & mucho$_{ES}$	no cognate

…

Task 4: Cognate Detection

plant_{EN} & plante_{FR}	cognate
plane_{EN} & plante_{FR}	no cognate
much_{EN} & mucho_{ES}	no cognate

...

$$\mathrm{MLP}(f(\mathrm{plant}), f(\mathrm{plante})) \rightarrow \texttt{cognate}$$

$$\mathrm{MLP}(f(\mathrm{plane}), f(\mathrm{plante})) \rightarrow \texttt{no-cognate}$$

$$\mathrm{MLP}(f(\mathrm{much}), f(\mathrm{mucho})) \rightarrow \texttt{no-cognate}$$

Task 4: Cognate Detection

plant$_{EN}$ & plante$_{FR}$ cognate

plane$_{EN}$ & plante$_{FR}$ no cognate

much$_{EN}$ & mucho$_{ES}$ no cognate

...

$$\mathrm{MLP}(f(\mathrm{plant}), f(\mathrm{plante})) \rightarrow \texttt{cognate}$$

$$\mathrm{MLP}(f(\mathrm{plane}), f(\mathrm{plante})) \rightarrow \texttt{no-cognate}$$

$$\mathrm{MLP}(f(\mathrm{much}), f(\mathrm{mucho})) \rightarrow \texttt{no-cognate}$$

Accuracy of MLP on embeddings by f.

Task 5: Sound Analogies

man : woman ↔ king : queen

Task 5: Sound Analogies

man : woman ↔ king : queen

$$f(\text{man}) - f(\text{woman}) + f(\text{king}) \rightsquigarrow f(\text{queen})$$

Task 5: Sound Analogies

man : woman ↔ king : queen

$$f(\mathrm{man}) - f(\mathrm{woman}) + f(\mathrm{king}) \leadsto f(\mathrm{queen})$$

`/dIn/` : `/tIn/` ↔ `/zIn/` : `/sIn/`

Task 5: Sound Analogies

man : woman ↔ king : queen

$$f(\text{man}) - f(\text{woman}) + f(\text{king}) \rightsquigarrow f(\text{queen})$$

`/dIn/` : `/tIn/` ↔ `/zIn/` : `/sIn/`

$$f(\texttt{/dIn/}) - f(\texttt{/tIn/}) + f(\texttt{/zIn/}) \rightsquigarrow f(\texttt{/sIn/})$$

Task 5: Sound Analogies

man : woman ↔ king : queen $\quad f(\text{man}) - f(\text{woman}) + f(\text{king}) \rightsquigarrow f(\text{queen})$

`/dIn/` : `/tIn/` ↔ `/zIn/` : `/sIn/` $\quad f(\texttt{/dIn/}) - f(\texttt{/tIn/}) + f(\texttt{/zIn/}) \rightsquigarrow f(\texttt{/sIn/})$

$$w_1 : w_2 \leftrightarrow {w_1}' : {w_2}'$$

Task 5: Sound Analogies

man : woman ↔ king : queen $\quad f(\text{man}) - f(\text{woman}) + f(\text{king}) \rightsquigarrow f(\text{queen})$

/dIn/ : /tIn/ ↔ /zIn/ : /sIn/ $\quad f(\texttt{/dIn/}) - f(\texttt{/tIn/}) + f(\texttt{/zIn/}) \rightsquigarrow f(\texttt{/sIn/})$

$$w_1 : w_2 \leftrightarrow {w_1}' : {w_2}'$$

How often is $f({w_2}')$ the closest neighbour to $(f(w_1) - f(w_2) + f({w_1}'))$?

Phonetic Word Embedding Models

Model 0: Count-based

s	ʊ	m	b	n	ɪ	ə	o	ː	ʃ	k	i	w	ɔ

sɔɪbiːn $\underset{f}{\longrightarrow}$

1	0	0	1	1	1	0	0	1	0	0	1	0	1

Model 0: Count-based

	s	ʊ	m	b	n	ɪ	ə	o	ː	ʃ	k	i	w	ɔ
sɔɪbiːn $\xrightarrow[f]{}$	1	0	0	1	1	1	0	0	1	0	0	1	0	1
kwiːn $\xrightarrow[f]{}$	0	0	0	0	1	0	0	0	1	0	1	1	1	0

Model 0: Count-based

		s	ʊ	m	b	n	ɪ	ə	o	ː	ʃ	k	i	w	ɔ
sɔɪbiːn	$\xrightarrow[f]{}$	1	0	0	1	1	1	0	0	1	0	0	1	0	1
kwiːn	$\xrightarrow[f]{}$	0	0	0	0	1	0	0	0	1	0	1	1	1	0
oʊʃən	$\xrightarrow[f]{}$	0	1	0	0	1	0	1	1	0	1	0	0	0	0

Model 0: Count-based

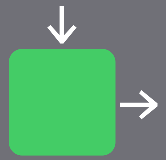
		s	ʊ	m	b	n	ɪ	ə	o	ː	ʃ	k	i	w	ɔ
sɔɪbiːn	$\underset{f}{\rightarrow}$	1	0	0	1	1	1	0	0	1	0	0	1	0	1
kwiːn	$\underset{f}{\rightarrow}$	0	0	0	0	1	0	0	0	1	0	1	1	1	0
oʊʃən	$\underset{f}{\rightarrow}$	0	1	0	0	1	0	1	1	0	1	0	0	0	0
moʊʃən	$\underset{f}{\rightarrow}$	0	1	1	0	1	0	1	1	0	1	0	0	0	0

Model 0: Count-based

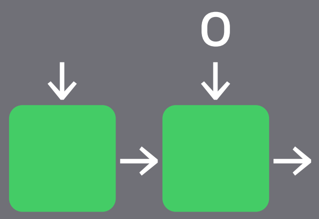
		s	ʊ	m	b	n	ɪ	ə	o	ː	ʃ	k	i	w	ɔ
sɔɪbiːn	$\xrightarrow{f}$	1	0	0	1	1	1	0	0	1	0	0	1	0	1
kwiːn	$\xrightarrow{f}$	0	0	0	0	1	0	0	0	1	0	1	1	1	0
oʊʃən	$\xrightarrow{f}$	0	1	0	0	1	0	1	1	0	1	0	0	0	0
moʊʃən	$\xrightarrow{f}$	0	1	1	0	1	0	1	1	0	1	0	0	0	0

Model 1: Autoencoder

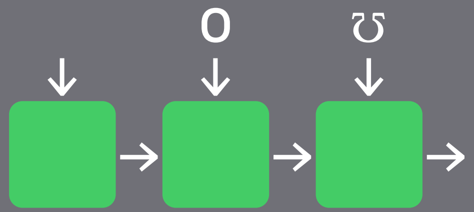
Model 1: Autoencoder



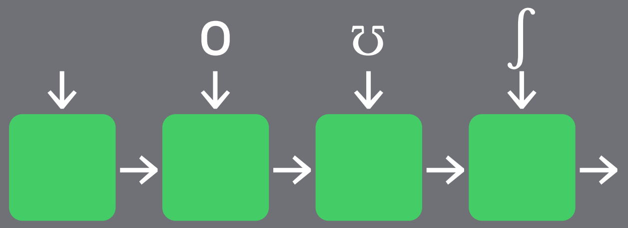
Model 1: Autoencoder



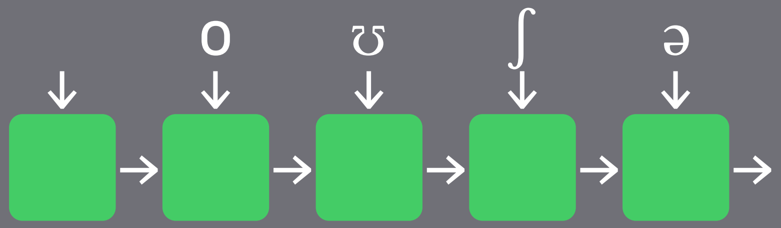
Model 1: Autoencoder



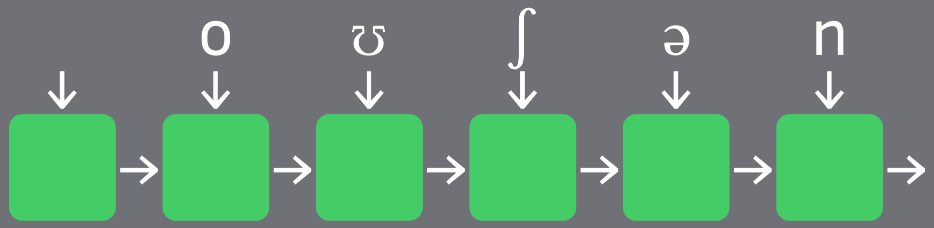
Model 1: Autoencoder



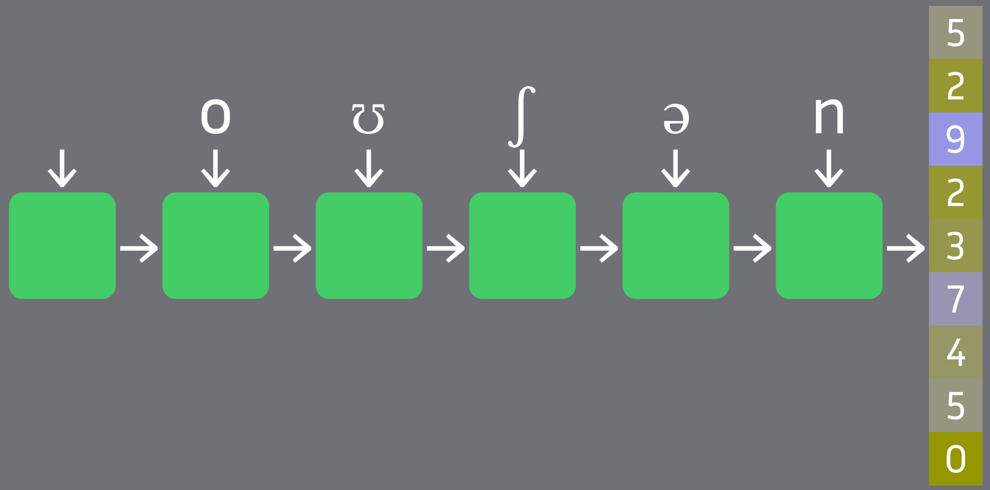
Model 1: Autoencoder



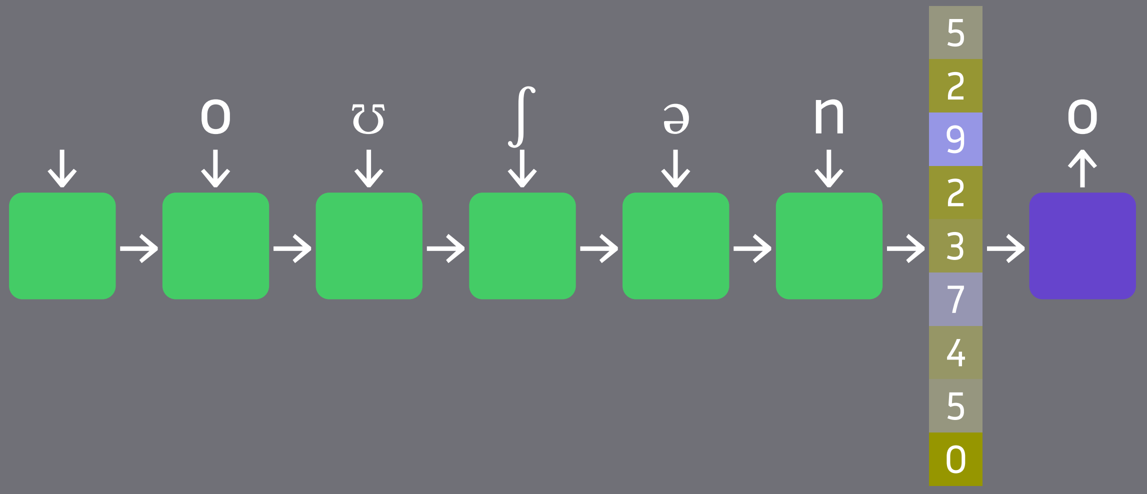
Model 1: Autoencoder



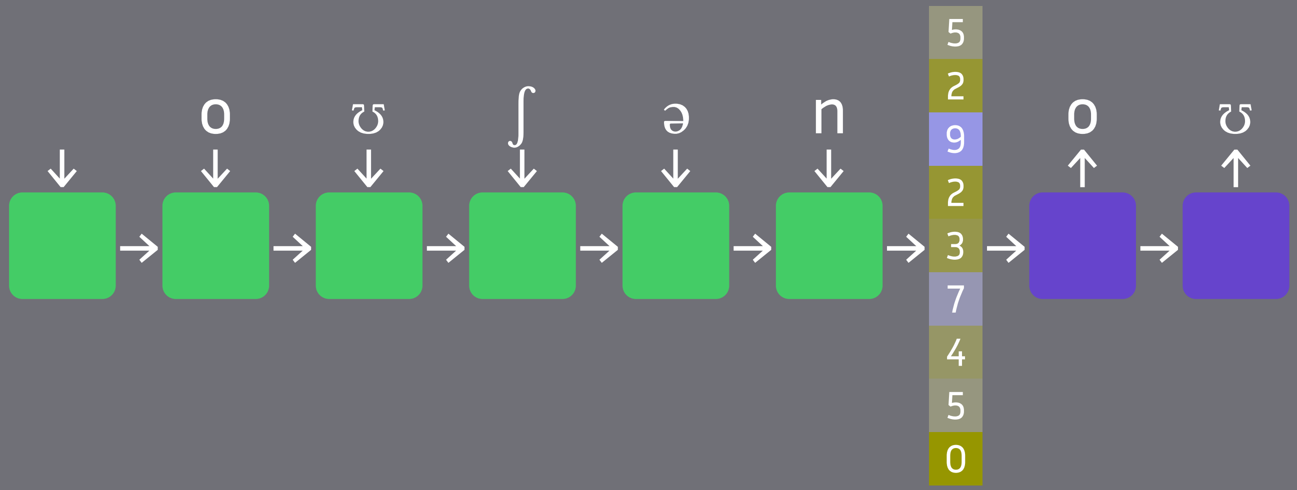
Model 1: Autoencoder



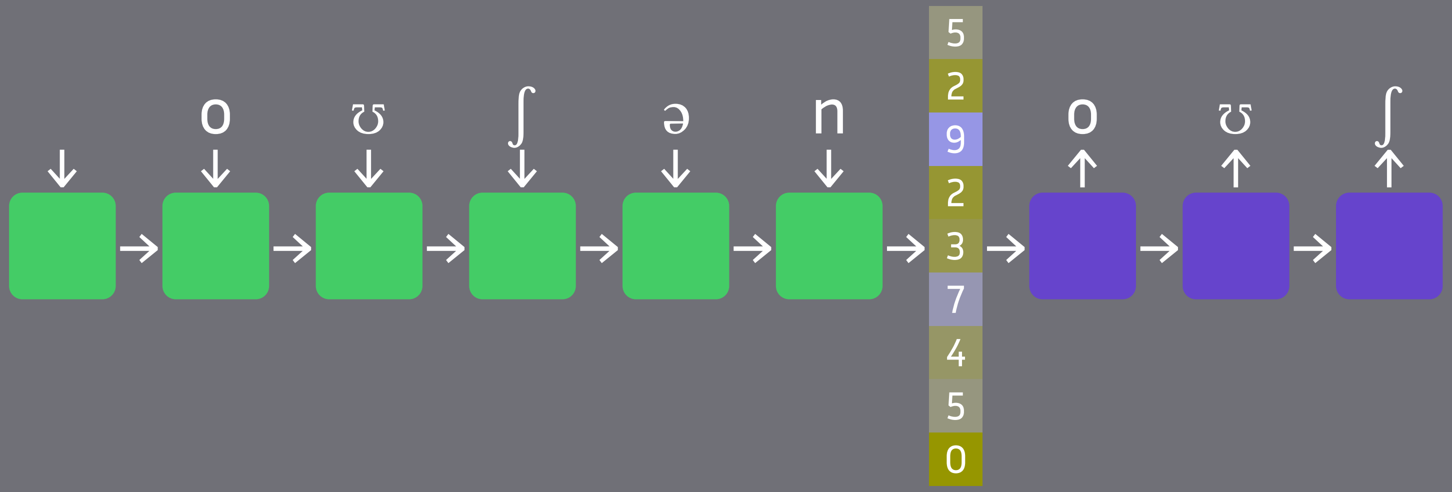
Model 1: Autoencoder



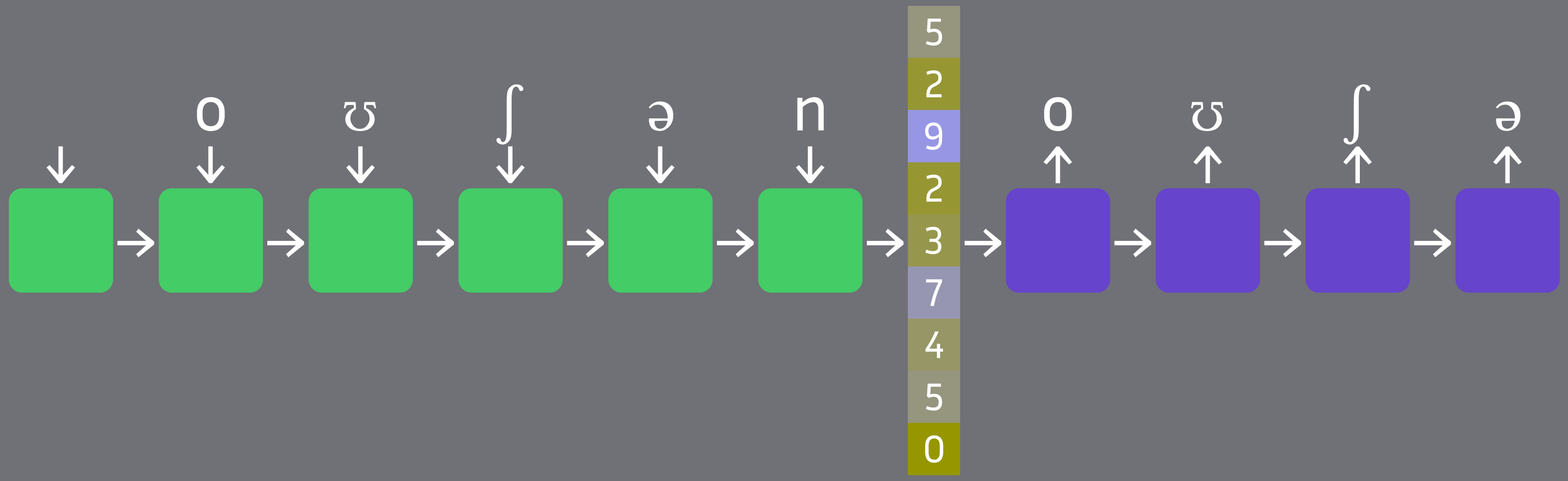
Model 1: Autoencoder



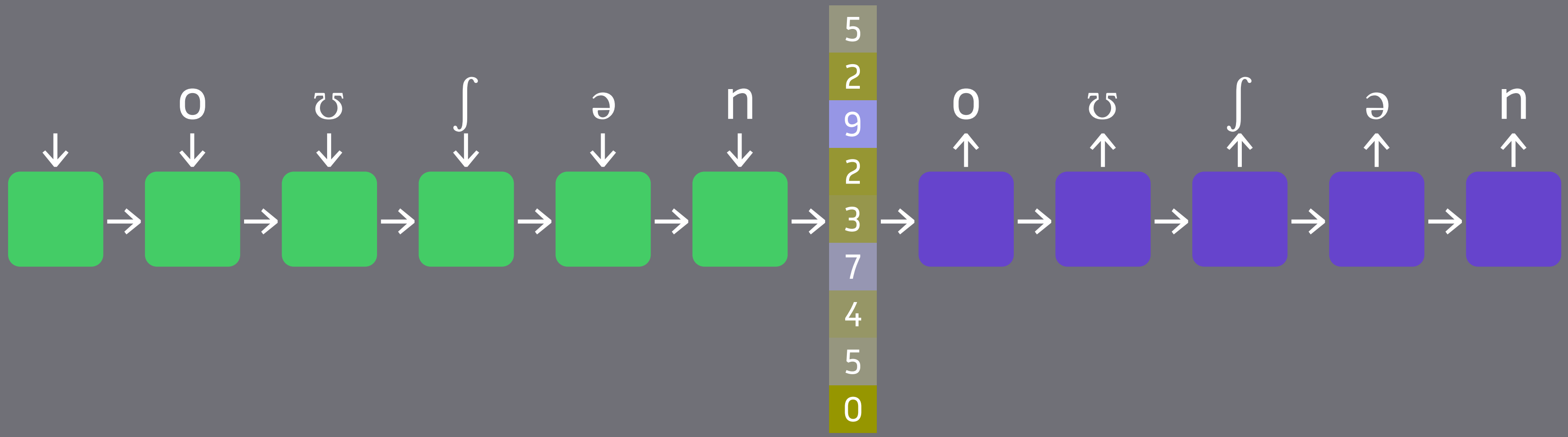
Model 1: Autoencoder



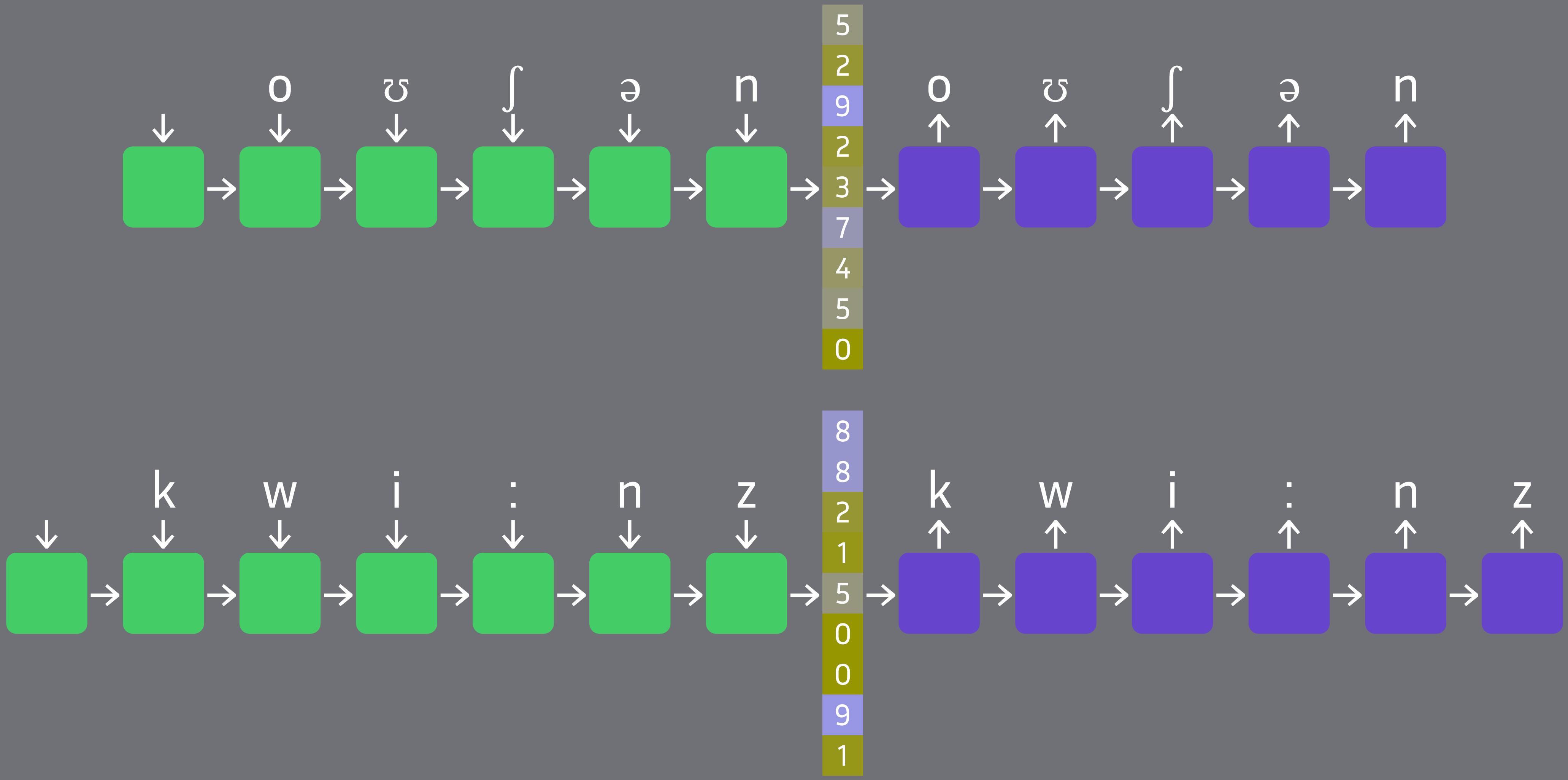
Model 1: Autoencoder



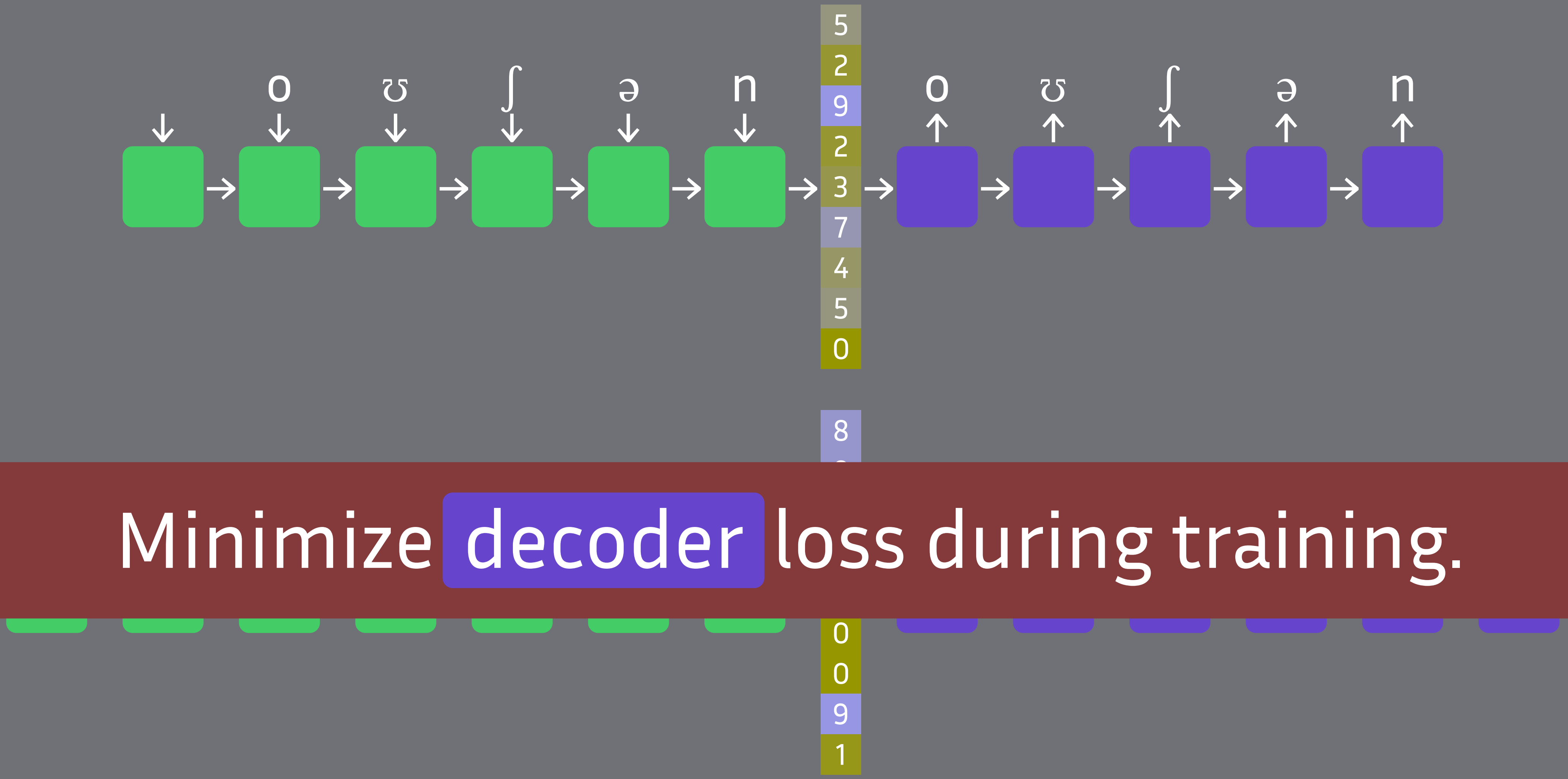
Model 1: Autoencoder



Model 1: Autoencoder

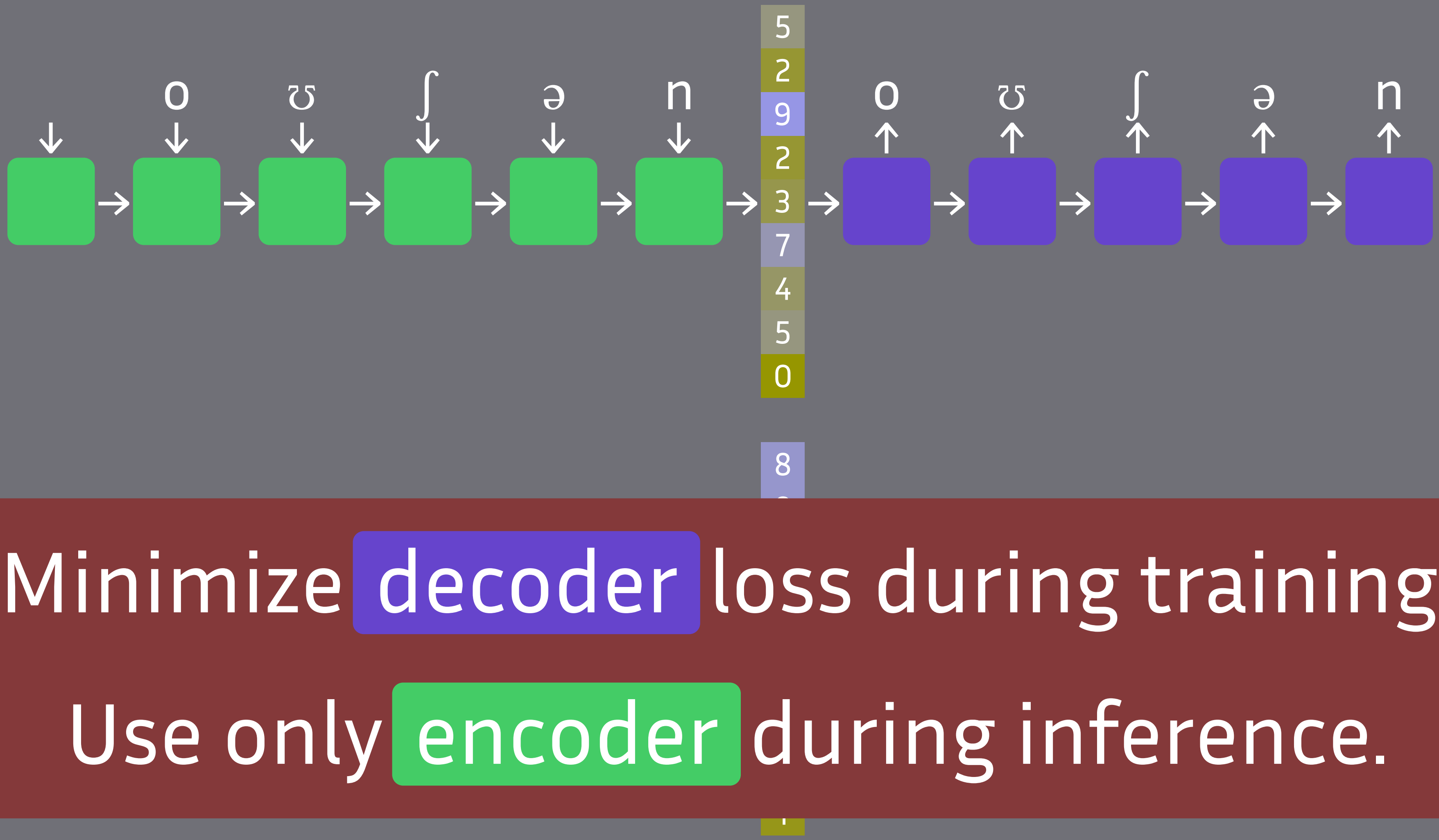


Model 1: Autoencoder



Minimize decoder loss during training.

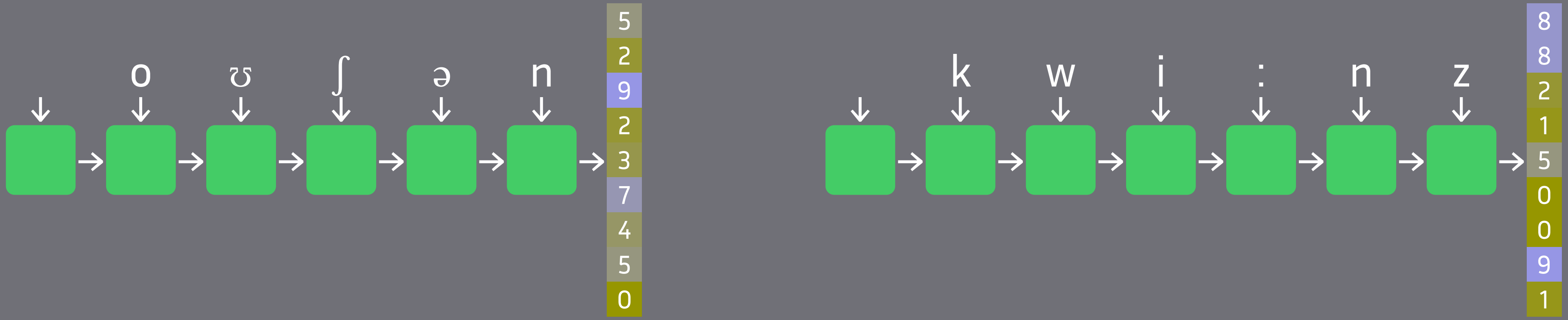
Model 1: Autoencoder



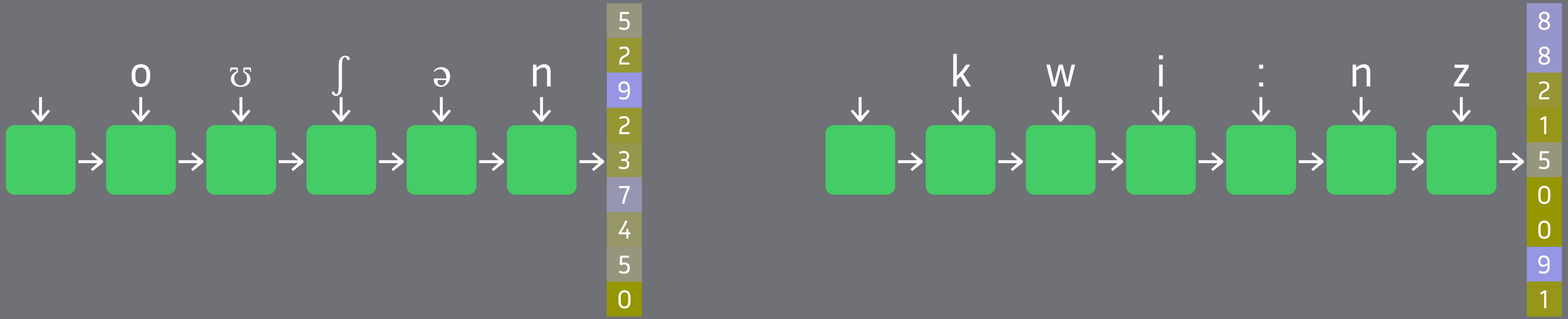
Minimize decoder loss during training.

Use only encoder during inference.

Model 2: Metric Learning



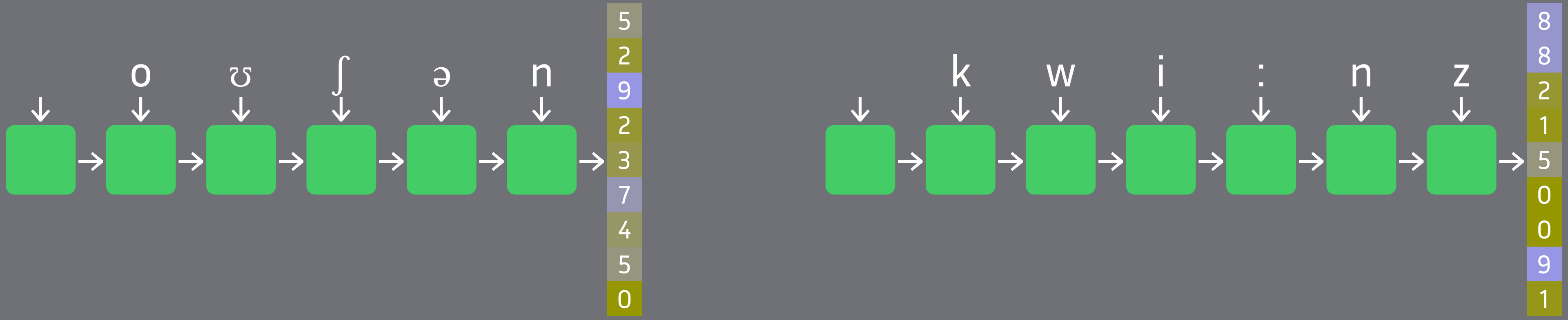
Model 2: Metric Learning



We know $S_H(\text{ocean}, \text{queens})$ and want S_V with f such that

$$S_V(f(\text{ocean}), f(\text{queens})) = S_H(\text{ocean}, \text{queens}))$$

Model 2: Metric Learning



We know $S_H(\text{ocean}, \text{queens})$ and want S_V with f such that

$$S_V(f(\text{ocean}), f(\text{queens})) = S_H(\text{ocean}, \text{queens}))$$

$$\mathcal{L} = \Big\| \, \|f(\text{ocean}) - f(\text{queens})\|_2 \; - \; S_H(\text{ocean}, \text{queens}) \Big\|_2$$

English Hello. How are you?

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?
German	Hallo. Wie geht’s?

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?
German	Hallo. Wie geht's?
Polish	Cześć. Jak się masz?

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?
German	Hallo. Wie geht’s?
Polish	Cześć. Jak się masz?
Spanish	Hola. ¿Cómo estás?

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?
German	Hallo. Wie geht’s?
Polish	Cześć. Jak się masz?
Spanish	Hola. ¿Cómo estás?
Swahili	Habari. Habari yako?

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?
German	Hallo. Wie geht’s?
Polish	Cześć. Jak się masz?
Spanish	Hola. ¿Cómo estás?
Swahili	Habari. Habari yako?
Uzbek	Salom. Qalaysiz?

PWESuite

English	Hello. How are you?
Amharic	ሀሎ. ስላም? (hālo. silami?)
Bengali	হ্যালো. আপনি কেমন আছেন? (Hyālō. Āpani kēmana āchēna?)
French	Bonjour. Comment vas-tu?
German	Hallo. Wie geht’s?
Polish	Cześć. Jak się masz?
Spanish	Hola. ¿Cómo estás?
Swahili	Habari. Habari yako?
Uzbek	Salom. Qalaysiz?

$\sim 200k$ unique tokens

PWESuite Results

	Model	Intrinsic Human Sim. (Pearson)	Art. Dist. (Pearson)	Retrieval (rank perc.)	Extrinsic Analogies (Acc@1)	Rhyme (accuracy)	Cognate (accuracy)	Overall
Ours	Metric Learner	0.46	0.94	0.98	84%	83%	64%	0.78
	Triplet Margin	0.65	0.96	1.00	100%	77%	66%	0.84 ⋆
	Count-based	0.82	0.10	0.84	13%	79%	68%	0.56
	Autoencoder	0.49	0.16	0.73	50%	61%	50%	0.50
Others'	Poetic Sound Sim.	0.74	0.12	0.78	35%	60%	57%	0.53
	phoneme2vec	0.77	0.09	0.80	17%	88%	64%	0.56
	Phon. Sim. Embd.	0.16	0.05	0.50	0%	51%	52%	0.29
Semantic	BPEmb	0.23	0.08	0.60	5%	54%	66%	0.36
	fastText	0.25	0.12	0.64	2%	58%	68%	0.38
	BERT	0.10	0.34	0.69	4%	58%	63%	0.40
	INSTRUCTOR	0.60	0.12	0.73	7%	54%	66%	0.45

PWESuite Results

	Model	Intrinsic Human Sim. (Pearson)	Art. Dist. (Pearson)	Retrieval (rank perc.)	Extrinsic Analogies (Acc@1)	Rhyme (accuracy)	Cognate (accuracy)	Overall
Ours	Metric Learner	0.46	0.94	0.98	84%	83%	64%	0.78
	Triplet Margin	0.65	0.96	1.00	100%	77%	66%	0.84 ⋆
	Count-based	0.82	0.10	0.84	13%	79%	68%	0.56
	Autoencoder	0.49	0.16	0.73	50%	61%	50%	0.50
Others'	Poetic Sound Sim.	0.74	0.12	0.78	35%	60%	57%	0.53
	phoneme2vec	0.77	0.09	0.80	17%	88%	64%	0.56
	Phon. Sim. Embd.	0.16	0.05	0.50	0%	51%	52%	0.29
Semantic	BPEmb	0.23	0.08	0.60	5%	54%	66%	0.36
	fastText	0.25	0.12	0.64	2%	58%	68%	0.38
	BERT	0.10	0.34	0.69	4%	58%	63%	0.40
	INSTRUCTOR	0.60	0.12	0.73	7%	54%	66%	0.45

PWESuite Results

	Model	Intrinsic Human Sim. (Pearson)	 Art. Dist. (Pearson)	 Retrieval (rank perc.)	Extrinsic Analogies (Acc@1)	 Rhyme (accuracy)	 Cognate (accuracy)	Overall
Ours	Metric Learner	0.46	0.94	0.98	84%	83%	64%	0.78
	Triplet Margin	0.65	0.96	1.00	100%	77%	66%	0.84 ⋆
	Count-based	0.82	0.10	0.84	13%	79%	68%	0.56
	Autoencoder	0.49	0.16	0.73	50%	61%	50%	0.50
Others'	Poetic Sound Sim.	0.74	0.12	0.78	35%	60%	57%	0.53
	phoneme2vec	0.77	0.09	0.80	17%	88%	64%	0.56
	Phon. Sim. Embd.	[illegible]	[illegible]	[illegible]	0%	51%	52%	0.29
Semantic	BPEmb	0.23	0.08	0.60	5%	54%	66%	0.36
	fastText	0.25	0.12	0.64	2%	58%	68%	0.38
	BERT	0.10	0.34	0.69	4%	58%	63%	0.40
	INSTRUCTOR	0.60	0.12	0.73	7%	54%	66%	0.45

PWESuite Results

	Model	Human Sim. [illegible]	Art. Dist. [illegible]	Retrieval [illegible]	Analogies (Acc@1)	Rhyme (accuracy)	Cognate (accuracy)	OVERALL
		INTRINSIC			EXTRINSIC			
Ours	Metric Learner	0.46	0.94	0.98	84%	83%	64%	0.78
	Triplet Margin	0.65	0.96	1.00	100%	77%	66%	0.84 ⋆
	Count-based	0.82	0.10	0.84	13%	79%	68%	0.56
	Autoencoder	0.49	0.16	0.73	50%	61%	50%	0.50
Others'	Poetic Sound Sim.	[illegible]	[illegible]	[illegible]	35%	60%	57%	0.53
	phoneme2vec	0.77	0.09	0.80	17%	88%	64%	0.56
	Phon. Sim. Embd.	0.16	0.05	0.50	0%	51%	52%	0.29
Semantic	BPEmb	0.23	0.08	0.60	5%	54%	66%	0.36
	fastText	0.25	0.12	0.64	2%	58%	68%	0.38
	BERT	0.10	0.34	0.69	4%	58%	63%	0.40
	INSTRUCTOR	0.60	0.12	0.73	7%	54%	66%	0.45

demo.ipynb

PWESuite: Phonetic Word Embeddings and Tasks They Facilitate

Vilém Zouhar[E=] Kalvin Chang[C=] Chenxuan Cui[C] Nathaniel Carlson[Y]
Nathaniel R. Robinson[C] Mrinmaya Sachan[E] David Mortensen[C]

[E]Department of Computer Science, ETH Zurich
[C]Language Technologies Institute, Carnegie Mellon University
[Y]Department of Computer Science, Brigham Young University
{vzouhar,msachan}@ethz.ch natbcar@gmail.com
{kalvinc,cxcui,nrrobins,dmortens}@cs.cmu.edu

Abstract

Mapping words into a fixed-dimensional vector space is the backbone of modern NLP. While most word embedding methods successfully encode semantic information, they overlook phonetic information that is crucial for many tasks. We develop three methods that use articulatory features to build phonetically informed word embeddings. To address the inconsistent evaluation of existing phonetic word embedding methods, we also contribute a task suite to fairly evaluate past, current, and future methods. We evaluate both (1) intrinsic aspects of phonetic word embeddings, such as word retrieval and correlation with sound similarity, and (2) extrinsic performance on tasks such as rhyme and cognate detection and sound analogies. We hope our task suite will promote reproducibility and inspire future phonetic embedding research.

Keywords: phonetic word embeddings, representation learning, phonology, articulatory features, evaluation

Code: github.com/zouharvi/pwesuite
Dataset: huggingface.co/datasets/zouharvi/pwesuite-eval

1. Introduction

Word embeddings are omnipresent in modern NLP (Le and Mikolov, 2014; Pennington et al., 2014; Almeida and Xexéo, 2019, inter alia). Their main benefit lies in compressing some information into fixed-dimensional vectors. These vectors can be used as machine-learning features for NLP applications, and their study can reveal linguistic insights (Hamilton et al., 2016; Ryskina et al., 2020; Francis et al., 2021). Word embeddings are often trained via methods from distributional semantics (Camacho-Collados and Pilehvar, 2018) and thus bear semantic information. For example, the embedding for the word *carrot* may encode higher similarity to embeddings for other vegetables than to that of *ocean*.

Some applications may require a different type of information to be encoded. The orthography, especially in English, can obscure the pronunciation. A poem generation model, for instance, may need embeddings to reflect that *ocean* rhymes with *motion* and not with a *soybean*, even though the spelling of the words' final syllables suggest otherwise (see Figure 1). Such embeddings, called *phonetic word embeddings*, contain phonetic information and have been of recent interest (Parrish, 2017; Yang and Hirschberg, 2019; Hu et al., 2020; Sharma et al., 2021).[1] The objective is that words with similar pronunciation should be mapped to vectors near each other in embedding space. Many tasks have benefited from incorporating phonetic word embeddings, including cognate and loanword detection (Rama, 2016; Nath et al., 2022b,a), named entity recognition (Bharadwaj et al., 2016; Chaudhary et al., 2018), spelling correction (Zhang et al., 2021), and speech recog-

Figure 1: Embedding function *f* projects words in various forms (left) to a vector space (right) such that words with a similar pronunciation (e.g., *ocean* and *motion*) are closer than words with a dissimilar pronunciation (e.g., *ocean* and *soybean*).

=Co-first authors.

[1]The technically correct term is *phonological word embeddings* but prior literature uses the term *phonetic*.

- unified evaluation suite (6 tasks)
- phonetic word embeddings (4 models)

PWESUITE: Phonetic Word Embeddings and Tasks They Facilitate

Vilém Zouhar[E=] Kalvin Chang[C=] Chenxuan Cui[C] Nathaniel Carlson[Y]
Nathaniel R. Robinson[C] Mrinmaya Sachan[E] David Mortensen[C]

[E]Department of Computer Science, ETH Zurich
[C]Language Technologies Institute, Carnegie Mellon University
[Y]Department of Computer Science, Brigham Young University
{vzouhar,msachan}@ethz.ch natbcar@gmail.com
{kalvinc,cxcui,nrrobins,dmortens}@cs.cmu.edu

Abstract

Mapping words into a fixed-dimensional vector space is the backbone of modern NLP. While most word embedding methods successfully encode semantic information, they overlook phonetic information that is crucial for many tasks. We develop three methods that use articulatory features to build phonetically informed word embeddings. To address the inconsistent evaluation of existing phonetic word embedding methods, we also contribute a task suite to fairly evaluate past, current, and future methods. We evaluate both (1) intrinsic aspects of phonetic word embeddings, such as word retrieval and correlation with sound similarity, and (2) extrinsic performance on tasks such as rhyme and cognate detection and sound analogies. We hope our task suite will promote reproducibility and inspire future phonetic embedding research.

Keywords: phonetic word embeddings, representation learning, phonology, articulatory features, evaluation

Code: github.com/zouharvi/pwesuite
Dataset: huggingface.co/datasets/zouharvi/pwesuite-eval

1. Introduction

Word embeddings are omnipresent in modern NLP (Le and Mikolov, 2014; Pennington et al., 2014; Almeida and Xexéo, 2019, inter alia). Their main benefit lies in compressing some information into fixed-dimensional vectors. These vectors can be used as machine-learning features for NLP applications, and their study can reveal linguistic insights (Hamilton et al., 2016; Ryskina et al., 2020; Francis et al., 2021). Word embeddings are often trained via methods from distributional semantics (Camacho-Collados and Pilehvar, 2018) and thus bear semantic information. For example, the embedding for the word *carrot* may encode higher similarity to embeddings for other vegetables than to that of *ocean*.

Some applications may require a different type of information to be encoded. The orthography, especially in English, can obscure the pronunciation. A poem generation model, for instance, may need embeddings to reflect that *ocean* rhymes with *motion* and not with a *soybean*, even though the spelling of the words' final syllables suggest otherwise (see Figure 1). Such embeddings, called *phonetic word embeddings*, contain phonetic information and have been of recent interest (Parrish, 2017; Yang and Hirschberg, 2019; Hu et al., 2020; Sharma et al., 2021).[1] The objective is that words with similar pronunciation should be mapped to vectors near each other in embedding space. Many tasks have benefited from incorporating phonetic word embeddings, including cognate and loanword detection (Rama, 2016; Nath et al., 2022b,a), named entity recognition (Bharadwaj et al., 2016; Chaudhary et al., 2018), spelling correction (Zhang et al., 2021), and speech recog-

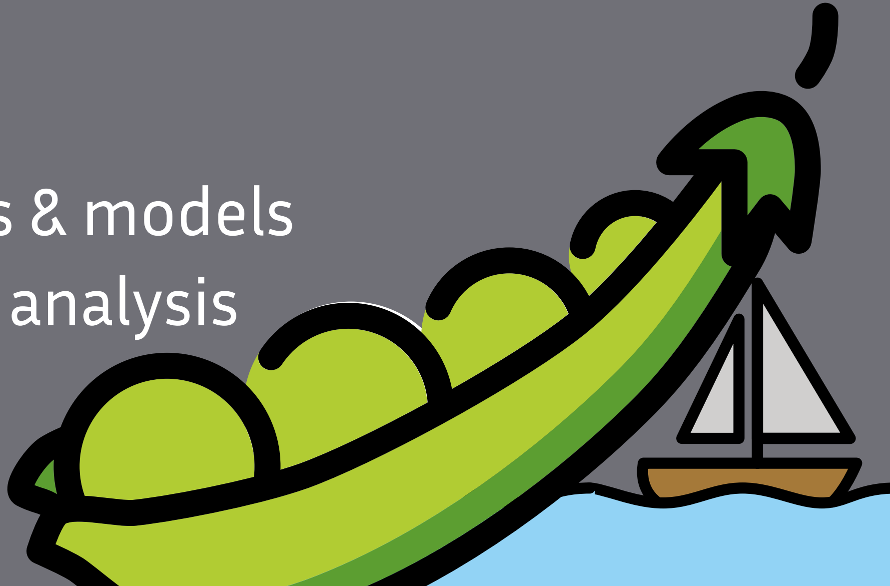
Figure 1: Embedding function *f* projects words in various forms (left) to a vector space (right) such that words with a similar pronunciation (e.g., *ocean* and *motion*) are closer than words with a dissimilar pronunciation (e.g., *ocean* and *soybean*).

[1]The technically correct term is *phonological word embeddings* but prior literature uses the term *phonetic*.

[=]Co-first authors.

- unified evaluation suite (6 tasks)
- phonetic word embeddings (4 models)

- new tasks & models
- linguistic analysis



PWESuite: Phonetic Word Embeddings and Tasks They Facilitate

Vilém Zouhar[E=] Kalvin Chang[C=] Chenxuan Cui[C] Nathaniel Carlson[Y]
Nathaniel R. Robinson[C] Mrinmaya Sachan[E] David Mortensen[C]

[E]Department of Computer Science, ETH Zurich
[C]Language Technologies Institute, Carnegie Mellon University
[Y]Department of Computer Science, Brigham Young University
{vzouhar,msachan}@ethz.ch natbcar@gmail.com
{kalvinc,cxcui,nrrobins,dmortens}@cs.cmu.edu

Abstract

Mapping words into a fixed-dimensional vector space is the backbone of modern NLP. While most word embedding methods successfully encode semantic information, they overlook phonetic information that is crucial for many tasks. We develop three methods that use articulatory features to build phonetically informed word embeddings. To address the inconsistent evaluation of existing phonetic word embedding methods, we also contribute a task suite to fairly evaluate past, current, and future methods. We evaluate both (1) intrinsic aspects of phonetic word embeddings, such as word retrieval and correlation with sound similarity, and (2) extrinsic performance on tasks such as rhyme and cognate detection and sound analogies. We hope our task suite will promote reproducibility and inspire future phonetic embedding research.

Keywords: phonetic word embeddings, representation learning, phonology, articulatory features, evaluation

Code: github.com/zouharvi/pwesuite
Dataset: huggingface.co/datasets/zouharvi/pwesuite-eval

1. Introduction

Word embeddings are omnipresent in modern NLP (Le and Mikolov, 2014; Pennington et al., 2014; Almeida and Xexéo, 2019, inter alia). Their main benefit lies in compressing some information into fixed-dimensional vectors. These vectors can be used as machine-learning features for NLP applications, and their study can reveal linguistic insights (Hamilton et al., 2016; Ryskina et al., 2020; Francis et al., 2021). Word embeddings are often trained via methods from distributional semantics (Camacho-Collados and Pilehvar, 2018) and thus bear semantic information. For example, the embedding for the word *carrot* may encode higher similarity to embeddings for other vegetables than to that of *ocean.*

Some applications may require a different type of information to be encoded. The orthography, especially in English, can obscure the pronunciation. A poem generation model, for instance, may need embeddings to reflect that *ocean* rhymes with *motion* and not with a *soybean*, even though the spelling of the words' final syllables suggest otherwise (see Figure 1). Such embeddings, called *phonetic word embeddings*, contain phonetic information and have been of recent interest (Parrish, 2017; Yang and Hirschberg, 2019; Hu et al., 2020; Sharma et al., 2021).[1] The objective is that words with similar pronunciation should be mapped to vectors near each other in embedding space. Many tasks have benefited from incorporating phonetic word embeddings, including cognate and loanword detection (Rama, 2016; Nath et al., 2022b,a), named entity recognition (Bharadwaj et al., 2016; Chaudhary et al., 2018), spelling correction (Zhang et al., 2021), and speech recog-

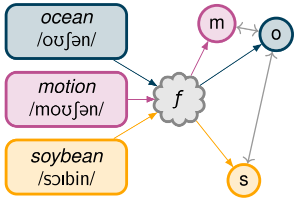


Figure 1: Embedding function *f* projects words in various forms (left) to a vector space (right) such that words with a similar pronunciation (e.g., *ocean* and *motion*) are closer than words with a dissimilar pronunciation (e.g., *ocean* and *soybean*).

=Co-first authors.

[1]The technically correct term is *phonological word embeddings* but prior literature uses the term *phonetic*.

Thank you

- unified evaluation suite (6 tasks)
- phonetic word embeddings (4 models)

- new tasks & models
- linguistic analysis