Reconstructing language ancestry by performing word prediction

Peter Dekker
University of Amsterdam
peter@peterdekker.eu

March 29, 2017

Workshop *Phylogenetic methods in historical linguistics*
Tübingen, March 27-30, 2017

Introduction

Introduction

- MSc student Artificial Intelligence, University of Amsterdam
- MSc thesis, supervised by:
  - Gerhard Jäger, SfS, University of Tübingen
  - Jelle Zuidema, ILLC, University of Amsterdam

Overview

Research question

- How are languages related?
- Which sounds change regularly between languages?
- Many methods depend on manual cognate judgments
- Method: word prediction, based on phonetic basic vocabulary lists
    - Predict words in language *A* from words in language *B*
    - Use regularity of sound change
- Serves as basis for:
    - Phylogenetic tree reconstruction, without manual cognate judgments
    - Identification of sound correspondences

Related work

- Cognate production: Beinborn et al. (2013); Ciobanu (2016)
  - Only on cognates, orthograpic input
  - Our method: both cognates and non-cognates, phonetic input
- Cognate detection: Inkpen et al. (2005); List (2012); Jäger et al. (2017); Rama (2016)
  - Related task, but binary classification instead of sequence generation

Method

Method overview

- Word prediction algorithm
- Applications:
  - Phylogenetic tree reconstruction
  - Identification of sound correspondences

Method

Word prediction

Machine learning

- Use machine learning paradigm
- Supervised learning
  - Train a model on pairs (x, y)
  - Predict y for an unseen x
  - Eg. image classification
- Linguistic motivation
  - Regular sound change is predictable

Word prediction

1. Train a model on pairs $(w_{c,A}, w_{c,B})$: words for concept c in languages A and B

Word prediction

1. Train a model on pairs ($w_{c,A}$, $w_{c,B}$): words for concept c in languages A and B

NL	DE
ɣeːst	gaɪst
nɛt	nɛt͡s
ɑprɪl	ʔapʁiːl
vaːdə	faːtɐ
kɔrt	kʊʁt͡s

Word prediction

1. Train a model on pairs ($w_{c,A}$, $w_{c,B}$): words for concept c in languages A and B
2. For a new concept d, give $w_{d,A}$ and predict $w_{d,B}$

NL	DE
ɣeːst	gaɪst
nɛt	nɛt͡s
ɑprɪl	ʔapʁiːl
vaːdə	faːtɐ
kɔrt	kʊʁt͡s

Word prediction

1. Train a model on pairs ($w_{c,A}$, $w_{c,B}$): words for concept c in languages A and B
2. For a new concept d, give $w_{d,A}$ and predict $w_{d,B}$

NL	DE
ɣeːst	gaɪst
nɛt	nɛt͡s
ɑprɪl	ʔapʁiːl
vaːdə	faːtɐ
kɔrt	kʊʁt͡s
vreːmt	

Word prediction

1. Train a model on pairs ($w_{c,A}$, $w_{c,B}$): words for concept c in languages A and B
2. For a new concept d, give $w_{d,A}$ and predict $w_{d,B}$

NL	DE
ɣeːst	gaɪst
nɛt	nɛt͡s
ɑprɪl	ʔapʁiːl
vaːdə	faːtɐ
kɔrt	kʊʁt͡s
vreːmt	**fʁɛmt**

Word prediction

1. Train a model on pairs ($w_{c,A}$, $w_{c,B}$): words for concept c in languages A and B
2. For a new concept d, give $w_{d,A}$ and predict $w_{d,B}$

NL	DE
ɣeːst	gaɪst
nɛt	nɛt͡s
ɑprɪl	ʔapʁiːl
vaːdə	faːtɐ
kɔrt	kʊʁt͡s
vreːmt	**fʁɛmt**

- Analogy to machine translation (Kondrak, 2002)

Word prediction

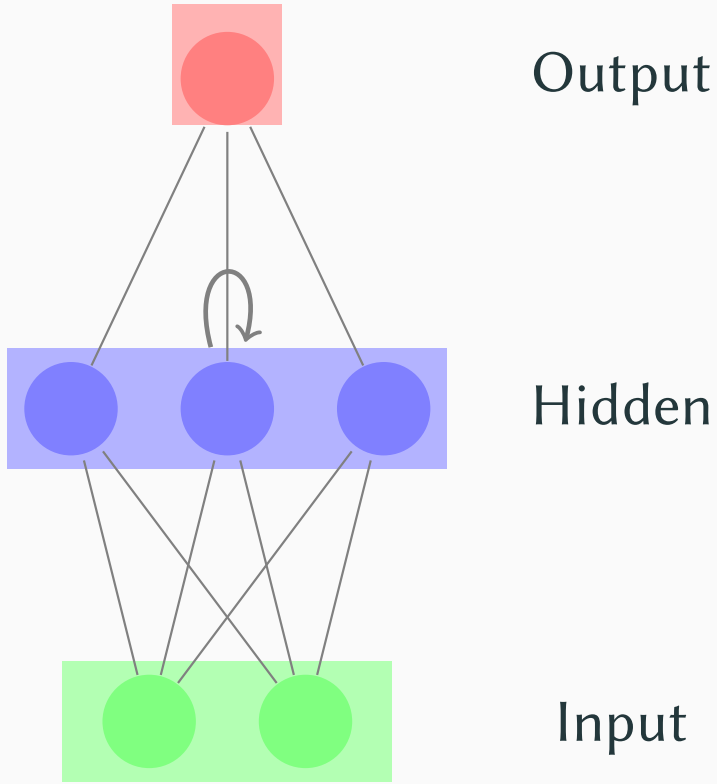
1. Train a model on pairs ($w_{c,A}$, $w_{c,B}$): words for concept c in languages A and B
2. For a new concept d, give $w_{d,A}$ and predict $w_{d,B}$

NL	DE
ɣeːst	gaɪst
nɛt	nɛt͡s
ɑprɪl	ʔapʁiːl
vaːdə	faːtɐ
kɔrt	kʊʁt͡s
vreːmt	**fʁɛmt**

- Analogy to machine translation (Kondrak, 2002)
- Learn to detect cognates, partial cognates and loanwords

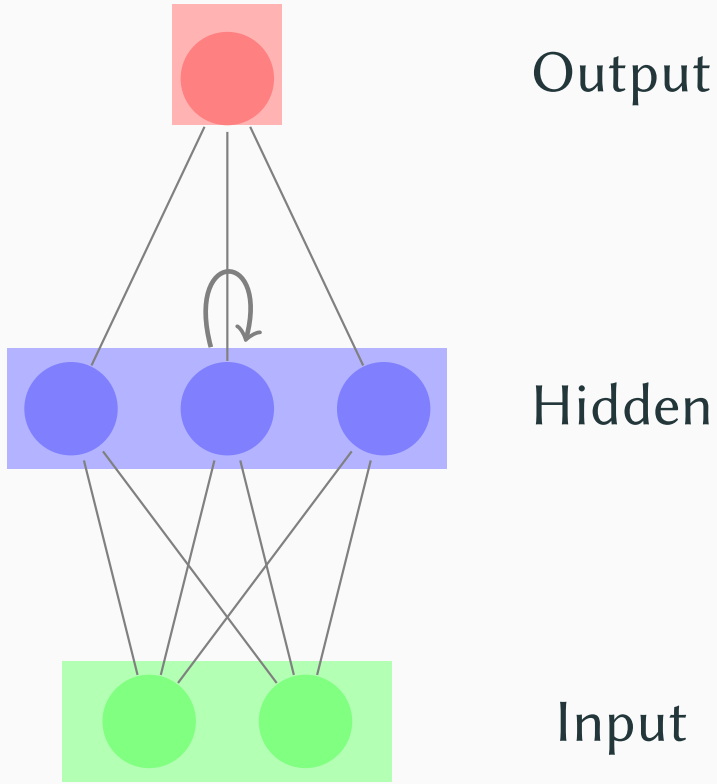
Recurrent neural network

- Neural network that models sequential data



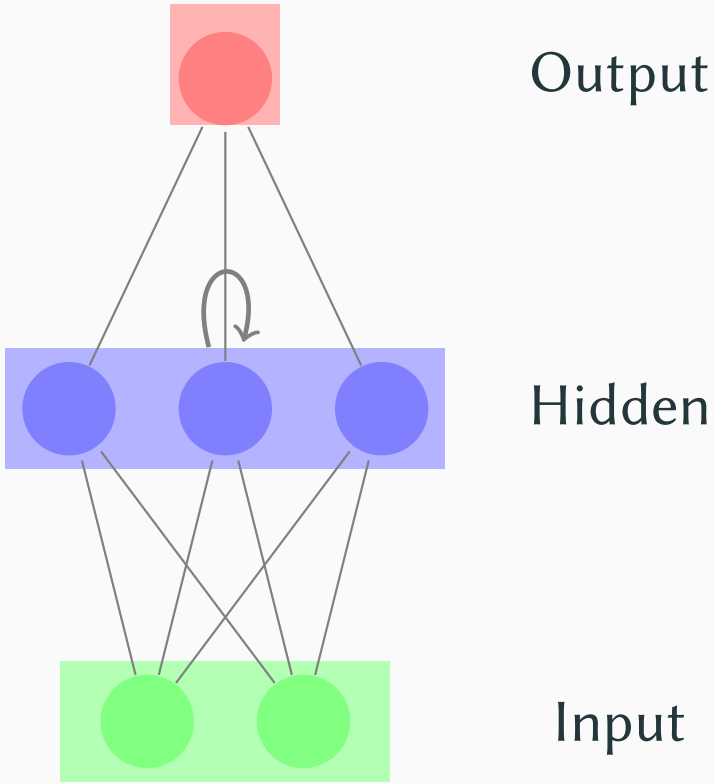
Recurrent neural network

- Neural network that models sequential data
- Recurrent connections
- Weights (=information) are shared between time steps



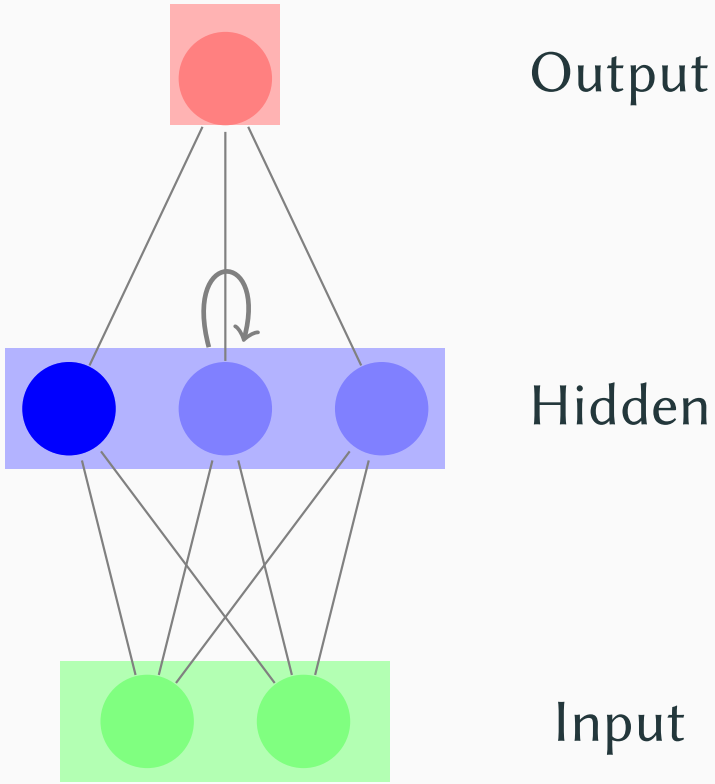
Recurrent neural network

- Neural network that models sequential data
- Recurrent connections
- Weights (=information) are shared between time steps
- Unfold one node over time:



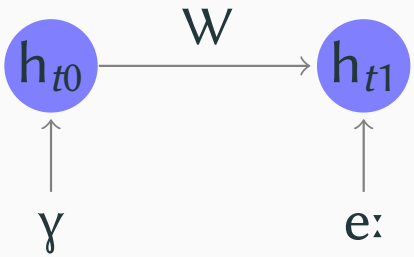
Recurrent neural network

- Neural network that models sequential data
- Recurrent connections
- Weights (=information) are shared between time steps
- Unfold one node over time:

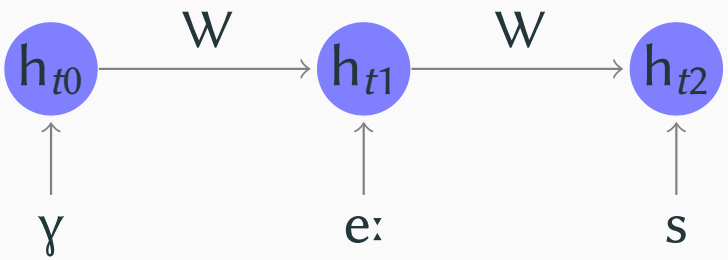


Recurrent Neural Network (RNN)

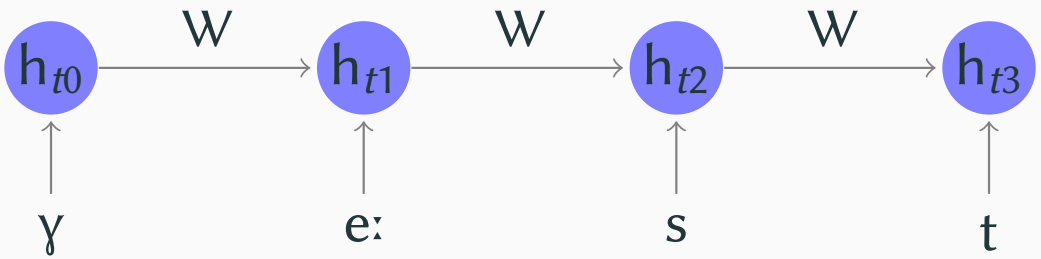
Recurrent Neural Network (RNN)



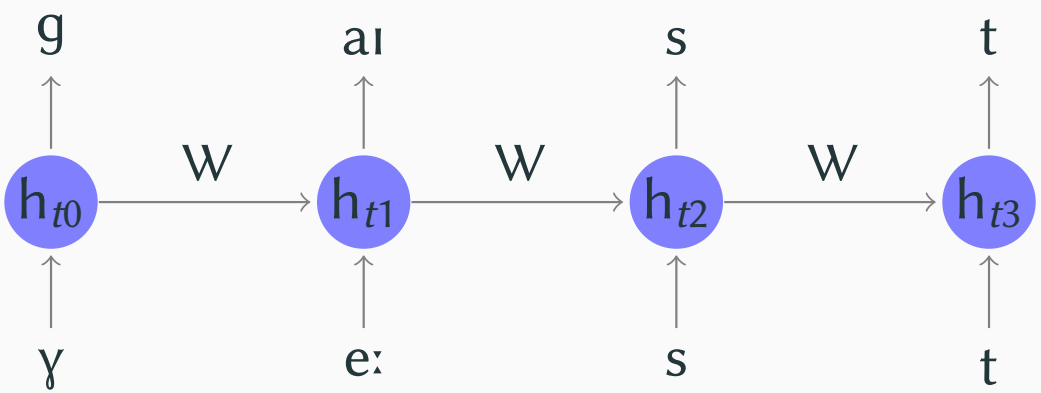
Recurrent Neural Network (RNN)



Recurrent Neural Network (RNN)

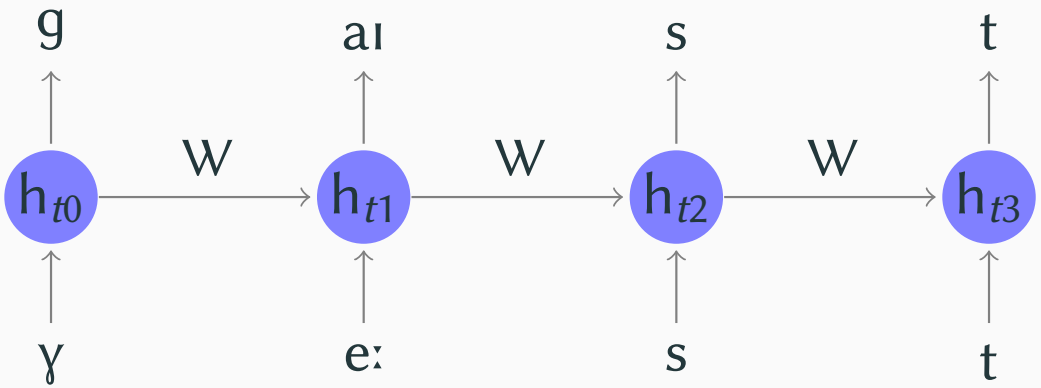


Recurrent Neural Network (RNN)



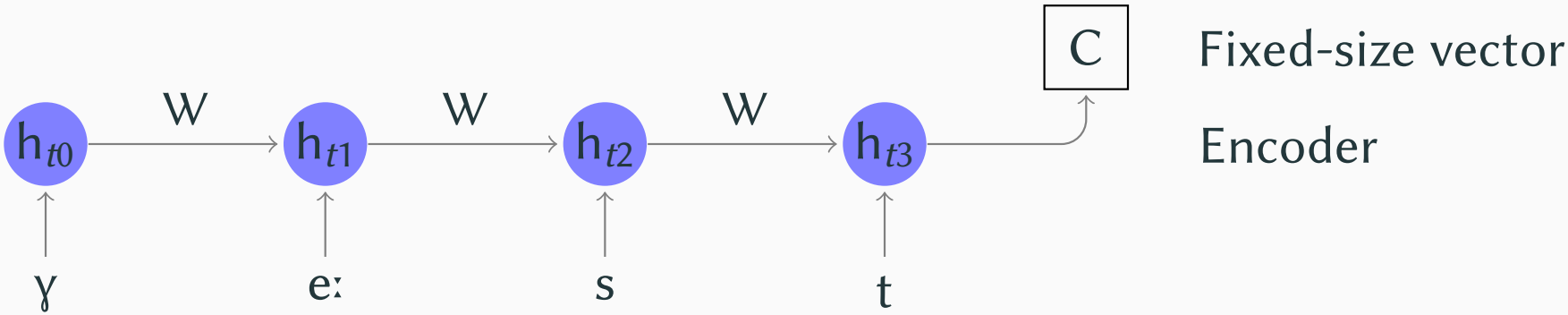
Recurrent Neural Network (RNN)

- Direct output of encoder: assumes same input and output length. Solution:
- Encoder-decoder structure
  - Successful in machine translation: Cho et al. (2014), Sutskever et al. (2014)



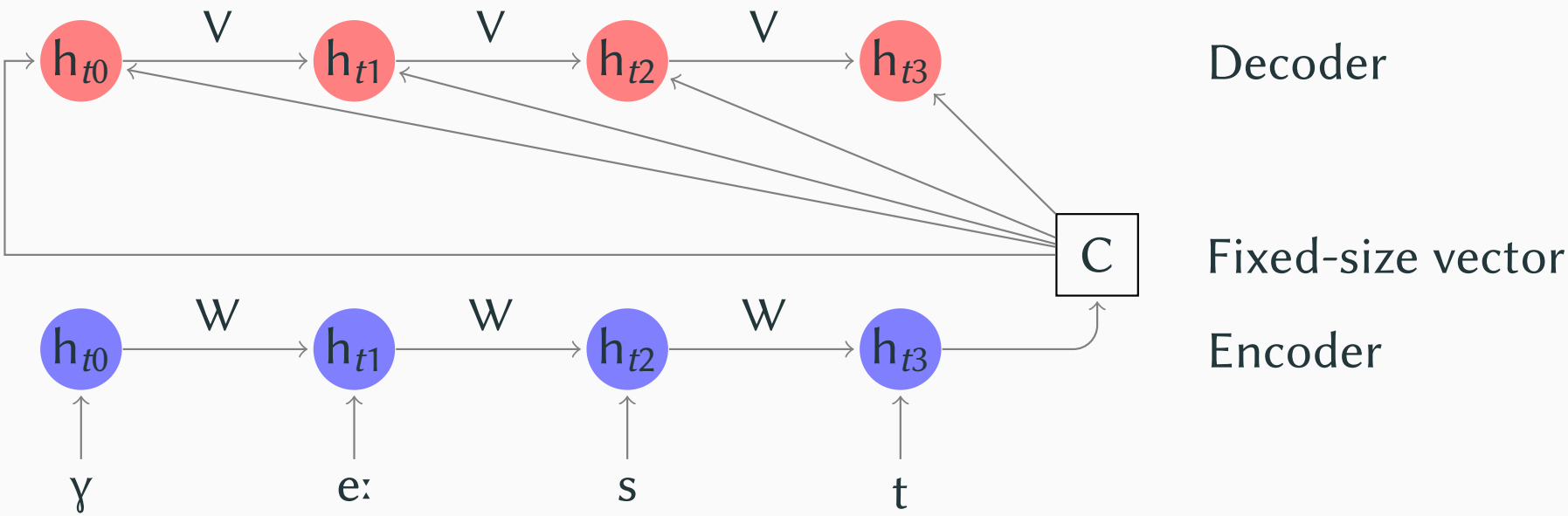
Recurrent Neural Network (RNN)

- Encoder-decoder structure
  - Successful in machine translation: Cho et al. (2014), Sutskever et al. (2014)



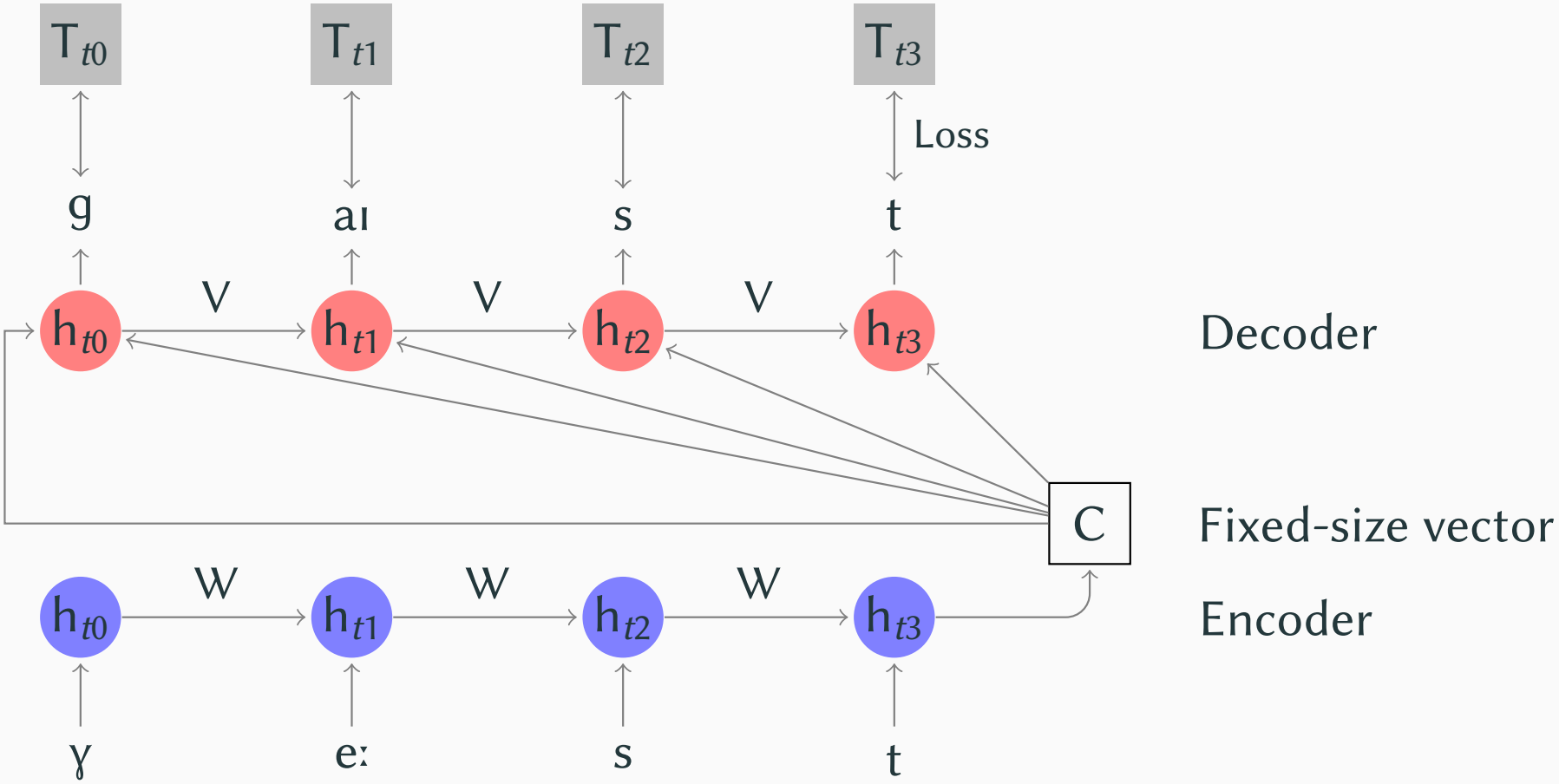
Recurrent Neural Network (RNN)

Our model



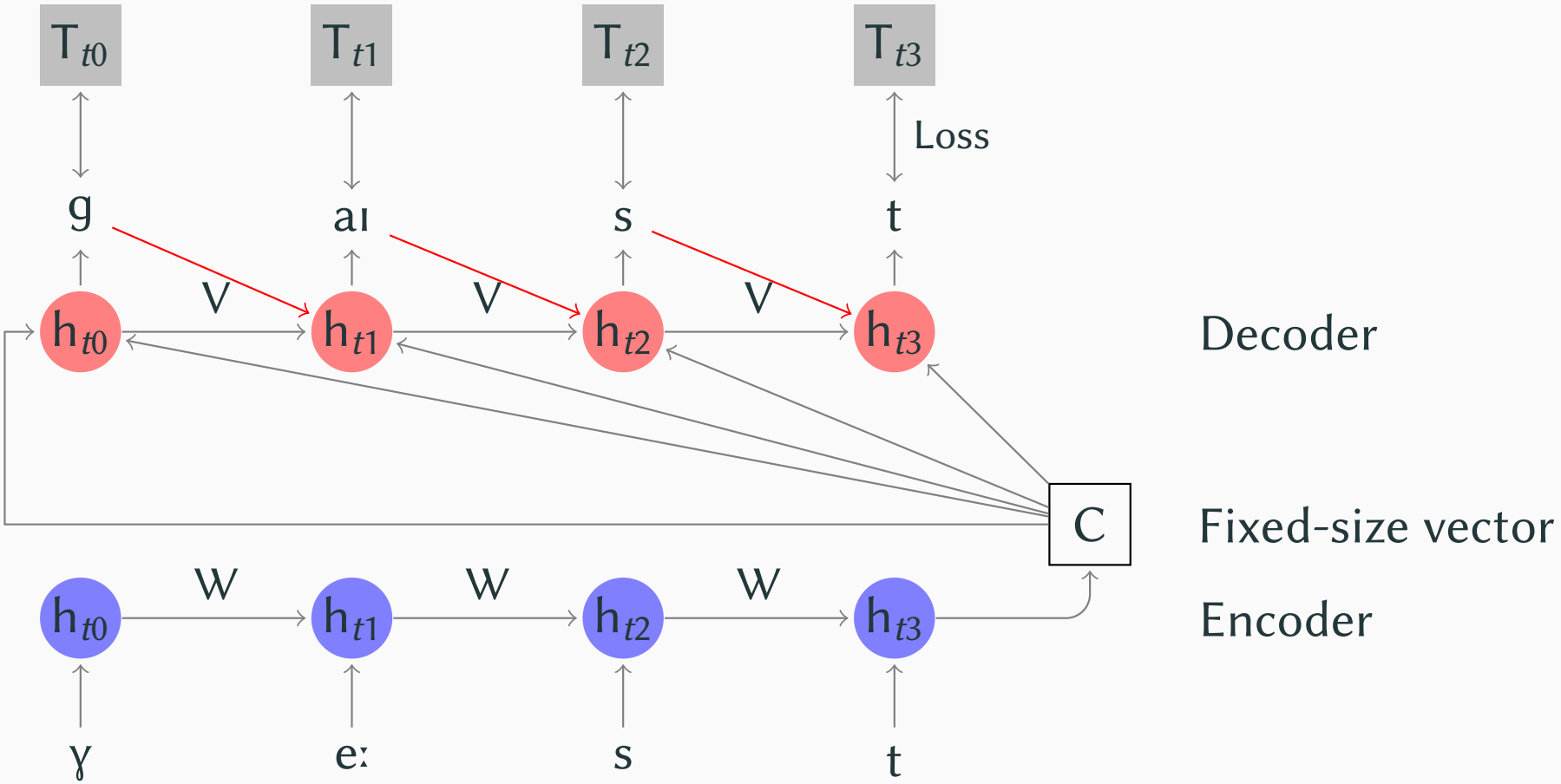
Recurrent Neural Network (RNN)

Our model



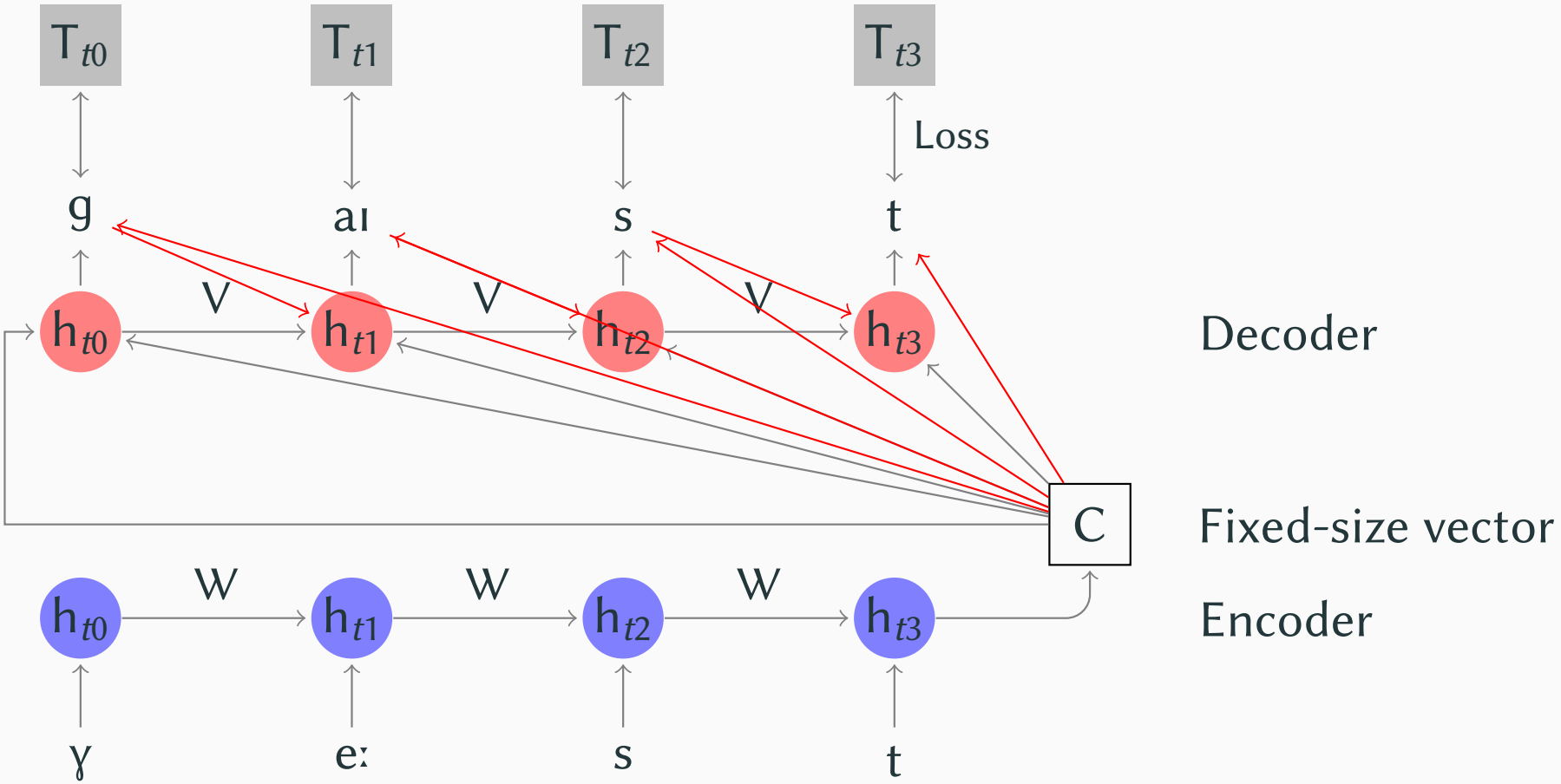
Recurrent Neural Network (RNN)

Sequence-to-sequence in Cho et al. (2014), not in our approach



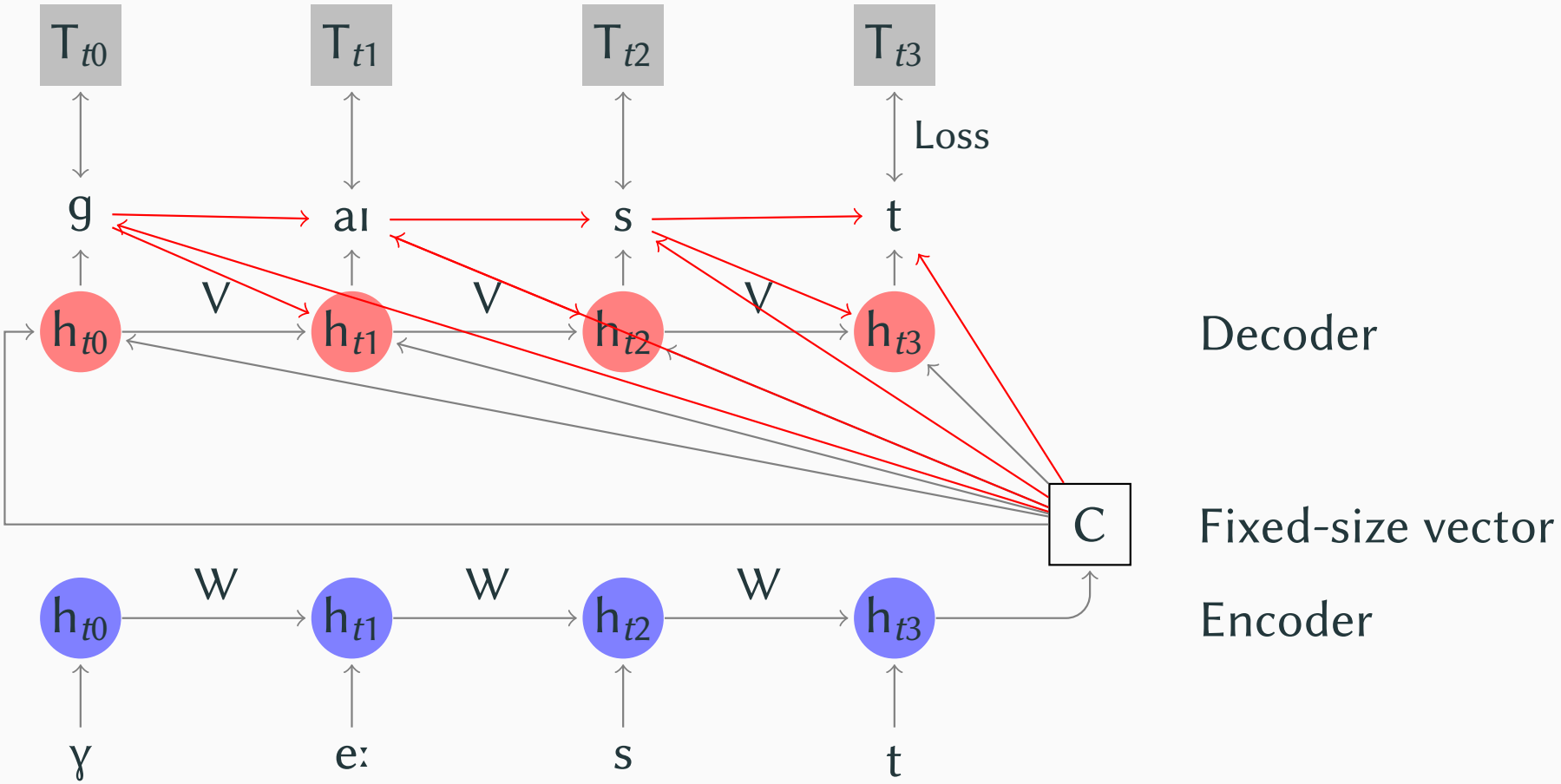
Recurrent Neural Network (RNN)

Sequence-to-sequence in Cho et al. (2014), not in our approach



Recurrent Neural Network (RNN)

Sequence-to-sequence in Cho et al. (2014), not in our approach



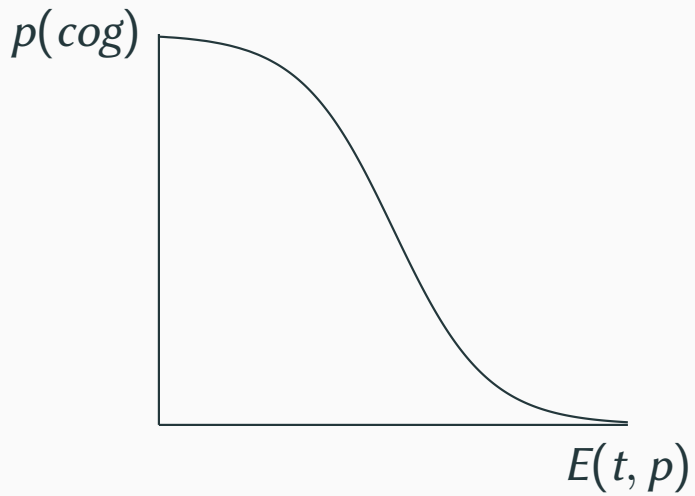
Loss function

- Network learns from the *loss* between target and prediction
- General loss function: cross-entropy
- *Cognacy prior*: learn more from probable cognates, less from non-cognates
    - Error between target and prediction, with sharp decline if error exceeds θ
    - θ based on mean error in training history

$$L = CE(t, p) \cdot p(cog)$$

$$p(cog) = \frac{1}{1 + e^{E(t,p)-\theta}}$$

$$\theta = \overline{E}_{history} + v\sigma$$



Model details

- 400 hidden units for encoder and decoder
- Gated Recurrent Units (GRU) (Cho et al., 2014) as recurrent nodes: remember long-distance dependencies
- Bidirectional encoder
- Encoder input is read until exact word length
- Decoder input has standard length, empty positions encoded by special character
- Implemented using *Lasagne* framework, based on *Theano*

Input format

- Input alphabet:
  - Phonetic representation of words: usually IPA
  - **ASJPcode**: 41 sound classes, many-to-one from IPA to ASJP (Brown et al., 2008)
- Input encoding:

Input format

- Input alphabet:
  - Phonetic representation of words: usually IPA
  - **ASJPcode**: 41 sound classes, many-to-one from IPA to ASJP (Brown et al., 2008)
- Input encoding:
  - **One-hot character encoding**:
    - Vector of length $n_{characters}$, with 1 at right character, 0 for all other characters
    - Single-label classification: softmax output layer, categorical cross-entropy loss

p	1	0	0	0	0
b	0	0	0	1	0

Input format

- Input alphabet:
  - Phonetic representation of words: usually IPA
  - **ASJPcode**: 41 sound classes, many-to-one from IPA to ASJP (Brown et al., 2008)
- Input encoding:
  - **One-hot character encoding**:
  - Phonetic encoding:
    - Vector of length $n_{features}$, with 1 at every feature that applies
    - Multi-label classification: sigmoid output layer, binary cross-entropy loss

	Voiced	Labial	Dental	...
p	0	1	0	...
b	1	1	0	...

Method

Phylogenetic tree reconstruction

Reconstructing a phylogenetic tree

- Intuition: performance on prediction corresponds to genetic relationship between languages
  - Cognate pairs are predictable through regular sound correspondences
  - Language pairs with high prediction score share more cognates
- Hierachical clustering of languages based on edit distance matrix between *target* and *prediction*
- For a language pair, distance is mean of distances in both directions
- UPGMA (Sokal and Michener, 1958), neighbor joining (Saitou and Nei, 1987) or other phylogenetic algorithm
- Baseline: clustering based on edit distances between *source* and *target*

Method

Identification of sound correspondences

Identification of sound correspondences

- Which sound correspondences occur regularly, in which contexts?
- Perform Needleman-Wunsch alignment (Needleman and Wunsch, 1970) between source →target and source→prediction
- Look at frequencies of substitutions

Results

Results

Word prediction

Word prediction

- NorthEuraLex dataset (Dellert, 2015)
  - Indo-European portion 1016 concepts for 31 languages (still unpublished)
- Settings in following experiments:
  - 15 epochs over training set of 800
  - Test set: 100

Word prediction: NL-DE

Source	Target	Prediction
rur3	GiG3n	GGG33n
vorst	fGost	fuGst
vErbet3r3	fEabEsan	fEaaeG3nn
Eiverix	flaisiS	EegiitS
zom3r	zoma	zum
3itnod3x3	ainlad3n	Eidsk3n
spor	Spua	SaaG3
xras	gGas	gGas
ent	Ent3	aint
sprek3	SpGES3n	S3EG3nn
blEiv3	blaib3n	blib3nn
Distance		0.54

Word prediction: EN-FR

Source	Target	Prediction
spred	apate	prE
stik	bato	afose
haumoC	kobyE	mumi
wil3u	sol	aaa
3j	puse	arot
bend	kurbe	bE
mir3	glas	mrry
swon	siN	s3
teibl	tabl3	traoe
fo	katr3	poaS
be3	urs	bE
8en	pyi	lE
Distance		0.87

Prediction performance

Baseline: source prediction and prediction of sounds using PMI (Jäger et al., 2017; Church and Hanks, 1990; Wieling et al., 2009)

Lang pair		Distance		
		Prediction	Source	PMI
ES	IT	0.55	0.51	**0.50**
DE	NL	0.56	0.62	**0.54**
NL	DE	0.58	0.62	**0.51**
IT	ES	0.59	**0.51**	0.54
FR	IT	**0.63**	0.72	0.65
CZ	RU	0.67	**0.60**	0.57
CZ	PL	0.68	0.62	**0.58**
FR	ES	0.68	0.76	**0.67**
PL	CZ	0.68	0.62	**0.52**
PL	RU	0.69	0.68	**0.60**

Lang pair		Distance		
		Prediction	Source	PMI
ES	EN	**0.84**	0.87	0.98
DE	PL	**0.84**	0.91	0.87
RU	FR	**0.85**	0.90	0.96
CZ	FR	**0.85**	0.88	0.95
EN	PL	**0.85**	0.89	0.88
EN	RU	0.85	0.88	**0.83**
PL	EN	**0.86**	0.89	1.00
EN	CZ	**0.86**	0.89	0.87
RU	EN	0.88	**0.88**	1.00
CZ	EN	**0.89**	0.89	0.98

Results

Phylogenetic tree reconstruction

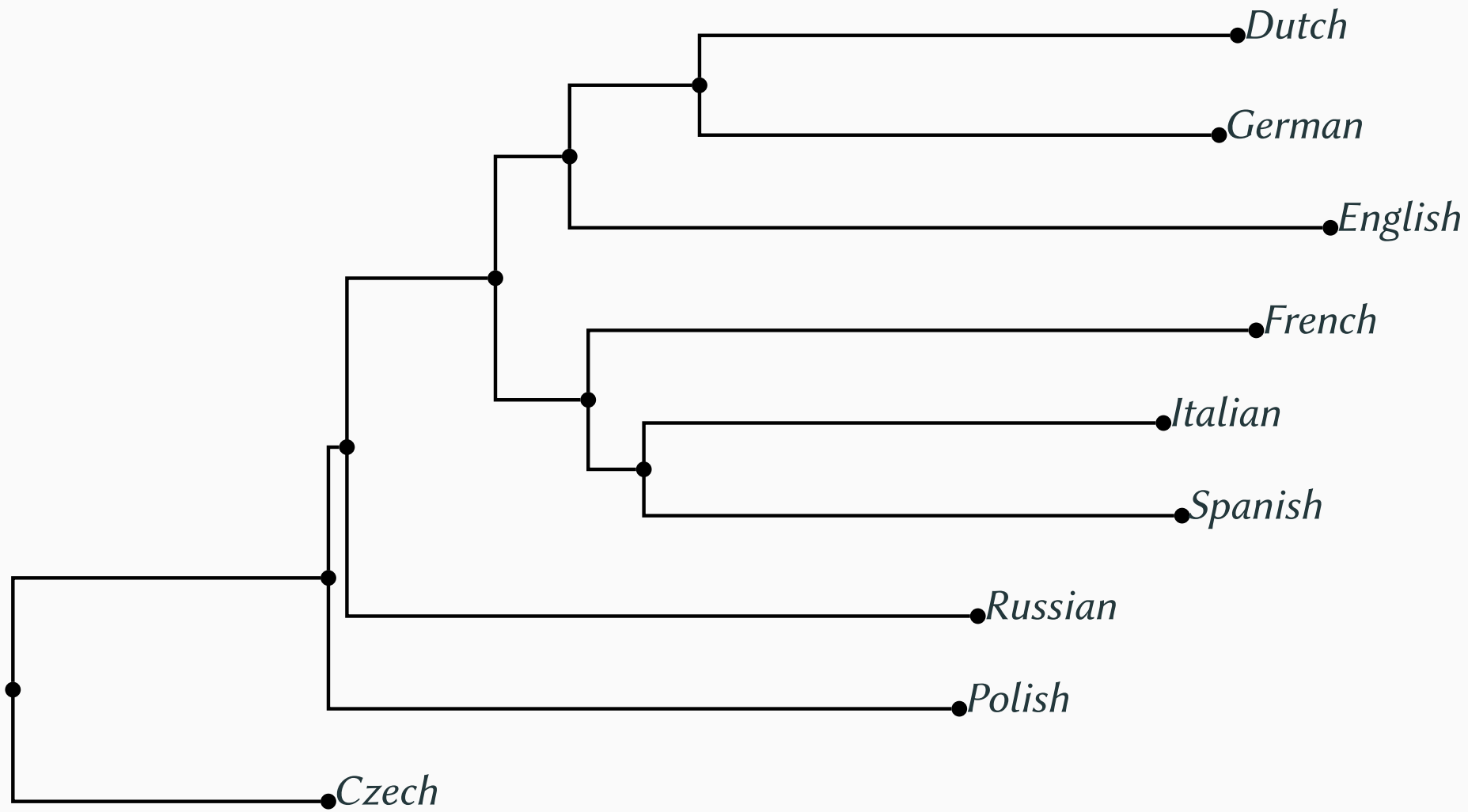
Phylogenetic tree reconstruction

- Perform word prediction for 9 Indo-European languages

	NL	DE	EN	RU	PL	CZ	FR	IT	ES
NL	0.00	0.57	0.75	0.81	0.82	0.83	0.82	0.80	0.78
DE	0.57	0.00	0.78	0.80	0.81	0.82	0.81	0.76	0.78
EN	0.75	0.78	0.00	0.87	0.86	0.88	0.83	0.82	0.82
RU	0.81	0.80	0.87	0.00	0.69	0.68	0.83	0.76	0.78
PL	0.82	0.81	0.86	0.69	0.00	0.68	0.83	0.79	0.79
CZ	0.83	0.82	0.88	0.68	0.68	0.00	0.85	0.77	0.80
FR	0.82	0.81	0.83	0.83	0.83	0.85	0.00	0.66	0.69
IT	0.80	0.76	0.82	0.76	0.79	0.77	0.66	0.00	0.57
ES	0.78	0.78	0.82	0.78	0.79	0.80	0.69	0.57	0.00

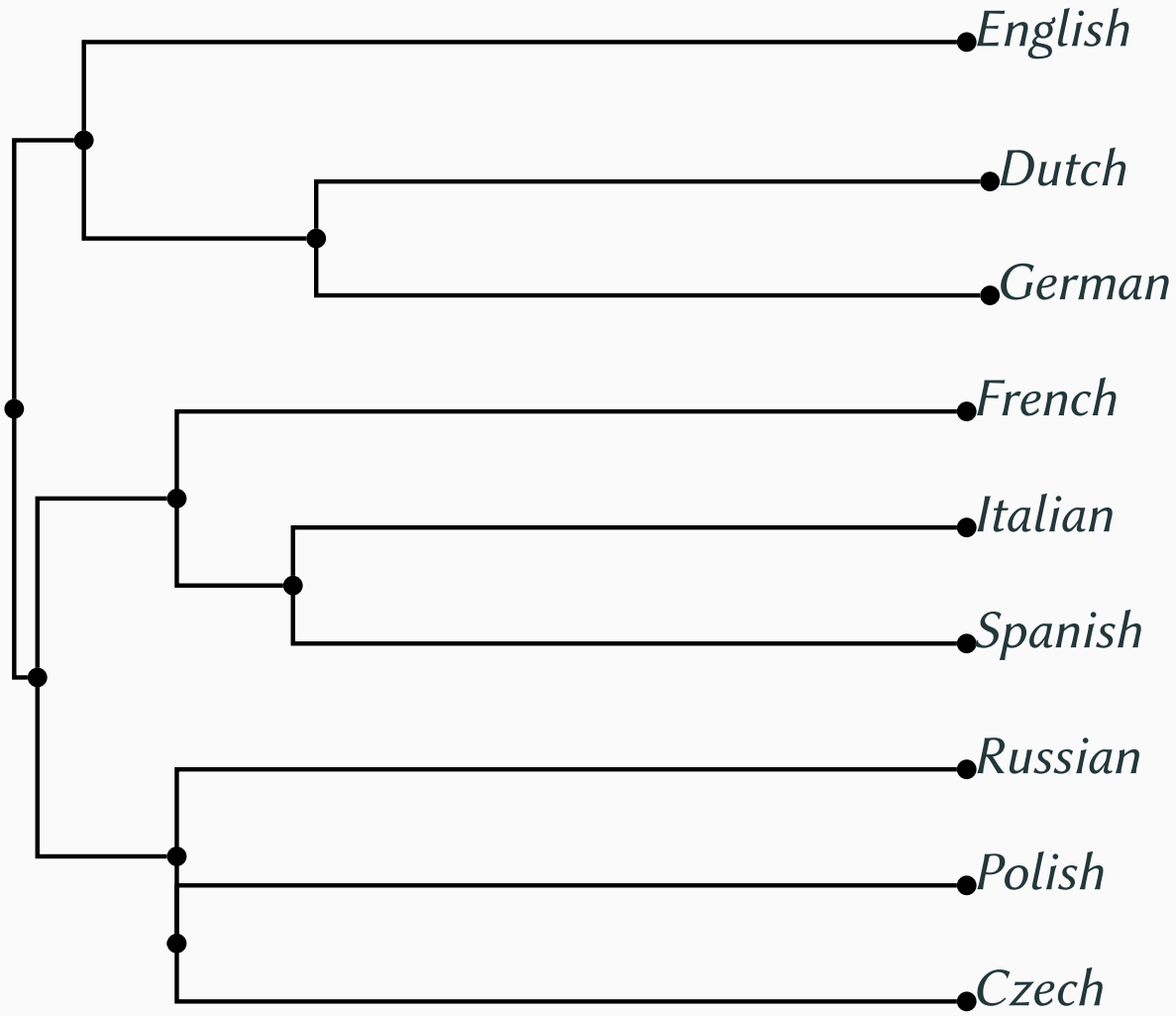
Phylogenetic tree: word prediction

Neighbor joining on word prediction distance matrix:



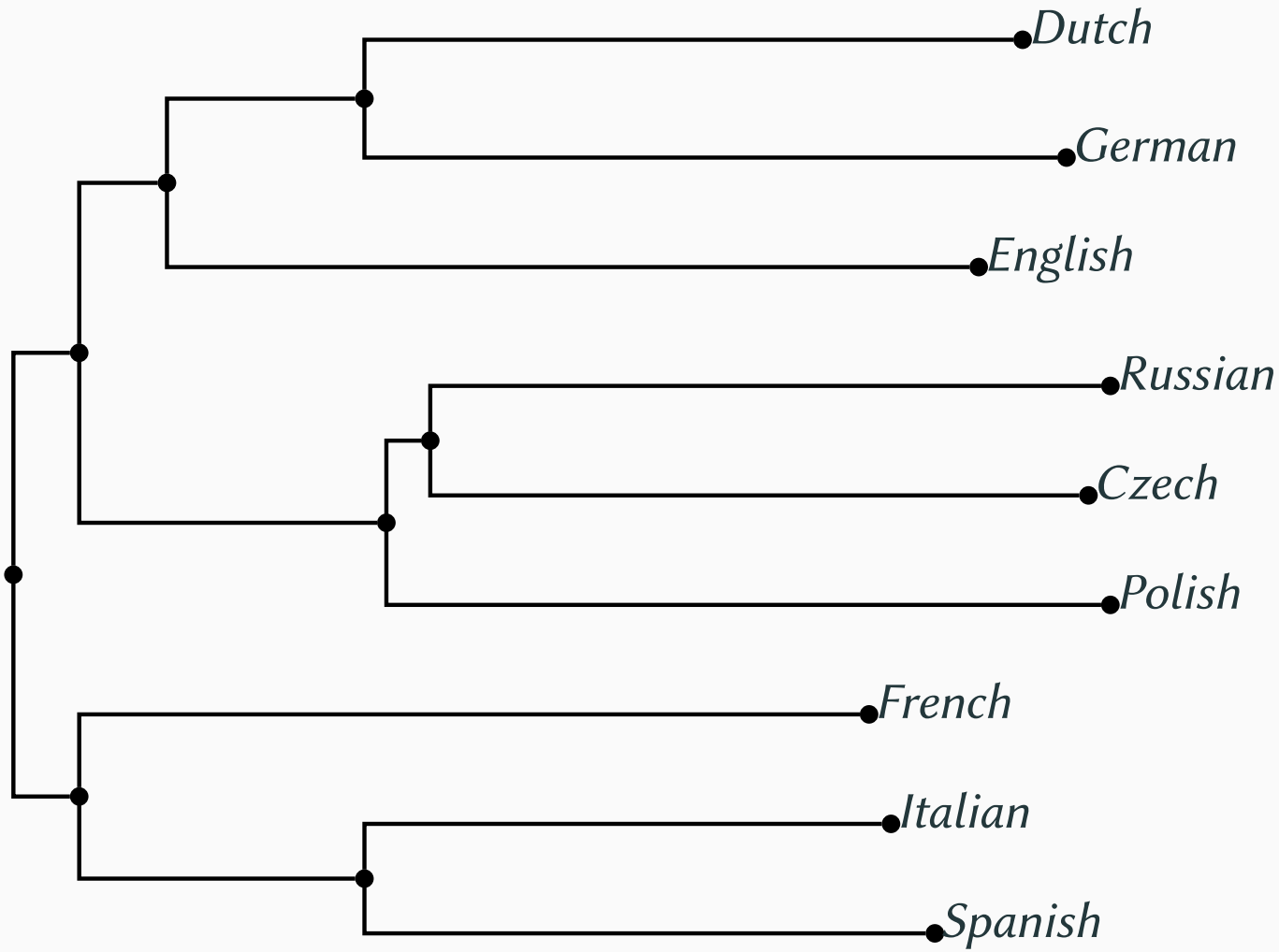
Phylogenetic tree: word prediction

UPGMA on word prediction distance matrix:



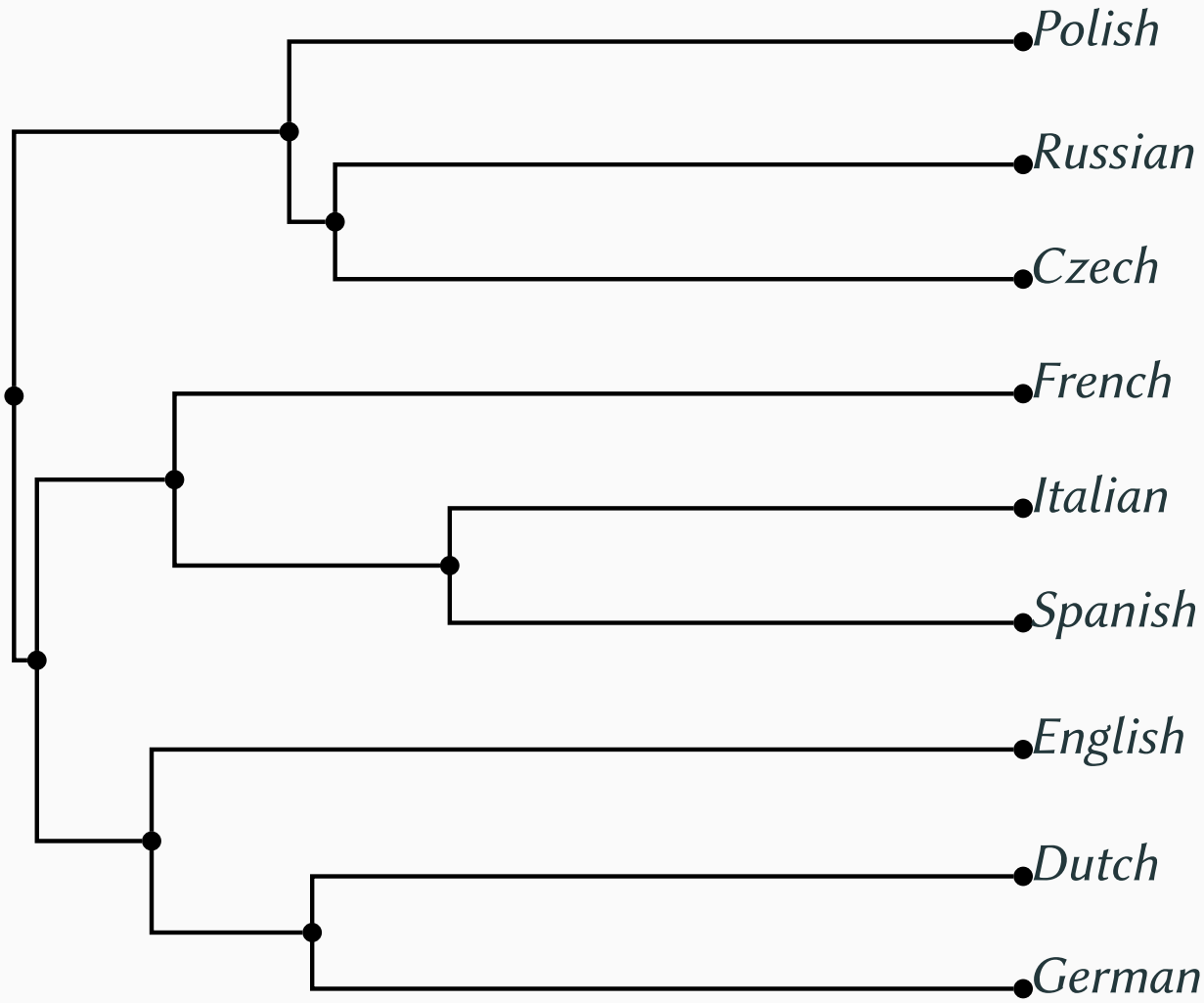
Phylogenetic tree: source-target baseline

Neighbor joining on **source-target baseline** distance matrix:



Phylogenetic tree: source-target baseline

UPGMA on **source-target baseline** distance matrix:



Results

Identification of sound correspondences

Identification of sound correspondences: NL-DE

Most frequent substitutions, using Needleman-Wunsch alignment:

Source	**Prediction**	Number
-	n	32
-	3	23
r	G	17
r	-	10
r	a	9
v	f	9
s	S	9
w	v	8
E	a	8
-	a	7

Source	**Target**	Number
-	n	31
r	G	21
-	3	15
E	a	10
-	a	10
r	a	10
w	v	8
s	S	8
e	E	7
t	c	7

Future work and conclusion

Future work

- Word prediction algorithm
  - Attention: use weighted sum of all encoder output steps (Bahdanau et al., 2014)
  - Obscured sound correspondences
    - Use more input languages, so model learns kind of proto-language
    - Multiple encoders, or even use of multiple languages in one encoder (Ha et al., 2016; Johnson et al., 2016)
  - Encode input as character embeddings
  - Link cognates across concepts
  - Perform evaluation only on cognates

Future work

- Word prediction algorithm
  - Attention: use weighted sum of all encoder output steps (Bahdanau et al., 2014)
  - Obscured sound correspondences
    - Use more input languages, so model learns kind of proto-language
    - Multiple encoders, or even use of multiple languages in one encoder (Ha et al., 2016; Johnson et al., 2016)
  - Encode input as character embeddings
  - Link cognates across concepts
  - Perform evaluation only on cognates
- Cognate detection

Future work

- Word prediction algorithm
  - Attention: use weighted sum of all encoder output steps (Bahdanau et al., 2014)
  - Obscured sound correspondences
    - Use more input languages, so model learns kind of proto-language
    - Multiple encoders, or even use of multiple languages in one encoder (Ha et al., 2016; Johnson et al., 2016)
  - Encode input as character embeddings
  - Link cognates across concepts
  - Perform evaluation only on cognates
- Cognate detection
- Use Bayesian MCMC for tree reconstruction

Future work

- Word prediction algorithm
  - Attention: use weighted sum of all encoder output steps (Bahdanau et al., 2014)
  - Obscured sound correspondences
    - Use more input languages, so model learns kind of proto-language
    - Multiple encoders, or even use of multiple languages in one encoder (Ha et al., 2016; Johnson et al., 2016)
  - Encode input as character embeddings
  - Link cognates across concepts
  - Perform evaluation only on cognates
- Cognate detection
- Use Bayesian MCMC for tree reconstruction
- Sound correspondences:
  - Extract sound correspondences from neural network

Conclusion

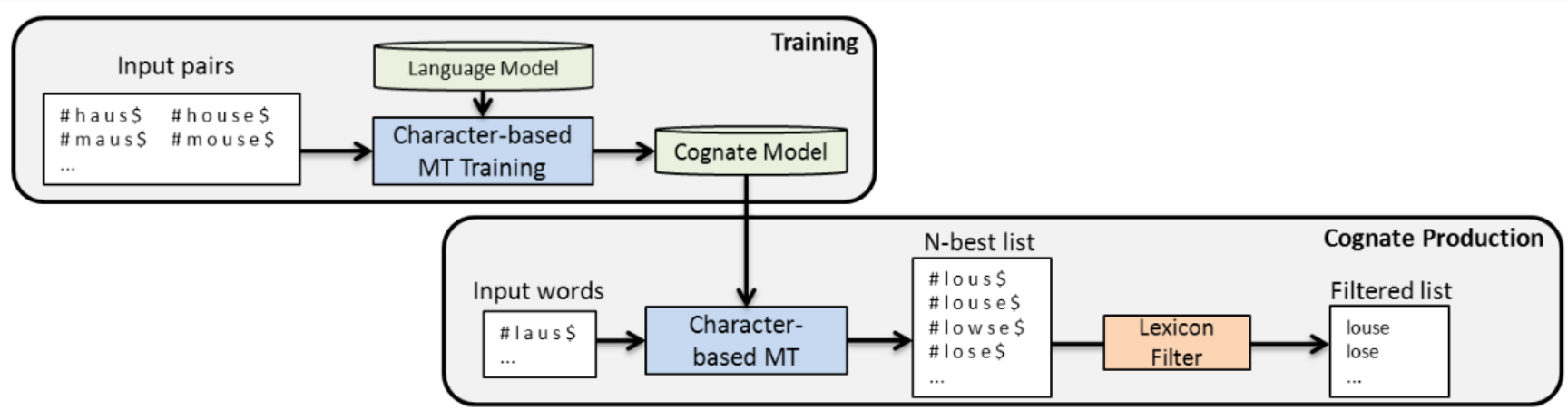
- Machine learning paradigm applicable to several tasks in historical linguistics:
  - Phylogenetic tree reconstruction
  - Identification of sound correspondences
  - Cognate detection
- Results of applications can become even more meaningful by improving prediction performance

Future work and conclusion

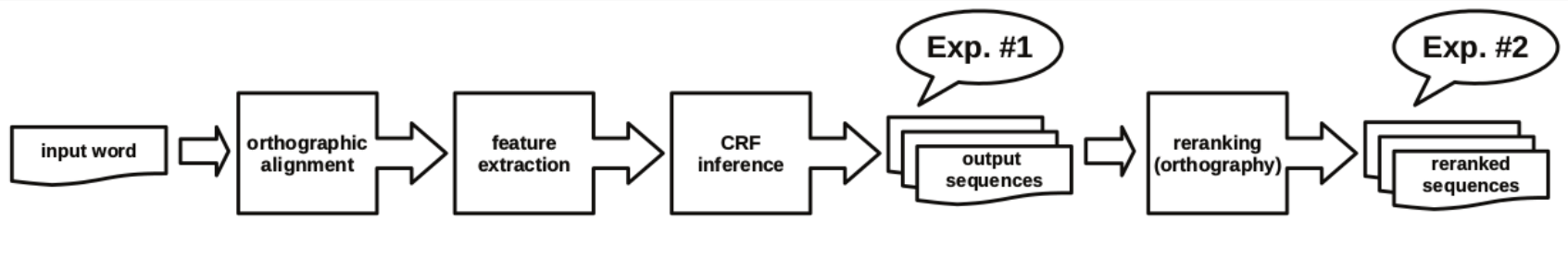
Additional information

Related work: cognate production

Beinborn et al. (2013):



Ciobanu (2016):



Cognate detection

- Perform word prediction for all language pairs between languages
- For every concept:
  - From prediction results of all language pairs, take into account word pairs for this concept
  - Cluster into cognate clusters based on prediction distance *for only this word pair*
- Compute B-Cubed F (Bagga and Baldwin, 1998) and compare to other approaches

References

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Beinborn, L., Zesch, T., and Gurevych, I. (2013). Cognate production using character-based machine translation. In *IJCNLP*, pages 883–891.

Brown, C. H., Holman, E. W., Wichmann, S., and Velupillai, V. (2008). Automated classification of the world′ s languages: a description of the method and preliminary results. *STUF-Language Typology and Universals Sprachtypologie und Universalienforschung*, 61(4):285–308.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Ciobanu, A. M. (2016). Sequence labeling for cognate production. *Procedia Computer Science*, 96:1391–1399.

Dellert, J. (2015). Compiling the uralic dataset for northeuralex, a lexicostatistical database of northern eurasia. In *Septentrio Conference Series*, number 2, pages 34–44.

Ha, T.-L., Niehues, J., and Waibel, A. (2016). Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Inkpen, D., Frunza, O., and Kondrak, G. (2005). Automatic identification of cognates and false friends in french and english. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 251–257.

Jäger, G., List, J.-M., and Sofroniev, P. (2017). Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists. *Mayan*, 895:0–05.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Kondrak, G. (2002). Determining recurrent sound correspondences by inducing translation models. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

List, J.-M. (2012). Lexstat: Automatic detection of cognates in multilingual wordlists. *EACL 2012*, page 117.

Needleman, S. B. and Wunsch, C. D. (1970). A gene method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453.

Rama, T. (2016). Siamese convolutional networks based on phonetic features for cognate identification. *arXiv preprint arXiv:1605.05172*.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Wieling, M., Prokić, J., and Nerbonne, J. (2009). Evaluating the pairwise string alignment of pronunciations. In *Proceedings of the EACL 2009 workshop on language technology and resources for cultural heritage, social sciences, humanities, and education*, pages 26–34. Association for Computational Linguistics.