Machine Learning of Phonotactics

Erik F. Tjong Kim Sang
Linguistics department
Uppsala University
Sweden
erik.tjong@ling.uu.se

overhead sheets: http://stp.ling.uu.se/~erikt/misc/groningen98.ps

Topic

Using machine learning techniques for deriving models that simulate natural language knowledge.

Goals

1. discovering what machine learning technique generates the best models,
2. finding out the influence of knowledge representation on the learning result and
3. determining whether initial language knowledge could improve the models generated by the learning process.

Learning problem

Deriving good models for the phonotactic structure of words.

Examples

bad, crwth, mloda, Nwanko, ptata

Extra restrictions

1. use Dutch data,
2. use monosyllabic data and
3. use only positive data during training phase.

Working method

1. Train with positive data.
2. Test with unseen positive data.
3. Test with negative data.

Training and test data

The data is available in two representation formats: the orthographic format and the phonetic format. The data was obtained from the CELEX Lexical Database.

training corpus: 5577 unique orthographic strings and 5084 unique phonetic strings

test corpus: 600 strings from each of the four test categories

Possible learning paradigms

1. statistical learning
2. connectionist learning
3. rule-based learning
4. genetic learning

Chosen learning algorithms

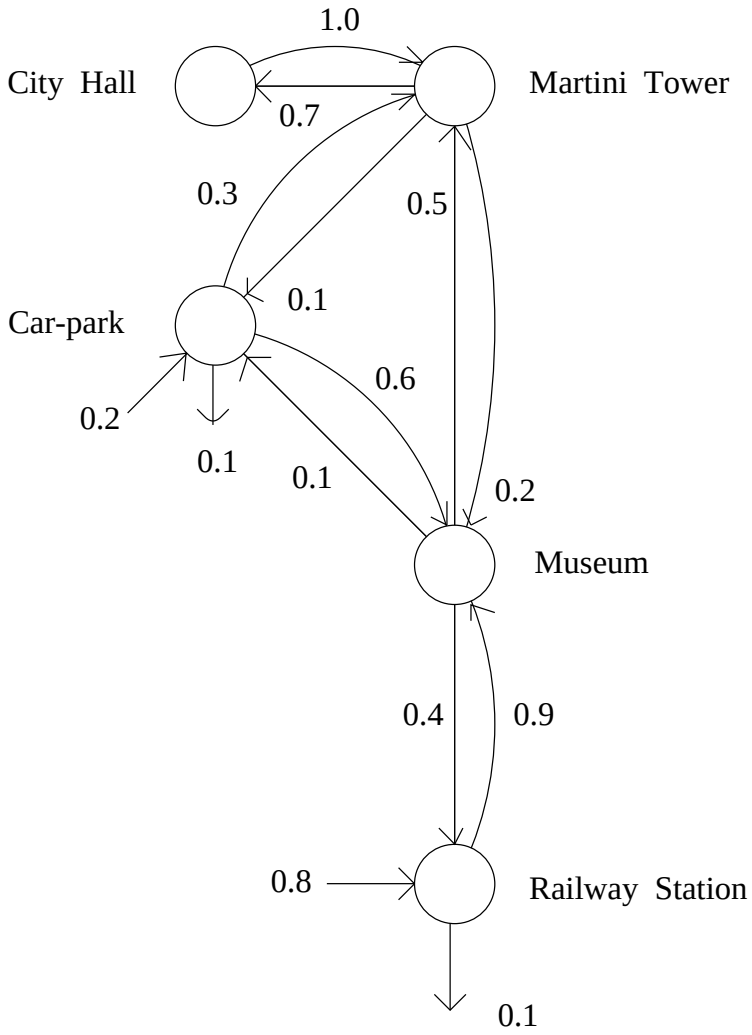
1. Hidden Markov Models (Rabiner and others)
2. Simple Recurrent Networks (Elman, Cleeremans)
3. Inductive Logic Programming (Muggleton)

Hidden Markov Models (HMMs)

- a collection of states
- each state can produce tokens with certain probabilities
- it is possible to move from one state to another

After training an HMM can assign scores (numbers between zero and one) to sequences of tokens.

An example HMM



HMM experiment questions

1. How do we make token production regard token context?
2. How do we prevent that nearly all short strings get higher evaluation scores than all long strings?

Answers

1. Regard strings as sequences of token bigrams.
2. Add extra initial and final token.

HMM results

The HMMs contained eight states which could process a subset of the 29 tokens.

Orthographic data

initialization	accepted positive data	rejected negative data
random	98.9%	91.0%
linguistic	98.9%	94.5%

Phonetic data

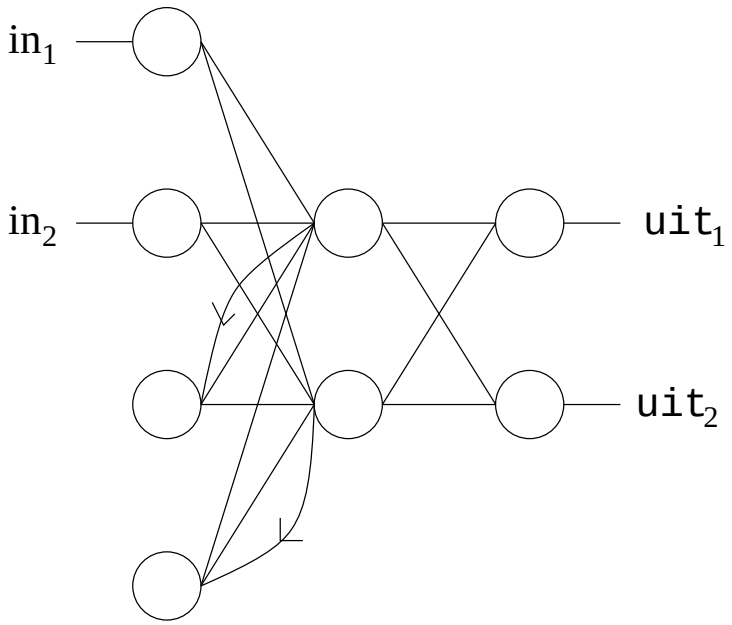
initialization	accepted positive data	rejected negative data
random	99.1%	98.3%
linguistic	99.1%	99.1%

Simple Recurrent Networks (SRNs)

- a collection of states
- each state accepts one or more numeric input signals
- each state generates exactly one output signal
- states send signals to each other via links with weights

After training an SRN computes an output pattern (a sequence of numbers) from an input pattern.

Example SRN



SRN experiment questions

1. How do we represent alphabetic characters as number sequences?
2. How do we train a pattern associator with positive data only?

Answers

1. Use the localist mapping: one binary digit per character.
2. Train the network to predict the next character in the string.

SRN results

The SRNs contained 29 cells in the input layer and the output layer. The best results have been obtained with a network with four cells in the hidden layer.

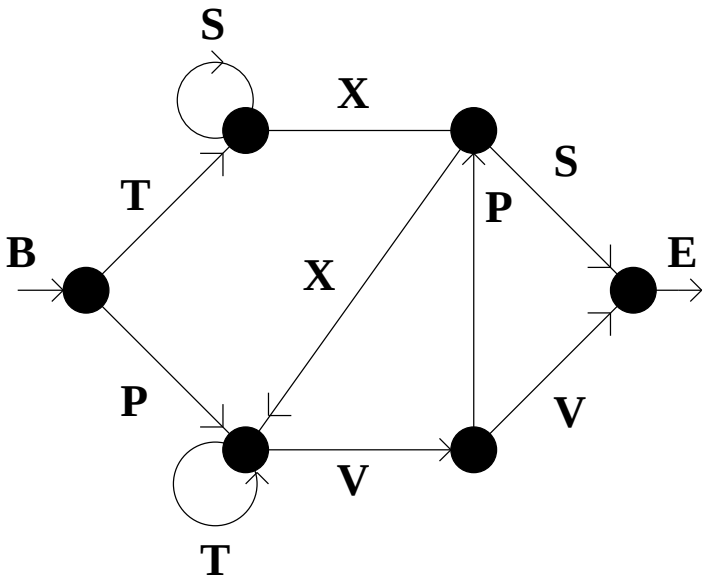
Orthographic data

initialization	accepted positive data	rejected negative data
random	100%	8.3%
linguistic	100%	4.8%

No experiments have been performed with phonetic data.

SRN performance explanation (1)

Inspiration: Cleeremans et.al. 1993: in experiments with strings generated by a Reber grammar SRNs have classified 100% of the strings correctly after training.



SRN performance explanation (2)

Tjong Kim Sang 1995: the SRN performance or strings generated by a Reber grammar will degrade when the complexity of the grammar is increased.

number of continuation symbols	accepted positive strings	rejected negative strings
2	100%	100%
3	100%	92.2%
4	100%	82.2%

Cause: signal/noise ration in network goes down.

Inductive Logic Programming (ILP)

A method for deriving hypotheses from some background knowledge and observations.

Example

P_1	All men are mortal.
P_2	Socrates is mortal.
IC	Socrates is a man.

ILP experiment questions

1. How do we limit the number of derived hypotheses?
2. What background knowledge do we use?

Answers

1. Use a limited set of hypothesis inferencing rules and remove hypotheses which can be derived from others.
2. Example: BACKGROUND KNOWLEDGE SUFFIX RULE: Suppose there exists a word W=$w_1 ... w_{n-1} w_n$ ($w_1 ... w_n$ are n characters) and a suffix hypothesis SH(w_{n-1},w_n).
   In that case the fact that W is a valid word will imply that $w_1 ... w_{n-1}$ is a valid word and vice versa.

ILP results

The size of the models generated by the ILP process varies between approximately 600 and 1200 rules.

Orthographic data

initialization	accepted positive data	rejected negative data
random	99.2%	60.0%
linguistic	97.8%	97.7%

Phonetic data

initialization	accepted positive data	rejected negative data
random	99.2%	70.6%
linguistic	99.2%	98.3%

Concluding remarks

Goals

1. discovering what machine learning technique generates the best models,
2. finding out the influence of knowledge representation on the learning result and
3. determining whether initial language knowledge could improve the models generated by the learning process.

Results

1. HMMs and ILP generate better models than SRNs.
2. The models generated for phonetic data are nearly always better than the models generated for orthographic data.
3. Initial linguistic knowledge improves the models, especially those generated by ILP.

Address

Address:	Vakgroep Alfa-informatica P.O. Box 716 9700 AS Groningen The Netherlands
telephone:	+31 50 3635970
WWW:	http://stp.ling.uu.se/~erikt/

Literature

- Erik F. Tjong Kim Sang, *Machine Learning of Phonotactics*, PhD thesis University of Groningen, The Netherlands, to appear in 1998.

Overhead sheets

http://stp.ling.uu.se/~erikt/misc/groningen98.ps