Introduction to CIT 831

Natural Language Processing

Wolfgang Menzel

Department of Informatics
Hamburg University

CIT 831 Natural Language Processing

1. Why is Natural Language Processing difficult?
2. Structuring the field
3. NLP between system development and science
4. The structure of the course

Readings

- Jurafsky, Daniel S., and James H. Martin (fothcoming) Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 3rd edition. Prentice-Hall.
- Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron: Deep Learning. MIT Press, 2016

Why is Natural Language Processing difficult?

- language is compositional
  - smaller units combine into larger ones
  - the meaning of a complex expression is determined by its structure and the meanings of its constituents (Gottlob Frege, 1879)
- language is complex
  - few elementary units, manifold ways to combine them
  - no upper length limitations for complex utterances

Why is Natural Language Processing difficult?

- language is ambiguous on all levels
  - phonological:
    - The same sound is spelled differently

      $/\text{fi:l}/ \rightarrow$ feel, $/\text{mi:l}/ \rightarrow$ meal
    - The same characters are pronounced differently

      read $\rightarrow /\text{ri:t}/$, read $\rightarrow /\text{r}\epsilon\text{d}/$, bear $\rightarrow /\text{b}\epsilon\text{:r}/$

Why is Natural Language Processing difficult?

- language is ambiguous on all levels
  - phonological:
    - The same sound is spelled differently

      /fi:l/ $\rightarrow$ feel, /mi:l/ $\rightarrow$ meal
    - The same characters are pronounced differently

      read $\rightarrow$ /ri:t/, read $\rightarrow$ /rϵd/, bear $\rightarrow$ /bϵ:r/
  - morphological:
    - -ed $\rightarrow$ past tense verb vs. past participle (vs. part of the stem)
    - -s $\rightarrow$ 3rd person singular vs. plural noun (vs. part of the stem)

Why is Natural Language Processing difficult?

- language is ambiguous on all levels
  - lexical:

rose/V	She rose from her chair.
	Disapproval rose from the audience.
	(She was afraid to rose.)
rose/N	This rose is beautiful.
	The flowers came in all shades of rose.
rose/A	I'll take the rose flowers.
light/N	I switched on the light.
	(That shed light on the issue.)
	In the light of the current situation ...
light/A	Light pressure might help.
	The light package came today.
light/V	We can light the fire with my matches.

Why is Natural Language Processing difficult?

- language is ambiguous on all levels
  - structural:
    - PP attachment:

      He saw the woman with the telescope.
    - Reduced relative clauses:

      We saw the Eiffel tower flying to Paris.

Why is Natural Language Processing difficult?

- language is flexible
  - the same or similar content can be expressed in very many different ways.
- language is shaped by individual or collective preferences
  - dialects, stylistic variations, ...
- language is dynamic
  - neologisms and dying-out words
  - semantic shift
  - meaning negotiation
- language uses mataphor, vagueness, underspecification, ...

Structuring the field

- applications
- linguistic descriptions
- knowledge acquisition
- system design
- modularization
- data structures
- tasks
- models
- methods and algorithms

Applications of NLP

???

Linguistic Descriptions

Complexity levels vs. semiotic perspectives

	syntax	semantics	pragmatics

Linguistic Descriptions

Complexity levels vs. semiotic perspectives

	syntax (form)	semantics (meaning)	pragmatics (purpose)

Linguistic Descriptions

Complexity levels vs. semiotic perspectives

	syntax (form)	semantics (meaning)	pragmatics (purpose)
phonology			

Linguistic Descriptions

Complexity levels vs. semiotic perspectives

	syntax (form)	semantics (meaning)	pragmatics (purpose)
phonology			
morphology			

Linguistic Descriptions

Complexity levels vs. semiotic perspectives

	syntax (form)	semantics (meaning)	pragmatics (purpose)
phonology			
morphology			
grammar			

Linguistic Descriptions

Complexity levels vs. semiotic perspectives

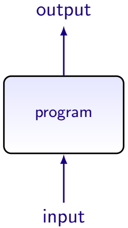
	syntax (form)	semantics (meaning)	pragmatics (purpose)
phonology			
morphology			
grammar			
discourse			

Knowledge acquisition

- manual compilation
  - using a formalism for knowledge representation
- machine learning
  - symbolic, probabilistic, neural, ...
  - supervised, unsupervised, semi-supervised, self-supervised

The history of NLP system design

Rule-based architectures

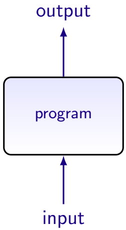


monolithic programs

1950 – 1970

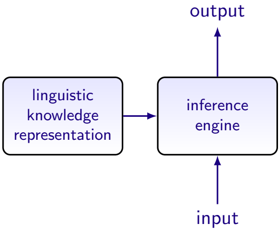
The history of NLP system design

Rule-based architectures



monolithic programs

1950 – 1970

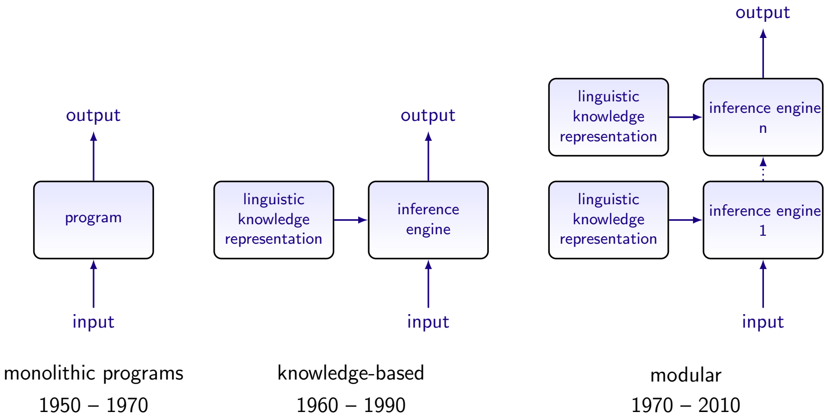


knowledge-based

1960 – 1990

The history of NLP system design

Rule-based architectures



Modularization

- Partial components for modular systems
  - morphological analysis
  - part-of-speech tagging
  - syntactic/semantic parsing
  - pragmatic analysis
  - named entity recognition
  - coreference resolution
  - semantic role labeling
  - text planning
  - text generation
- results are fed as features into a subsequent component

Modularization

(Good) reasons for developing modular systems:

- complexity reduction: simpler solutions for (simpler) partial tasks
- availability of training data: less expensive data collection (and annotation) for simpler tasks

Modularization

(Good) reasons for developing modular systems:

- complexity reduction: simpler solutions for (simpler) partial tasks
- availability of training data: less expensive data collection (and annotation) for simpler tasks

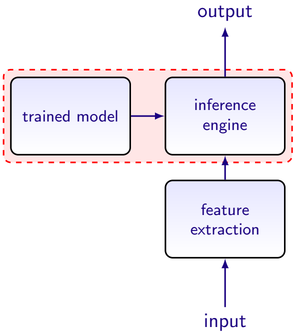
Drawbacks of modular systems

- error percolation: wrong decisions in earlier components might cause severe breakdowns in later ones
- ambiguity percolation: degree of ambiguity may rise exponentially with the number of components in a pipeline
- models are developed (i.e. optimized) separately and need to be fitted

The history of NLP system design

Traditional machine learning (symbolic/stochastic/neural)

often almost indistinguishable

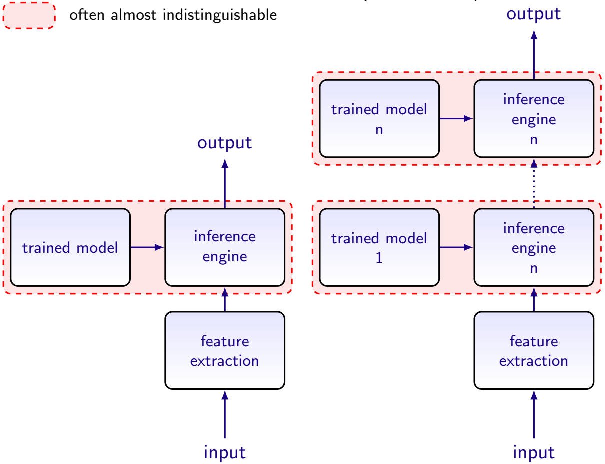


monolithic

1980 – 2000

The history of NLP system design

Traditional machine learning (symbolic/stochastic/neural)



monolithic

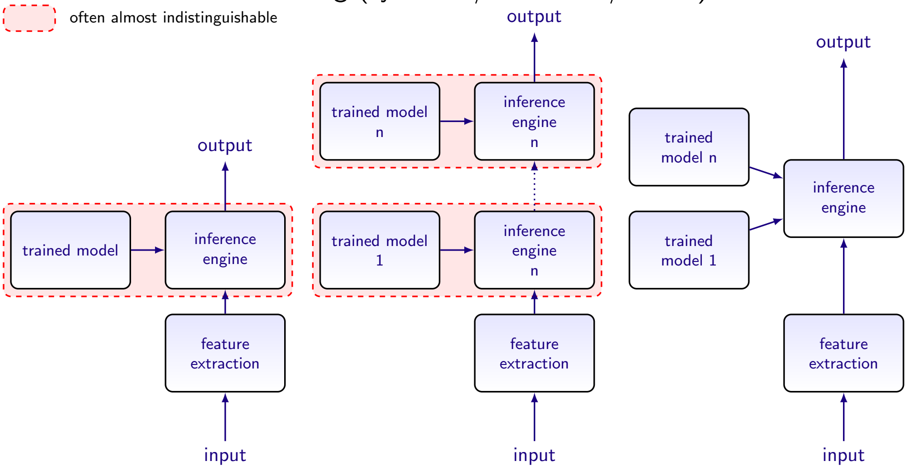
1980 – 2000

modular components

1970 – 2010

The history of NLP system design

Traditional machine learning (symbolic/stochastic/neural)



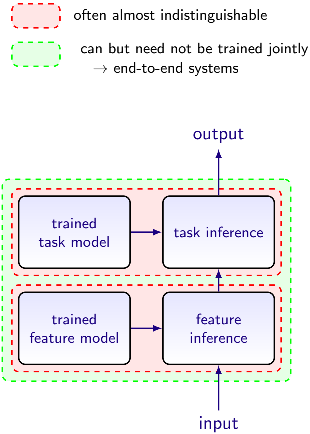
monolithic
1980 – 2000

modular components
1970 – 2010

modular models
1980 – 2020

The history of NLP system design

Representation learning (mainly neural architectures)



shallow

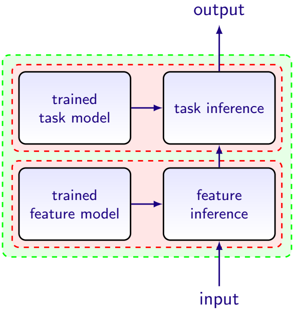
2010 –

The history of NLP system design

Representation learning (mainly neural architectures)

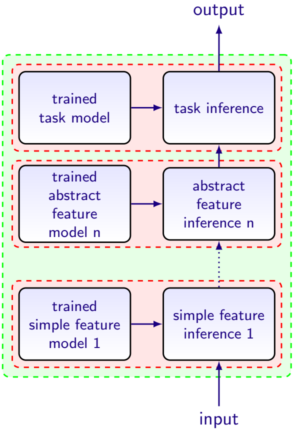
often almost indistinguishable

can but need not be trained jointly
→ end-to-end systems



shallow

2010 –



deep

2020 –

The history of NLP system design

1950 - 1970	monolithic systems: single program end-to-end
1960 - 2010	modularized systems: several cooperating components, most often arranged in a pipeline
2000 -	monolithic systems: trained end-to-end

Some terminology

- (Applications of NLP)
- Representations for NLP
- NLP tasks
- Models for NLP
- Methods/Algorithms for NLP

Input/Output Representations for NLP

- symbolic:
  - strings
  - ???

NLP tasks

- classification

  NL item $\mapsto$ category
- ???

Models

- classification/prediction
  - (naive) Bayes classifier
  - support vector machines
  - decision trees
  - multi-layer perceptrons
- measuring complexity
  - (linear) regression models
  - (multilayer) perceptron
- measuring similarity
  - cosine similarity (vectors)
  - alignments (strings, trees, graphs)

Models

- sequence-to-sequence transformation
  - finite state machines
  - (transformation) rules
  - hidden Markov models
  - recurrent neural networks
- structural prediction
  - context-free, dependency, unification-based and constraint-based grammars
- discourse planning, generation
  - special purpose formalisms

Important methods/algorithms

- symbolic
    - search (mostly for optimization)
        - systematic (taboo) search
        - (randomized) gradient descend
        - beam search, best first search
    - alignment
        - dynamic programming
    - structured prediction
        - (model specific) parsing algorithms (Earley, CYK, ...)
        - dynamic programming
        - term/graph unification
        - stochastic and neural classifiers
        - (stochastic) constraint satisfaction
    - probabilistic inference, argmax
- subsymbolic
    - backpropagation
    - softmax

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development
- like every other branch of engineering it needs *insights*
- that's where *science* starts!

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development
- like every other branch of engineering it needs *insights*
- that's where *science* starts!
- insights are created by asking questions

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development
- like every other branch of engineering it needs *insights*
- that's where *science* starts!
- insights are created by asking questions
- which kind of questions?

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development
- like every other branch of engineering it needs *insights*
- that's where *science* starts!
- insights are created by asking questions
- which kind of questions?
    - What?

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development
- like every other branch of engineering it needs *insights*
- that's where *science* starts!
- insights are created by asking questions
- which kind of questions?
  - What?
  - How?

NLP between system development and science

- NLP aims at developing computational systems
- but it should not be reduced to system development
- like every other branch of engineering it needs *insights*
- that's where *science* starts!
- insights are created by asking questions
- which kind of questions?
    - What?
    - How?
    - Why?

NLP between system development and science

Typical research questions (about methods)

- Feasibility: Is method A well suited to solve problem B?
  - this includes evaluation methods

NLP between system development and science

Typical research questions (about methods)

- Feasibility: Is method A well suited to solve problem B?
  - this includes evaluation methods
- Transfer: Can a method be ported to another problem or a different language?
  - What are the problems expected or observed?
- Evaluation: How reliable does method A solve problem B
  - formal measures (e.g. string, tree or graph similarity)
  - comparison with human judgements (e.g. translation, relevance of documents)
- Comparison: Is method A better suited than method B to solve problem C?
  - use of standardized test sets

NLP between system development and science

Typical research questions (about methods)

- Feasibility: Is method A well suited to solve problem B?
  - this includes evaluation methods
- Transfer: Can a method be ported to another problem or a different language?
  - What are the problems expected or observed?
- Evaluation: How reliable does method A solve problem B
  - formal measures (e.g. string, tree or graph similarity)
  - comparison with human judgements (e.g. translation, relevance of documents)
- Comparison: Is method A better suited than method B to solve problem C?
  - use of standardized test sets

NLP between system development and science

Typical research questions (about methods, continued)

- Complexity: How expensive it is to apply method A to problem B?
  - theoretical/empirical complexity results
  - worst case, typical case, ...

NLP between system development and science

Typical research questions (about methods, continued)

- Complexity: How expensive it is to apply method A to problem B?
  - theoretical/empirical complexity results
  - worst case, typical case, ...
- Explanation: What are the reasons for success / failure of a method?

NLP between system development and science

Typical research questions (about representations)

- Relevance: Does knowledge about a linguistic phenomenon help to better solve a given problem?

NLP between system development and science

Typical research questions (about representations)

- Relevance: Does knowledge about a linguistic phenomenon help to better solve a given problem?
- Adequacy: How well a given kind of representation can distinguish between task relevant input data?
  - e.g. bag of words are unable to deal with negation or word order

NLP between system development and science

Typical research questions (about representations)

- Relevance: Does knowledge about a linguistic phenomenon help to better solve a given problem?
- Adequacy: How well a given kind of representation can distinguish between task relevant input data?
  - e.g. bag of words are unable to deal with negation or word order
- Which kind of annotation should be used?
  - Can the problem be mapped to another one, by using an alternative annotation?

NLP between system development and science

Typical research questions (about representations)

- Relevance: Does knowledge about a linguistic phenomenon help to better solve a given problem?
- Adequacy: How well a given kind of representation can distinguish between task relevant input data?
  - e.g. bag of words are unable to deal with negation or word order
- Which kind of annotation should be used?
  - Can the problem be mapped to another one, by using an alternative annotation?
- Probing: What kind of knowledge is contained in a representation of type A?
  - mostly used to inspect subsymbolic vector representations

NLP between system development and science

Typical research questions (about representations)

- Relevance: Does knowledge about a linguistic phenomenon help to better solve a given problem?
- Adequacy: How well a given kind of representation can distinguish between task relevant input data?
  - e.g. bag of words are unable to deal with negation or word order
- Which kind of annotation should be used?
  - Can the problem be mapped to another one, by using an alternative annotation?
- Probing: What kind of knowledge is contained in a representation of type A?
  - mostly used to inspect subsymbolic vector representations
- Model complexity: e.g. how many dimensions are needed?

The structure of the course

Learning goals:

- learning of fundamental concepts of contemporary NLP tasks, tools and applications
- training of elementary techniques of scientific work, i.e.
  - formulating research questions,
  - conducting literature studies,
  - presenting scientific results and
  - writing a scientific text in its different phases: drafting, revising, reviewing

The structure of the course

Two major components:

- the reading club:
  reading, discussing and understanding novel concepts
- the writing club:
  presenting, discussing and publishing research questions and insights

The structure of the course

The reading club:

- home: read commonly agreed upon chapters of (Jurafsky and Martin, forthcoming)
- home: formulate your questions about
    - missing foundations
    - difficulties to understand
    - relationships to other concepts from NLP and CS
    - comparison of different methods
    - transfer of ideas to other problems or languages
    - ...
- home: post your questions on the Etherpad for the course
- meeting: cooperative attempts to answer the questions
- home: try to answer the questions still open

How to read a paper/book?

Reading via iterative refinement and selection

How to read a paper/book?

Reading via iterative refinement and selection

- Preselection: Is the paper worth considering?
  - Who is the author?
  - Where has it been published?
  - Has it been cited? By whom? How?

How to read a paper/book?

Reading via iterative refinement and selection

- Preselection: Is the paper worth considering?
  - Who is the author?
  - Where has it been published?
  - Has it been cited? By whom? How?
- Gisting/Ranking: How important it is to read the paper?
  - What the paper is about?
  - What kind of paper is it?
    - Review
    - Tutorial about solutions, evaluation methods, tools, models, or data sets
    - Research paper

How to read a paper/book?

Reading via iterative refinement and selection

- Preselection: Is the paper worth considering?
  - Who is the author?
  - Where has it been published?
  - Has it been cited? By whom? How?
- Gisting/Ranking: How important it is to read the paper?
  - What the paper is about?
  - What kind of paper is it?
    - Review
    - Tutorial about solutions, evaluation methods, tools, models, or data sets
    - Research paper
  - How well does it fit my personal research interests?
  - Is it interesting? Is it of general interest?
  - Are the reported results promising?

How to read a paper/book?

- Shallow understanding: Does the paper contribute important ideas to my research?
  - What was the goal of the research reported?
  - What are the methods/tools applied?
  - What other methods they are based on?
  - What is the novel contribution?
  - Are the results better to alternative approaches?

How to read a paper/book?

- Shallow understanding: Does the paper contribute important ideas to my research?
  - What was the goal of the research reported?
  - What are the methods/tools applied?
  - What other methods they are based on?
  - What is the novel contribution?
  - Are the results better to alternative approaches?
- Deep understanding: How does the approach really work?
  - What are the differences to alternative approaches?
  - Could I replicate the results?
  - Could I adapt the approach to my problem?
  - Which aspects of the solution contributed most to its success?
  - Can these aspects be ported to my problem/approach?

How to read a paper?

- Postprocessing/Documentation:
  - Add the paper to your personal data base.
  - Write a short summary of the paper.
  - Re-read the paper after having seen related ones.
  - Revise the summary if necessary.

The structure of the course

The writing club:

- home: formulate one or several (research) questions which might be of interest to the other participants of the course
- home: collect material needed to answer the question(s)
- meeting: give a talk about the question(s) and the answer(s) found
- meeting: discuss the content of the presentation
- meeting: provide feedback to the quality of the presentation
- home: write an essay about your "research" question(s)
  - motivate the question (What was unclear and why? etc.)
  - provide the background information required to understand the question(s)
  - provide your answer(s) and justifying its/their appropriateness
  - put your answers into perspective (What remains to be found out? How important are your findings? etc.)

The structure of the course

The writing club (continued):

- home: write of a review about the essay of another participant with the goal to given her/him helpful feedback on how to improve the essay?
- home: revise your essay according to the feedback received
- home: write a review about the essay of another participant with the goal to inform a fictional program committee of a conference (or the editorial board of a journal) about the strengths and weaknesses of the essay