Faculty of Computer Science, Dalhousie University *25-Sep-2025*

CSCI 4152/6509 — Natural Language Processing

Lecture 2: Ambiguities in NLP; Course Project

Location: Studley LSC-Psychology P5260 Instructor: Vlado Keselj
Time: 14:35 – 15:55

Previous Lecture

- Syllabus and web site review
- Course Introduction

Introduction to NLP
- Definition of NLP
- Some NLP applications
- NLP as a research area
- Short history of NLP
- NLP methodology overview
- Levels of NLP

2.6 Why is NLP Hard?

Since the start of NLP it has been known to be deceptive in term of difficulty of the major NLP tasks, such as machine translation or question answering. It is relatively easy to build toy systems that can perform these tasks successfully on small examples, and they give appearance that scaling up to a large domain will not be difficult. However, very often scaling up these systems to a usable coverage of examples has proven not only to require much more time but actually impossible due to emerging complexities. We will try to here to explain some main sources of this hidden difficulty of NLP.

NLP is Generally Hard

- NLP problems were tackled since 1950s
  - progress has been surprisingly slow and difficult
- Some external evidence of why NLP would be hard:
  - Turing test (imitation game)
  - Evidence from neuro-science:
    "A defining difference between man and non-human primates has been found in the circuitry of brain cells involved in language, according to researchers at the Medical College of Georgia."

`https://www.sciencedaily.com/releases/2001/09/010905071926.htm`

Some Computational Reasons that NLP is Hard

1. *highly ambiguous*
   - not easy to program disambiguation
2. *vague* (the principle of minimal effort)
   - not easy to program the context and a priori knowledge
3. *universal* (domain independent)

- not easy to program general knowledge representation

All of these require reasoning (inference)

Natural Language Processing (NLP) is an interesting area in the sense that many tasks appear to be solvable in a relatively straightforward way, but after solving a small number of examples and trying to scale up to a more general language scope they turn up to be much more difficult, and in practice even impossible to solve. It is useful to understand what is the source of this surprising and often hidden difficulty so that we can more easily distinguish very difficult tasks from more feasible ones, and so that we can modify a given task in a way that makes it more manageable.

It is generally recognized that ambiguities in any natural language are the main source of difficulty in NLP, and they are made even worse by minor grammatical mistakes that we freqently make. We will add two more properties of natural languages that make NLP difficult and identify that the three main sources of difficulty are:

- ambiguity,
- vagueness, and
- universality

of natural languages.

Ambiguity. Natural languages are ambiguous. They are ambiguous frequently and at different levels of NLP. When we use language as humans we resolve many literal ambiguities by using our complex understanding of the context, but frequently language is ambiguous even to us. Many language jokes are based on these inherent ambiguities. This means that if we write a program to process language and it operation depends on resolving these ambiguities, it may be a difficult or even impossible to solve some input cases. In further text, we will show many examples of ambiguities.

Vagueness. Another source of difficulty is vagueness of language in its typical use.

Universality. The third source of NLP difficulty is universality of language in the sense that the same natural language is typically used to describe very different areas of human interest: It is used for small talk, for communicating common sense knowledge, scientific knowledge, history, physics, abstract mathematics, legal rules, jokes, and so on.

Slide notes:

Ambiguities at Many Levels of NLP

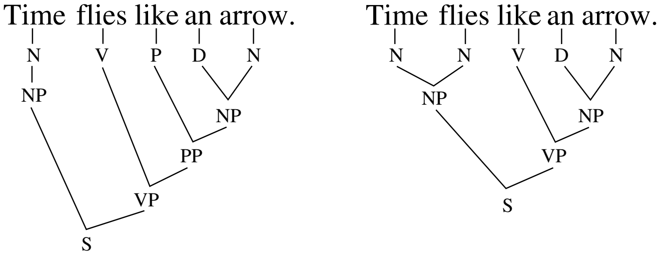
- Ambiguities of different types happen at all levels of NLP
- We will look at some examples at different levels of NLP

Phonological Ambiguities

- For example, the following words sound the same:
- two and too, sometimes even to
- would and wood
- there and their
- it's and its
- sea and see
- I scream and ice cream

Syntactic Level Ambiguity

Let us consider the following sentence for example: Time flies like an arrow. A obvious meaning is that time is passing quickly. However, if we consider the sentence in the following extended context: Time flies like an arrow, and fruit flies like a banana. another possible meaning becomes apparent. This meaning states that there exists a kind of insects called fruit flies and they like to eat arrow fruit. These two meanings correspond to two different syntactic interpretations of the sentence, which can be represented as the following two parse trees:



The following are some similar examples of syntactic ambiguity:

- Swat flies like ants.
- I saw a man with a telescope.
- I made her duck.
- I bought a computer with a smart card.
- The cow was found by a stream by a farmer.
- Flying planes can be dangerous.
- They are hunting dogs.
- Eye Drops Off Shelf.
- I'm glad I'm a man, and so is Lola.
- Somewhere in the world a woman gives birth every nine minutes.

Semantic Ambiguities

- **semantic lexical ambiguity,** e.g. The word "hot" may have many different meanings, such as having high temperature, spicy, intense, good looking, or stolen.
- Semantic ambiguity examples at the phrase level:
  1. What does "coast road" mean? Is it a road that leads to a coast, or a road that follows the coast?
  2. "carriage return" — Is it a return of a carriage, or an ASCII character?
  3. "kick the bucket", and other idioms
- **referential ambiguity** — a kind of semantic ambiguity, or it can be considered discourse ambiguity Example: 'It,' or 'he' in a text – what and who does it refer to?

Pragmatic-level Ambiguity

Examples:
- 12am — is it noon or midnight?
- What date is 10/11/12. Nov 10 or Oct 11 of 2012?

Ambiguities at the pragmatic level include various different pragmatic interpretations of meaning. For example, it is commonly understood and defined that 12am denotes midnight, but one could follow a logic reasoning that since it follows 11am, it may mean noon. As a similar time notation example, a pragmatic interpretation of date notation such as '10/11/12' differs in Europe and North America for example, where in Europe it means 10-November-2012, and in U.S. it may mean October 11, 2012.

3 About Course Project

The course project for graduate students (CSCI 6509), who are in a research stream (e.g., MCS, PhD, or other thesis students) should follow the basic structure of a typical research project, such as the research work on a thesis, only on a smaller scale. The undergraduate students (CSCI 4152) can choose as well to do a research project, or they can do a more purely implementation-focused project. The graduate students who are not in a research stream (e.g., MACS or MDI non-thesis students) may select a research topic, but also an application or business-oriented project with core methodology based on NLP.

You can form project teams of up to four students, or work individually.

The final paper should be in the form of a technical report. The presentations will be up to 8 minutes long, followed by 4 minutes for questions and switching speakers. The graduate students in the research stream programs must give individual presentations, while graduate students in professionally-oriented streams and undergraduate students can present as a team, or individually if they prefer. The teams can include mix of graduate and undergraduate students.

Regarding the presentation dates, the presentation free slots will be posted during the term on the course calendar page, and once they are posted the students can express preference for a presentation slot by email. The presentations will be scheduled based on first-come-first-served basis.

3.1 Deliverables

The deliverables related to the project are: Project Proposal (P0), Project Statement (P1), oral presentation (P), and the project report (R). The deadlines for deliverables are as follows:

- P0 — project topic proposal, worth 1%, plain text by email, due on **Friday Oct 10** by midnight,
- P1 — a project statement, worth 5 %, PDF, due on **Friday Nov 7** by midnight
- P — oral presentation, book time slot early, submit slides or slides overview (PDF or PPT) 24h ahead, and
- R — project report, electronically, due on **Tuesday Dec 9** by midnight

Note about emails: All emails related to the course must have the course number included in the subject, such as CSCI4152, CSCI6509, or, the best option is CSCI4152/6509.

This would help make sure that the emails are not misplaced as spam, and that they are given a high priority. Some deliverables may be needed to be submitted by email, and they will be given further specification about subject line contents. In all cases, the course number is required to be in the subject line.

Please read also the project web page on the course web site at:
`https://web.cs.dal.ca/~vlado/csci6509/project.html`
The page contains more details about project deliverables, and will also be updated with the latest information during the term.

P0 — Project Topic Proposal

Worth: 1% of the final mark.

You will need to choose a topic for your project. By the due date, you need to send to the instructor an email in plain text with the following information:

- tentative title,
- the list of team members, and
- one paragraph description of the topic.

Please do this as soon as you have chosen a topic. If two or more students or groups have the same or very similar topics, the one that sends P0 later may be required to change the topic. If the topic is not sufficiently relevant to the course, you may be asked to change it.

P1 — Project Statement

The project statement is worth 5% of the final mark.

The project statement will be submitted using GitLab (plain text or PDF). It should be about 2 pages long. It must include:

- Project title,
- Names of the member(s) of the group,
- Problem statement,
- List of possible approaches with citations to relevant work,
- Project plan for the rest of the term, and
- List of references.

The statement should identify a feasible project. P1 will be marked based on its completeness, clarity of presentation, and research on and analysis of related work.

P — Oral Presentation

Worth: 10% of the final mark, including class participation.

You are required to submit the slides of the presentation at least 24 hours before your presentation by email to the instructor. The slides can be original slides, or an slides handout (e.g., 6 slides per page), and they must be in a PowerPoint or PDF format. I will assume that you will use your computer for presentation, but if you need a computer, let the instructor know and we can arrange that you use the instructor's computer. If you use the instructor's computer, you can only use PDF or PowerPoint slides, without use of Internet or running any other programs, and the slides **must** be sent before the presentation.

Duration: The presentations should last up to 8 minutes, with 2 additional minutes reserved for questions and changing speakers, for the total of 12 minutes. **Content:** There is a significant flexibility in choosing the topic of your presentation, but it should be related to the project. It could be the work you have done up to that time, or what you plan to do. It is a good idea to include research or other related work that you did so far. You could also present a related method from the textbook or another paper.

Evaluation scheme for presentations:

- content: how interesting and valuable is the presentation, appropriateness of the topic, appropriateness for the audience and the time allocated,
- presentation: clarity, eye contact with the audience; it should be vivid, interesting; it is not a good idea to read from a paper, or from the slides; avoid looking too much to the slides rather than to the audience; do not just present to one person (e.g., instructor) — talk to the whole audience, organization and structure of the presentation should be well-planned; time length should be appropriate,
- slides: organization of the presentation; slides content: appropriate amount of text, use of figures,
- question answering: listening and answering the questions being asked, appropriate answers, answering the actual question to the point, but not going into a too lengthy additional discussion.

R — Project Report

Worth: 20% of the final mark.

The written project report is submitted in an electronic form by the last day of classes. A project report must be submitted as a PDF document. The reports are kept in archive with the instructor for several years.

A typical structure of a report:

- Title, author, course name, date
- Abstract
- 1. Introduction, 2. Related work
- 3. Problem description, Methodology
- 4. Experiment design, implementation

- 5. Evaluation
- 6. Conclusion
- References, Appendices

The following are more details about the stucture of report sections:

- Title, Author, Course name,
- Abstract — make sure it is an abstract of the whole paper and not just a part of the introduction. The abstract should be brief, definitely not longer than a half of a page.
- Introduction — introduce the problem; get a reader's attention; explain motivation and significance of the problem, define paper objectives. It is said that the title, abstract, introduction, and the whole paper should in a way express the same story in 10, 100, 1000, and 10,000 words, respectively. (Do not take these numbers literally!)
- Related Work — cover related work. Has this topic been studied yet? Do not just give an annotated list of citations. Give a critical analysis of the previous work.
- Problem Definition and Methodology — there is no good research without a research problem. Define it precisely. Describe your methodology, algorithms, system overview, or similar.
- Experiment Design — experiments are not mandatory, but some form of evaluation of your approach should exist.
- Evaluation, Discussion of evaluation results
- Conclusion
- References
- Appendices

This structure is just a guideline and parts may not be relevant to your project. There are no fixed requirements about the length of the paper, since it may depend on the type of the project and number of people in a group. It is expected that a project report contains at least 8 pages, and it may be sufficient for an A+ project if some implementational or experimental work has been done.

3.2 Choosing a Project Topic

One approach to choosing a project topic is as follows:

1. Choose an NLP-related problem that is important and interesting in your opinion. You should have some ideas about how it could be solved, and about what interesting results you could obtain by the end of term. The discussed problem should be feasible in this sense, but it should not be trivial.
2. The next step is to search through existing published work and find out about existing solutions on the same problem, or to the closest similar problem. You can start with the textbook.
3. Design your method, implement it, and run experiments; possibly try method variations.
4. Analyze results. Revisit your methodology if needed.
5. Finish the report. Keep writing during the term.

While the above guidelines describe a typical research project in NLP, you can also consider some alternative forms:

Alternative Project Types

- theoretical project: You can focus on establishing a formal framework and proving theoretical results, usually regarding algorithm complexity of some solutions. Still, it may be a good idea to have some experimental results even in this type of project. This kind of project is still very research-oriented.
- implementation: You can put more emphasis on the implementational part of the project. This usually means developing a system prototype with multiple functionalities. In this case, you can devote more space in your report to the design, testing, and user documentation. This kind of project could fit well undergraduate students, and they could choose to implement some algorithm that is well-understood, and not necessarily very relevant to the latest research.

- software evaluation: Choose one or more existing software tools, download them, learn to use them, and use them to solve a problem. Report on your evaluation of the tool, instructions about its usage, advantages and limitations of the tool, and your particular approach. This kind of project is likely more appropriate for undergraduate students.
- survey: The survey format is a critical review of the current research in a narrow sub-area of NLP. If you choose this option, make sure that you do not cover a too wide, or already well-understood area, with published surveys on the topic. I would discourage you to go with this option, unless it is a part of your wider research program. It is difficult to write a good survey paper in a one-term time.

Resources

- NLP Research Links on the course web page
- `http://acl.ldc.upenn.edu/` — ACL Anthology
- Google scholar and other scientific Internet resources
- Dalhousie library

Example Themes

- These are some themes related to current research at Dal CS
- However, you are encouraged to think about other, different areas
- Themes:
  - Analysis of social media data (e.g., Twitter)
  - Author attribution and profiling
  - Sentiment analysis
  - Processing of email data
  - Language, dialect detection; demographic analysis using NLP, etc.

Topics of Some Previous Course Projects

The following are some of the course project topics used in previous terms:

- The Effects of Sentence Simplification as a Preprocessing Step in Text Summarization
- An Analysis of Predictive Text Software and Algorithms
- Extraction of Topics and Clustering of Documents using Topic Modeling Algorithm
- Role of Emoticons for Sentiment Analysis
- Author Profiling for Keyboard Layouts to Understanding User Typing Pattern
- Natural Language Math Problem Assistance Tool
- Canadian Happiness Level Mapping by Using Twitter Data
- Detection of Emotion and Emotion Stimuli in Text
- Text Classification of Consumer Financial Complaints to make Accurate Predictions
- Classifying Political Bias in News Articles
- Sentiment Analysis of Named Entities taken from Live RSS News Feeds
- Sentiment Analysis of Twitter Comments Related to COVID-19
- Performing Stemming for the English language using the Tangled Program Graphs (TPG) Framework
- Classification of Disaster Tweets
- Deep Language Model Representation of Interactive Document Clustering
- Semantic Parsing of Natural Language Queries to Generate SQL Queries (Text-to-SQL)
- Sentiment Analysis With 10-k Forms
- Company Ratings by Running Sentiment Analysis on Social Media Posts
- IMDB Movie Review Sentiment Analysis Using Naives Bayes and SVM Ensemble classification
- Recognizing Pronouns and Misgendering on Social Media Platforms
- Apply NER on Movie Searching Queries

- Using Phylogenetic Distance as a Loss Metric to Compute High Quality DNA Marker Gene Embeddings
- Clustering Movies by Plot Summaries
- Sentiment Analysis of Product Reviews
- Music Playlist Generation using Emoji Sentiment Analysis Transformers
- Fake News Detection using Named Entity Recognition
- Strongly Local Graph Clustering to Identify Biased Semantically Similar Word Sets
- Efficient Parsing in a Highly Structured Language Model
- Malware Detection Using N-gram Method
- Performing Sentiment Analysis on a Live Feed of Tweets during Sports Games
- Social Media Bots Detection using NLP Technique
- Toxicity Detection using NLP
- Aspect-Based Sentiment Analysis
- Amazon Fine Food Review Score Prediction
- Question Answering System on SQuAD Dataset
- Using Extractive Summarization to Highlight Relevant Sentences in Papers
- Generating Short Description of Videos
- Extractive Summarization of Wikipedia Top Search Results
- Email Subject Generation - Different Approaches
- Spam Text Classification using ML Models in Python
- Similar Game Recommendation Based on Game Description
- Sentiment Analysis on Product Reviews
- Conversational Question Answering System
- Detecting Dalhousie's Official Twitter Account's Main Interests
- Classifying Malicious and Benign Domains
- Analysing Word2vec Training Model of Neural Networks for NLP
- Identification and Analysis of Spam on Twitter using a Phone Number based Approach
- Generating Pseudo Code and Comments from Python Source Code
- Analysing Yelp Reviews using NLP Techniques
- Artist and Genre Classification using Song Lyrics
- Classification of Medical Literature in PubMed
- Text Categorization Using Distributional Clustering and Learning Logic
- Next Word Prediction
- Information Extraction of Hurricaine Data Using NLP Techniques
- Determining Emotion from Text using Natural Language Toolkit for Python
- Music Generation with Self-Attention
- Preservation of Pragmatic Meaning Across Common Machine Translation Applications
- Using Google's BERT to Better Understand Low-resource
- Using NLP Techniques to Build a Fake News Detector
- An Alternative Approach to Word Sense Disambiguation
- Health care and Natural Language Processing
- Topic Segmentation Using NLP Techniques in SMS and Email
- The Effects of Sentence Simplification as a Preprocessing Step in Text Summarization
- Entity Extraction from Internet Forums
- Use of Extended Backus-Naur Form and GLL Parsing in NLP
- Finding the Best Dish of a Restaurant using NLP Techniques
- Accuracy Comparison of Machine Learning Algorithms with Different Feature Set Selection
- Multi-document Summarization with Reinforcement Learning and Latent Semantic Analysis
- Ordering of Literary Works by Publication Date, via Supervised Learning
- Implementing a Chat Bot for Emotion Tracking
- Detecting Offensive Language in Tweets
- The Account-making Guru Chatbot
- The Natural Language Google CLI

- Efficient Set Intersection Counting for Phrase Relatedness
- NLP for Clinical Decision Support Systems — A Research Survey
- Generative RNN for Generating Song Lyrics
- A Message-Passing Algorithm Implementation
- Email Searching Program
- Sentiment Analysis of Real-Time Event with Twitter
- Extracting Coarse Phylogenetic Relationships from Biological Literature
- Language Identification of Twitter Data using N-Grams, and Distance Measures
- Opinion Mining for Online Product Reviews
- A Comparison of Sentiment Classification Approaches
- Evaluating the Quality of Product Ideas using Text Classifiction
- Expertise Finiding
- Peanut Bet Parser
- Issues Faced by Human Users when Interacting with Conversational Agents and Implications for Voice User Interface Design
- Naive Bayes Method of Spam Classification
- Detection of Spam in Short Messaging Service in Smartphone using Natural Language Processing
- Algorithmic Assignment of Relevance Values to Conditional Sentence: A Discourse Based Approach
- Document Classification using Deep Boltzmann Machine
- Speech Recognition and Learning Algorithms
- Preprocessing to Improve Performance of Google Tri-Gram-Based Word Similarity
- Classification of Hobbyist Community Inclusion Based on Twitter Posts
- Visualization of Twitter Emotion Based on Naive Bayes Classifier
- Building a Multilingual Model to Understand Relationship between Concepts using Wikipedia
- Implementing a Small Talk Chat Bot
- Exploring Large-Scale NLP with Twitter Hashtag Summarization
- Computer Assisted Scoring of Text Coherence for an Essay to Improve Readability
- Classifying Movie Reviews Using Naive Bayes Text Classification
- Connecting Social Media Posts and Music Lyrics to Perform Sentiment Analysis
- A Sentiment Analysis Plugin for Wordpress
- Tag Twitter Posts with Event Detected
- Implementation of Authorship Profiling: Gender and Geographical Location
- Sentiment Analysis and Authorship Attribution in Games Journalism
- Apache Solr Search Result Ranking on Islandora
- Rapper Recognizer
- Detecting Spam in Massively Multiplayer Role-playing Games
- Application of Evolutionary Computation to Text Summarization
- Evaluating Part-Of-Speech Tagging with Hidden Markov Model in Citar
- Automatically identifying Speedy deletable articles in Wikipedia
- Positivity Text Analysis on Social Media
- 4 Ws Chat Bot
- Analyzing Social Media for Determining Age Groups and Zodiac signs
- Semantic Role Disambiguation In Phrase Representations Of Context-Aware Recursive Neural Networks
- Sentiment analysis of Twitter updates involving athletes
- Generating Chess Portable Game Notation files from articles and books
- Using N-grams to Calculate the Trustworthy, Objective, Complete, and Well-written Levels of Wikipedia Articles
- Character n-gram based analysis and visualization of the sentiment content of text documents
- A Visual Analytics Approach for User-driven Clustering
- Online Spam Classification with Symbiotic Bid-Based Genetic Programming
- An n-gram approach to multiple mood classification of song lyrics.
- Concept Hierarchy Generation from Wikipedia

- An Evaluation of the Efficacy of Restricted Boltzmann Machines for Sentiment Classification
- Hierarchical aspect-based sumamrization and sentiment analysis of online reviews
- Classification of Horse Tack Reviews via an N-gram Model
- Alternative Methods of Input: Speech Recognition in Games
- Naive Bayse and Tf-IDF Hybrid Approach for Spam Filtering
- Comparison of Email Spam Detection Techniques
- Sentiment Analysis of Twitter Data
- Research on Sentiment Analysis of Movie Reviews to Help the Blogger
- Sarcasm detection in online reviews
- Behaviour Analysis and the Theoretical Application of Text Processing Techniques
- Software Evaluation and Analysis of The Stanford Parser
- Character n-gram based analysis and visualization of the sentiment content of text documents
- A Visual Analytics Approach for User-driven Clustering
- Online Spam Classification with Symbiotic Bid-Based Genetic Programming
- An n-gram approach to multiple mood classification of song lyrics
- Concept Hierarchy Generation from Wikipedia
- An Evaluation of the Efficacy of Restricted Boltzmann Machines for Sentiment Classification
- Hierarchical aspect-based summarization and sentiment analysis of online reviews
- Classification of Horse Tack Reviews via an N-gram Model
- Alternative Methods of Input: Speech Recognition in Games
- Naive Bayes and TF-IDF Hybrid Approach for Spam Filtering
- Sentiment Analysis of Twitter Data
- Research on Sentiment Analysis of Movie Reviews to Help the Blogger
- Sarcasm detection in online reviews
- Behaviour Analysis and the Theoretical Application of Text Processing Techniques
- Sentiment Analysis of Real-Time Events on Twitter
- Morphosyntactic Annotation of Esperanto
- Clustering Large Amounts of Noisy Short Twitter Feeds
- Detecting Spam Comments by Analyzing Posted Comments in Blogs
- Algorithms for Linear Time Unification
- Support Vector Machines for Forum Troll Classification
- A Survey of Financial Information Extraction Research
- A Multi-objective evolutionary algorithm for textual data clustering
- A Study on “Co-Training” – a Semi Supervised Learning
- FAQ-based Question Answering
- A Template-Based Approach to Paraphrasing
- Question Classification System using Support Vector Machines
- Comparing Term Extraction Techniques for Man Pages
- A Review of Techniques for Context-Dependent Spell Checking
- An Investigation of a Multi-Objective Genetic Algorithm for Document Clustering
- Implementation of ”Intelligent Copy” (icp) and an Evaluation of WEKA Classification Methods
- MedOnto: Medical Ontology Learning System
- Automatic Keyphrase Extraction for Marine Sciences Text
- To Be Or Not To Be Shakespeare: Using Genetic Algorithm to Build an Authorship Profile for Use in Text Classification
- An Implementation Oriented Introduction to Automatic Speech Recognition
- Part-of-Speech Tagging with Restricted Boltzmann Machines
- Searching for Relevance: A Study on the Relatedness of Articles on Wikipedia
- Sentiment Classification of Conversational Language
- Morphological Analysis of Afan Oromo
- Financial Forecasting with Annual Reports: An Application of N-grams and Readability Scores
- A Character N-gram and Word N-gram Approach to Classification of Literature by Literary Period

- Context Aware Text Repair
- Rule-based Acronym Extraction and Expansion
- e-English Normalization: Converting SMS-Text to Correct English
- Generating bash Scripts from Natural Language
- Blog Generation using a Dialog Agent
- Text Format Segmentation using HMM
- Automatic Inter-Document Link Generation
- Experiments in Character N-gram Based Information Retrieval
- Character N-gram Based Approach to Classification of Movie Reviews
- Compiling Program Code from Natural Language: A Learning Tool for Students Learning Object Oriented Programming
- A Comparison of Rule-Based and Data-Driven Algorithms for Automatic Syllabification of Italian Words
- N-grams and Spam: Using N-gram Analysis to Detect Spam Email Messages
- A Practical Method for Extracting Prefixes and Suffixes of Biological Terms
- Email Authorship Attribution using N-Grams
- Source Text Disambiguation for Improved Machine Translation
- Improving Word Alignments Through Matrix Factorization
- An Unsupervised Approach to Morphological Analysis
- Automatic Composer Recognition – An N-gram-based Approach
- Improving Automatic Term Extraction using Shallow Parsing
- Using Natural Language Queries for E-mail Retrieval
- Context-Dependent Spelling Correction in Languages with No Word Boundaries
- A Comparative Study of Text Categorization using Naive Bayes Classifier with Different Feature Space and Dimensionality Reduction Methodologies
- Semantic Annotation of Conference Notifications in Resource Description Framework
- N-gram Collection using Suffix Arrays
- N-gram-based Classification of Plain-text Privacy Policies
- Comparing Co-Clustering using N-grams and Words
- A Stochastic Method for Software String Translation
- A Simple C++ N-gram Extraction Package
- N-gram-based Hierarchical Text Clustering for PPML Data
- Probable Solutions of Monoalphabetic Substitution Ciphers via Word-Gram Analysis
- Authorship Attribution using Compression and Clustering
- From Natural Language to Java
- Improving Naive Bayes Classification using Natural Language Processing
- Implementing ExtrAns
- A Second-Order Hidden Markov Model for Part-Of-Speech Tagging
- A Study of Connectionist Methods in Natural Language Parsing
- Information Retrieval Performance using Morphology, Part of Speech Tagging, and Semantic Expansion
- Document Clustering with Automatic Term Extraction
- An Approach to Evaluating the Readability of Texts
- Optimising Naive Bayesian Networks for Spam Detection
- Evolved Transformations in Brill's Transformation-based Tagger
- Automatic Term Extraction in Large Text Corpora
- Proper Noun Detection for Search Engines
- An NLP based Topic Modelling and Sentiment Analysis of Small Text Corpus using Word Embedding
- ChatBot which Identifies Depression
- Semantic Conversion of Words using word2vec Model for Machine Learning Applications
- Toxic Comment Classification
- Similar Wine Recommendation Based on Wine Reviews
- Fake News Detection
- Student-Professor Interest Matching

- Machine Comprehension using SQUAD Dataset
- Emotion Analysis of Tweets
- An Intelligent Dialogue System for Ordering at Drive-Thru
- Correlation between the popularity of bitcoin related topics on Twitter and bitcoin's historical market price in bitcoin history
- Selective Encryption for text data using Natural Language Processing
- Application and Algorithms of Topic Modeling
- Attention Networks for Active Learning assisted Information Retrieval
- Music as a Natural Language: Common Natural Language Processing Methodologies and their Applications in a Musical Setting
- Bipartite Anomaly Detection With Probabilistic Metric Space Models
- Feed Me - Voice Based Food Assistant
- Translate Natural Language (English sentences) to Code
- The Lingua Franca of Cricket: How Text Summarization Differs between the UK and India when it Comes to this Beloved Sport
- Evaluation Metric to Measure Semantic Similarity in Medical/Clinical Notes Transcribed by Speech Recognition Tools
- Embedding Cognitive Neuroscience Keywords onto Brain Anatomy using Functional Magnetic Resonance Imaging Activations
- Predictive Text and Spell Check
- Toxic Comment Classification
- Analysis of the "Perfect" Chocolate Chip Cookie
- Learning Speech Synthesis with Speech Recognition
- Naive Bayes for Spam Detection
- Native Language Identification based on English Text
- Extracting Information from Course Syllabi
- seq2seq RNN and LSTM for Machine Translation and Beyond

Part II

Stream-based Text Processing

Slide notes:

- Considering text as a stream of characters, words, and lines of text
- Review of Finite Automata and Regular Expressions
- Review of Unix-style text processing
- Introduction to Perl
- Morphology fundamentals
- N-grams
- Reading: Chapter 2, Jurafsky and Martin

In this part, we will consider language and text as a stream of characters, words, and lines of text, and look into some processing models that are applicable in this environment. We will first refresh out knowledge about finite automata and regular expressions, and some common Unix-based tools that can be used for basic text processing. We will then introduce the Perl programming language as an extension of these tools. Following this, we will introduce elements of morphology, and introduce character and word n-grams.

4 Review of Automata and Regular Expressions

4.1 Finite-State Automata

Finite-State Automata

- Regular Expressions and Regular Languages
- Regular Languages can be described using
    - Regular Expressions
    - Regular Grammars
    - Finite-State Automata (DFA and NFA)
- DFA = Deterministic Finite Automaton
- NFA = Non-deterministic Finite Automaton
- also referred to as Finite-State Machines

Typical Low-level NLP Tasks

- Pre-processing text
- Tokenization
- Sentence Segmentation
- Morphological Processing (e.g., Stemming)
- “Vectorizing” Text
- Information Extraction (simpler cases)
- *and so on*

Example Task: Removing HTML Tags

Deterministic Finite Automaton

- Formally defined as a 5-tuple: $(Q, \Sigma, \delta, q_0, F)$
    - Q is a set of states
    - Σ is an input alphabet
    - $\delta : Q \times \Sigma \rightarrow Q$ is a transition function
    - $q_0 \in Q$ is the start state
    - $F \subset Q$ is a set of final or accepting states
- Graph representation is frequently used
- Consider finite automata for sets of strings:
  `baaa...a!  ha-ha-...-ha  up-up-down-up-down-up-up-...down`

You should notice that we define a DFA (Deterministic Finite Automaton) in such way that for each state and each character from the input alphabet, there is exactly one transition to another state. In some definitions of DFA, such as the one given in the book, this is relaxed by having at most one transition, and if a transition does not exist, it is assumed to lead to a non-listed “dead” state. We are going to explicitly show this “dead” state in our examples.