Computational Linguistics

Computational Morphology

Suhaila Saee & Bali Ranaivo-Malançon

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

August 2014

What is Morphology?

The study of internal structures of word (analysis) and their formation (generation)

- Internal structures is made of different morphological units
- Morphological unit is a smallest unit which a word is made up (Katamba F., 1993)

Example of morphological units:

Boys = boy + s

What is a Computational morphology?

Computational morphology looks at the processing of structure and forms of word, in both their written and spoken form.

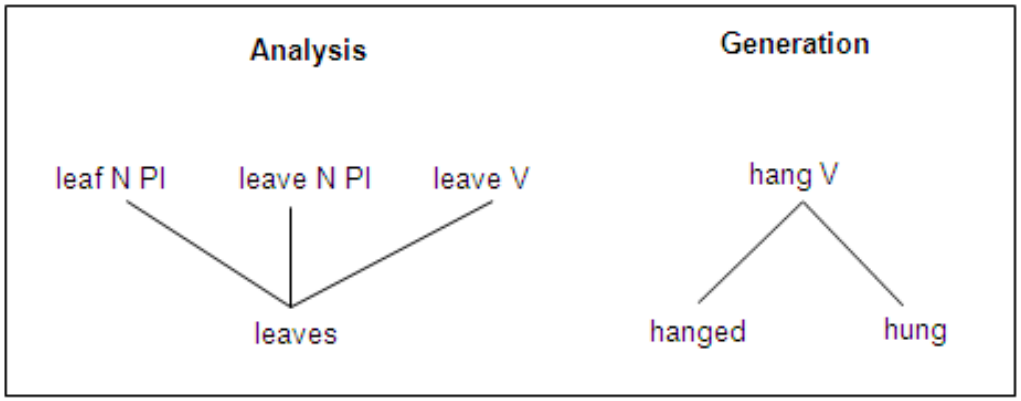

Figure: Elements in morphology analysis and generation

What are the tasks required?

1. **Morphological analysis**: identification of the different components of a full word-form
2. **Morphological disambiguation**: determination of the correct segmentation of a word-form which has multiple analysis
   - Input: *incompatibilities*
   - Analysis 1: *in+ con + patible+ ity+ s*
   - Analysis 2: *incompatibility+ NounPlural*
3. **Morphological generation**
   - Input: *incompatibility+Plural*
   - Generation: *incompatibilities*

Morphophonology / Morphographemy

- **Alternations** (or **modifications**) may appear when two morphological units are combined
- Morphophonology/Morphographemy deals with the rules governing these alternations

Examples:
easy+er becomes *easier*
fox+s becomes *foxes*

Morphotactics

- or the syntax of words
- Morphological units do not combine freely: they follow some rules
  - The five morphemes in *internationalisation* can only be combined in one way:
    - inter-nation-al-ise-ation
  - The other combinations are illicit
    - *inter-nation-ise-al-ation
    - *inter-nation-al-ation-ise

Common techniques

1. **Two-level morphology** *(rule-based)*
   - Uses individual hand-crafted finite state models to represent context-sensitive stem-changes
2. **Morphology induction** *(machine learning)*
   - The process of inferring different types of morphological information from annotated or raw data
     - Segmenting words into their constituent morphemes (root/stem and affixes)
     - Segmenting words to find the root/stem
   - Often deals with inflection only

Two-level morphology

- A general computational model for word-form recognition and production (Koskenniemi, 1983)
- **Features:**
  1. able to handle complex languages such as Finnish, Turkish and German (Karttunen & Beesley, 2001)
  2. most of complex and low-density languages applied two-level morphology due to its unique features (Karttunen & Beesley, 2005):
     - language independent
     - bi-directional that is applicable for analysis and generation
     - has two components which are lexicon and a set of two-level rules
     - flexibility to all complex languages including U-RL

Two-level morphology (cont'd)

- **How it works?**
  symbol-to-symbol correspondence between two-levels:
  1. lexical (how words are formed from morphemes in lexicon)
  2. surface (how words appear in text) levels

Lexical level:	f o x + plural ('+' is a morpheme boundary)
Surface level:	f o x e s
Two-level rules:	+:e ↔ x:x _ 0:s

- **Example** see here

Two-level morphology (cont'd)

- **Drawbacks:**
  1. requires a lot of linguistic works and expertise
     Wintner (2007) found that this approach also:
  2. requires huge lexicon and rules to perform best results and
  3. time consuming when creating or updating rules and lexicon of new or current languages
  4. lead to high cost in maintenance

Two-level morphology: architecture

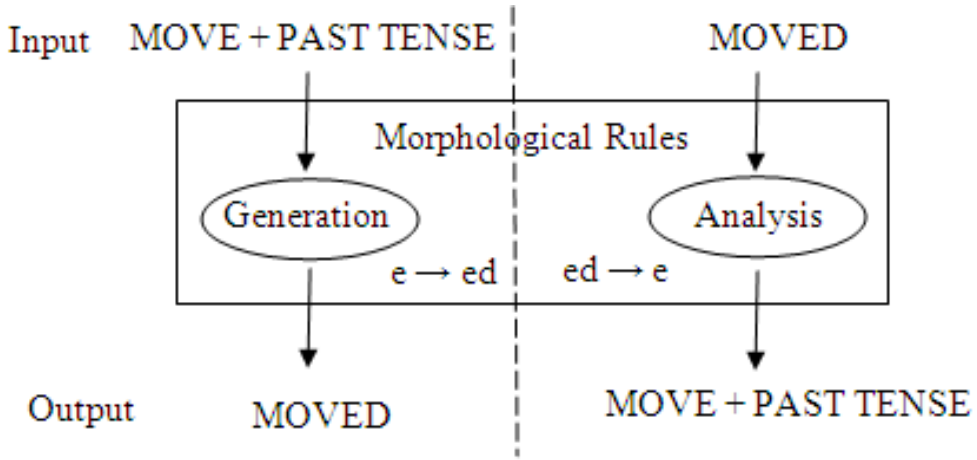

Figure: General idea of two-level morphology

Morphology Induction

- An induction process applying machine learning technique
- **Features:**
  1. An unsupervised learning: learns the morphology from raw set data without any direct access to the particular structure (Goldsmith, 2001; Monson, 2008; Creutz, 2006; Hammarström, 2009)
  2. Three approaches: a) group and abstract, b) frequency and border, and c) features and classes (Hammarström, 2007)
  3. frequency and border approach (approach (b)) - the most common and widely used
  4. minimize human expertise control

Morphology Induction (cont'd)

- **How it works?**
  inducing morphological rules from the corpus:
  1. input - a text
  2. minimum description length (MDL) technique - to induce segmentation between stems and affixes.
  3. output - list of affixes and stems
- **Drawbacks:** producing poor result for sparse-data
- **Possible solution:** Semi-automatic machine learning

Computational morphology subfields

- Stemming
- Lemmatisation

What is stemming?

- The process that strips off the ending of words
- The output is called the **stem**
- The tool is called a **stemmer**
- Stemming is widely used in the information retrieval domain but it is also needed in machine translation
- For instance:

Input:	duties
Specified ending:	s
Output stem:	dutie

Input:	duties
Specified ending:	es
Output stem:	duti

Input:	duties
Specified ending:	ies
Output stem:	dut

What is lemmatisation?

- A process of reducing **word-forms** to their corresponding **lemma**
- The tool is called a **lemmatiser**
- A lemma is the word that is used as the head of a definition in a dictionary
- For instance:

Input:	duties
Output lemma:	duty

Noun: duty ♫ doo-tee

1. The social force that binds you to the courses of action demanded by that force
   "we must instill a sense of duty in our children"; "every right implies a responsibility; every opportunity, an obligation; every possession, a duty"
2. Work that you are obliged to perform for moral or legal reasons
   "the duties of the job"
3. A government tax on imports or exports
   "they signed a treaty to lower duties on trade between their countries"

Derived:

Adjective duteous

Tool: Linguistica

- An *unsupervised learning morphological analyser* ▸ Linguistica
- Underlying model: Minimum Description Length
- No distinction is made between inflectional and derivational affixes
- Does not handle irregular morphology
- Free program written in C++ for Windows, Mac OS X, and Linux

Linguistica: Input & Output

- INPUT: list of words (from 5,000 words to 500,000 words)
- OUTPUTS
  - List of stems, prefixes, and suffixes
  - List of signatures

Linguistica: Signature

- A **Signature** = a list of all suffixes (prefixes) appearing in a given corpus with a given stem
- A stem in a corpus has a unique signature
- A signature has a unique set of stems associated with it

Tool: Porter Stemmer

- Established stemmer and widely used in text processing
- The stemmer is available in many languages
- Encodings of the algorithm are available in various programming languages e.g., Java, Perl, Phyton and Csharp
- The official website of Porter stemmer

Tool: Paice/Husk Stemmer

- The stemmer is a conflation based iterative stemmer
- Known as a strong and aggressive stemmer compares to Porter stemmer
- For details, go to ▸ Paise/ Husk stemmer

Other Stemmer tools

- Lancaster University provides an official web-site for various implementations of the stemming algorithm, together with links to other useful stemming resources
- For further information, click to ▸ Other stemming algorithms
- For demo purposes, NLTK also provides demo for the stemmers ▸ NLTK stemmer

Tool: WordNet Lemmatiser

- The NLTK Lemmatization method is based on WordNet's WordNet corpus and uses built-in morphy function
- The program is available in Phyton
- Go to WORDNET LEMMATISER to download the module from NLTK package ▸ download
- For demo of the lemmatizer, click to ▸ Demo

Evaluation of morphology analsis

- Accuracy = number of inflections for which the correct root was found, divided by the total number of inflections in the test set
- Precision = number of inflections for which the root was correctly identified, divided by the total number of inflections for which a root was identified at all
- Precision is used to evaluate the performance of the model
- Coverage = number of inflections that were aligned with a (correct or incorrect) root, divided by the total number of inflections in the test set

Evaluation of stemming

- Evaluation approaches use in stemming
  1. error counting - counting the numbers of two kinds of errors (under-stemming and over-stemming) that occur during stemming
  2. stemmer strength - a 'strong' or 'heavy' stemmer merges a much wider variety of forms
  3. inter-stemmer similarity - comparing two separate stemming algorithms through the output they produce
- For further information, click to ▸ evaluation techniques

References

Katamba Francis. (1993). Morphology. MacMillan Press Ltd.

Karttunen, L., & Beesley, K. R. (2005). Twenty-five years of finite-state morphology. Inquiries Into Words - a Festschrift for Kimmo Koskenniemi on his 60th Birthday, 71–83. Citeseer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.4114&rep=rep1&type=pdf.

Koskenniemi, K. (1983). Two-Level Morphology: A General Computational Model for Word-form Recognition and Production. PhD Thesis. University of Helsinki.

Wintner, S. (2007). Strengths and weaknesses of finite-state technology: a case study in morphological grammar development. Natural Language Engineering, 14(04), 457-469. DOI: 10.1017/S1351324907004676.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. Computational linguistics, 27(2), 153–198. MIT Press. Retrieved from http://www.mitpressjournals.org/doi/abs/10.1162/089120101750300490.

Hammarstrom, H. (2007). A Survey and Classification of Methods for (Mostly) Unsupervised Learning of Morphology. Proceeding of 16th Nordic Conference of Computational Linguistics (NODALIDA 2007), (Joakim Nivre; Heiki-Jaan Kaalep; Kadri Muischnek and Mare Koit (Eds.), 292–296. DOI: 10.1007/s11227-009-0338-x.

Hammarstrom, H. (2009). Unsupervised Learning of Morphology and the Languages of the World. Thesis for the Degree of Doctor of Engineering. Chalmers University of Technology and Goteborg University. Retrieved from http://guoa.ub.gu.se/dspace/handle/2077/21418.

Monson, C. (2008). ParaMor: from Paradigm Structure to Natural Language Morphology Induction. Retrieved from http://www.cs.cmu.edu/ cmonson/Thesis/Thesis-March14-2008-ForFullCommittee.pdf.