Morfessor 2.0: Toolkit for Statistical Morphological Segmentation – Model

Probabilistic Model definition

- Full description in (Virpioja, 2012; Virpioja et al., 2013)
- Generative model
  \[ p(\text{analyses words} | \theta) \]
The model generates pairs of words and analysis (the segmentation of a word into morphs)
- Tokenization function
  \[ a = \phi(w; \theta) \]
- Cost derivation

\[ \theta_{\text{MAP}} = \arg \max_{\theta} p(\theta | D_{\text{data}}) \]
\[ L(\theta, D) = -\log p(\theta) - \log p(D | \theta) \]

The data \( D \) is a list of (non-segmented) words to learn the model from in unsupervised manner.

Data Likelihood

\[ \log p(D | \theta) = \sum_{j=1}^{N} \log p(W = w_j | \theta) \]
\[ = \sum_{j=1}^{N} \log \sum_{a \in \Phi(w_j)} p(A = a | \theta), \]

Morfessor Baseline assumes independence of words. Also, only valid tokenisations of need to be considered. Morfessor selects only one tokenisation (analysis) for each word at a time, by introducing a hidden variable \( Y \).

\[ \log p(D | \theta, Y) = \sum_{j=1}^{N} \log p(y_j | \theta) \]
\[ = \sum_{j=1}^{N} \log p(m_{j,1}, \ldots, m_{j,y_j}; \theta_{\text{selected analysis}}) \]

Prior

(Creutz and Lagus, 2007) The parameters of Morfessor Baseline encode the properties of the morph lexicon:

\[ p(\theta) = p(\mu) \times \mu! \times p(\text{morphs}) \times p(\text{morph permutations}) \times p(\text{properties}(m_1), \ldots, \text{properties}(m_p)). \]

\[ p(\sigma_i) = p(L = |\sigma_i|) \frac{1}{|\sigma_i|} \prod_{j=1}^{N_P} p(C = \sigma_{ij}) \]

Algorithm

(Creutz and Lagus, 2002)

function LOCALBATCHTRAIN(D, \epsilon)

\[ \theta, Y \leftarrow \text{INITMODEL}(D_W) \]

\[ L_{\text{old}} \leftarrow \infty \]
\[ L_{\text{new}} \leftarrow L(D_W, \theta, Y) \]

while \( L_{\text{new}} < L_{\text{old}} - \epsilon \) do

\[ j \leftarrow \text{RANDOMPERMUTATION}(1, \ldots, N) \]

for \( j \in J \) do

\[ \theta_j, Y \leftarrow \text{LOCALSEARCH}(w_j, D, \theta, Y) \]

\[ L_{\text{old}} \leftarrow L_{\text{new}} \]
\[ L_{\text{new}} \leftarrow L(D, \theta, Y) \]

return \( \theta, Y \)

Likelihood weighting and Semi-supervised training

Likelihood weighting with \( \alpha \) (Virpioja et al., 2011)

\[ L(\theta, D) = -\log p(\theta) \]
\[ - \alpha \log p(D | \theta) \]
\[ - \beta \log p(D_{\text{Annotated data}} | \theta). \]

For semi-supervised learning another term is added to the cost, the likelihood of a set of annotations coming from the model. Also here a weight \( \beta \) is introduced to control the effect.

References


Morfessor 2.0: Toolkit for Statistical Morphological Segmentation – Codebase

Previous: Morfessor 1.0
- Written in perl
- Limited utf-8 support
- Older codebase, unsuitable for extensions

Morfessor 2.0
- Complete rewrite
- Inclusion of many previously published features
- Extensible for new features and algorithms

Usage
- Library interface
  - Directly use Morfessor from python scripts
- Command line interface
  - Run training evaluation and segmentation from the command line
  - Almost complete coverage of Morfessor functionality

Features
- On-line training
- Training speed-up with random skips
- Frequency threshold and dampening for words in training data
- Possibility to weight training data likelihoods
- Optimization of data likelihood weight based on development set
- Optimization of data likelihood weight based on average morph type length
- Optimization of data likelihood weight based on average morphs / word

Items in grey will be released in Morfessor 2.1

Implementation
- Python
  - Runs on Python2, Python3 and Pypy interpreters
  - Best performance: Pypy
- Unit-agnostic code
  - Split words into morphs, or sentences into phrases

Demo
- Build on top of library interface
- Will be available online in future

Distribution
- Source code available at GitHub
- Packages available from the PYthon Package Index (Pypi)
- BSD Open Source License

Upcoming developments
- Morfessor 2.0 based model with morphotactic constraints (categories)
- Data selection techniques

Acknowledgements
The authors have received funding from the EC’s 7th Framework Programme (FP7/2007–2013) under grant agreement n°287678 and the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant n°251170) and the LASTU Programme (grants n°256887 and 259934). The experiments were performed using computer resources within the Aalto University School of Science "Science-IT" project.

Links
- GitHub: https://github.com/aalto-speech/morfessor
- Pypi: pip install morfessor
- Documentation: https://pypi.python.org/pypi/Morfessor

Aalto University
Peter Smit, Sami Virpioja, Stig-Arne Grönroos and Mikko Kurimo
Aalto University
Aalto University - Morpho project
The work on Morfessor is funded through multiple projects. The work on Morfessor 2.0 were done in the scope projects mentioned below. For updates on Morfessor, go to http://www.cis.hut.fi/projects/morpho/ and subscribe to the mailing list!

COIN

CompuBrain: Computational modelling of brain's language
A project between O.V. Lounasmaa Laboratory and Departments of Information and Computer Science and Signal Processing and Acoustics at Aalto University, funded by Academy of Finland. Aims for brain-based models of language by bringing together expertise on neuroimaging and computational modelling of language.

Morpho challenge 2014
After the challenges in 2005 and 2007-2010 Aalto University plans to organize a new Morpho challenge. Possible Tasks:
- Generation of new words for language modeling and spelling correction
- Bilingual morphological analysis for machine translation.
Starts in 2014, as soon as task and data are ready. More information http://research.ics.aalto.fi/events/morphochallenge/. Subscribe to the mailing list!