A Language-Independent Unsupervised Model for Morphological Segmentation

Vera Demberg

Institute for Communicative and Collaborative Systems (ICCS)
University of Edinburgh

October 25, 2006

Overview

Why analyse words morphologically?

- Motivation
  - Decrease data sparseness
  - Smaller lexica
  - Relate words
- Applications
  - Machine Translation
  - Speech Recognition
  - Text-to-Speech Systems
  - Information Retrieval
  - Question Answering

Why analyse words morphologically?

- Motivation
  - Decrease data sparseness
  - Smaller lexica
  - Relate words
- Applications
  - Machine Translation
  - Speech Recognition
  - Text-to-Speech Systems
  - Information Retrieval
  - Question Answering

Why use an unsupervised method?

Unsupervised vs. Rule-based

+ Less domain-dependent
+ Lower development cost
+ Good generalizability to new languages
– Quality

Types of Affixes

- **Prefixes**
  un-do, re-open
- **Suffixes**
  work, work-ing, work-ed, work-s
- **Infixes**
  sulat 'write', s-um-ulat 'wrote', s-in-ulat 'was written' (Tagalog)
- **Circumfixes**
  ge-mach-t 'done', ge-sproch-en 'said' (German)
- **Stem Variation**
  - ablauting: *sing, sang, sung*
  - umlauting: *Garten, Gärten*
  - vowel harmony: *ev – evler, kitap – kitaplar* (Turkish)
  - deletion / insertion: *care, caring; panic, panicked; travel, travelling*

Morphological Processing Tasks

- **Segmentation**

  Trainingssprünge → *Training+s+sprüng+e*

- **Lemmatization**

  Trainingssprünge → *Trainingssprung*

- **Semantic relations**

  correlate: *Sprünge – Sprungs – Sprung – Sprüngen*

- **Automatic induction of affixational paradigms**

  {*-s -ed -ing*}
  {*-en -ung -te -t -e -end -est -et -st -ten -tet*}
  {*-baren -lich -barer*}
  {*-er -e -erei -t -ern*}

Q: How to find and relate all affixes that signify e.g. past tense?

Morphological Processing Tasks

- **Segmentation**

  Trainingssprünge → *Training+s+sprüng+e*

- **Lemmatization**

  Trainingssprünge → *Trainingssprung*

- **Semantic relations**

  correlate: *Sprünge – Sprungs – Sprung – Sprüngen*

- **Automatic induction of affixational paradigms**

  {*-s -ed -ing*}
  {*-en -ung -te -t -e -end -est -et -st -ten -tet*}
  {*-baren -lich -barer*}
  {*-er -e -erei -t -ern*}

Q: How to find and relate all affixes that signify e.g. past tense?

Previous Approaches

- Letter Successor Variety / Conditional Entropy
  Harris [1955]; Hafer and Weiss [1974]; Saffran *et al.* [1996]; Bordag [2006]; Bernhard [2006]; Keshava and Pitler [2006]
- Phonological Relationships between Related Words
  Neuvel and Fulop [2002]; Schone and Jurafsky [2001, 2000]
- Minimum Description Length
  Goldsmith [2001]; Creutz and Lagus [2006]

Typical Problems

- Frequently co-occurring letter sequences
  schw, qu, th
- Over-segmentation
  sw+ing, t+rain, t+own, t+weak, c+hair
- Splitting at stem variations
  Spr+ung, Spr+ünge, spr+ingen, spr+ang, spr+änge
- Violation of morphotactic constraints
  ed+ward, s+e+e+gang, t+röstung

Language-dependency and Unsupervisedness

- Constraints can lead to high performance gains
  - lengths of affixes and stems
  - properties of certain letters
  - structure of words
- Development cost for new language
- Modelling morphotactics
- Underlying assumptions:
  concatenative vs. non-concatenative morphology

The original RePortS algorithm (Keshava and Pitler [2006])

Three steps:

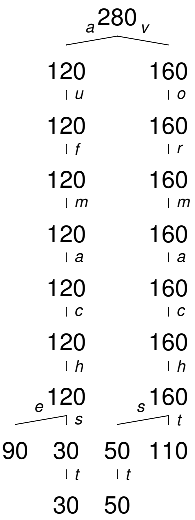
1. Building up data structure
2. Finding affixes
3. Segmenting words

Step 1: Data structure

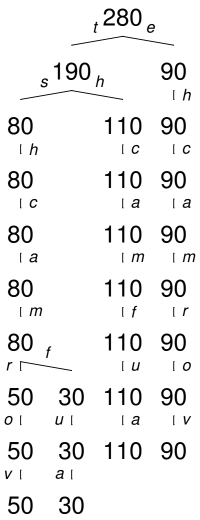
(a) lexicon

toy lexicon:	
⋮	
aufmacht	90
aufmachst	30
vormache	110
vormachst	50
⋮	

(b) forward tree



(c) backward tree



Step 2: Finding affixes

word of form: "$\alpha AB\beta$", example: $\underbrace{wor}_{\alpha}\underbrace{k}_{A}\underbrace{i}_{B}\underbrace{ng}_{\beta}$

find suffix $B\beta$	find prefix αA
1. αA in corpus	1. βB in corpus
2. $P_f(A\|\alpha) \approx 1$	2. $P_b(B\|\beta) \approx 1$
3. $P_f(B\|\alpha A) < 1$	3. $P_b(A\|B\beta) < 1$

```
Ranking algorithm
if (cond. satisfied)
   score += 19;
else
   score -= 1;
```

Language-specific assumptions:

- all stems are valid words in the lexicon.
- affixes occur at the beginning or end of words only.
- affixation does not change stems.

Step 2: Finding affixes

word of form: "$\alpha AB\beta$", example: $\underbrace{wor}_{\alpha}\underbrace{k}_{A}\underbrace{i}_{B}\underbrace{ng}_{\beta}$

find suffix $B\beta$	find prefix αA
1. αA in corpus	1. βB in corpus
2. $P_f(A\|\alpha) \approx 1$	2. $P_b(B\|\beta) \approx 1$
3. $P_f(B\|\alpha A) < 1$	3. $P_b(A\|B\beta) < 1$

```
Ranking algorithm
if (cond. satisfied)
  score += 19;
else
  score -= 1;
```

Language-specific assumptions:

- all stems are valid words in the lexicon.
- affixes occur at the beginning or end of words only.
- affixation does not change stems.

Step 3: Segmenting words

while *length*(*stem*) $>$ *length*(*word*)$/2$ **or** no matching affixes **do**
    bestP $\Leftarrow 1$
    for all *affix* $\in$ *affixList* **do**
        if stem = substr.affix **and** $P_{trans}(substr, affix) < bestP$ **then**
            bestP $\Leftarrow P_{trans}(substr, affix)$
            bestAffix $\Leftarrow$ *affix*
        end if
    end for
    stem $\Leftarrow$ *substr*
    store *bestAffix*
end while

Advantages and Problems of this simple approach:

+ most probable single affix given rest word is peeled off
– no context taken into account
– morphotactically impossible segmentations occur often
– cannot segment beyond an unknown morpheme

Step 3: Segmenting words

while *length*(*stem*) $>$ *length*(*word*)$/2$ **or** no matching affixes **do**
    $bestP \Leftarrow 1$
    for all $affix \in affixList$ **do**
        if stem = substr.affix **and** $P_{trans}(substr, affix) < bestP$ **then**
            $bestP \Leftarrow P_{trans}(substr, affix)$
            $bestAffix \Leftarrow affix$
        end if
    end for
    $stem \Leftarrow substr$
    store *bestAffix*
end while

Advantages and Problems of this simple approach:

+ most probable single affix given rest word is peeled off
– no context taken into account
– morphotactically impossible segmentations occur often
– cannot segment beyond an unknown morpheme

Inhibitively low recall low for German / Turkish / Finnish

Stems are often no valid words and therefore not contained in corpus.

example: "abhol"

German corpus:
abholst
abholen
abholt
abhole
Abholung

Why does it work in English?

Consider example of affixes ***ism***, ***ance***, ***ation***:

English	German
*Catholic–Catholic**ism***	*kathol**isch**–Kathol**izismus**–Kathol**ik***
*accept–accept**ance***	*akzept**ieren**–Akzept**anz***
*adapt–adapt**ation***	*adapt**ieren**–Adapt**ation***

Inhibitively low recall low for German / Turkish / Finnish

Stems are often no valid words and therefore not contained in corpus.

example: "abhol"

German corpus:
abholst
abholen
abholt
abhole
Abholung

Why does it work in English?

Consider example of affixes ***ism***, ***ance***, ***ation***:

English	German
*Catholic–Catholic**ism***	*katholi**sch**–Katholi**zismus**–Kathol**ik***
*accept–accept**ance***	*akzept**ieren**–Akzept**anz***
*adapt–adapt**ation***	*adapt**ieren**–Adapt**ation***

Acquire List of High Quality Stem Candidates

1. **Create a list of candidate stems**

studentenaus	{schuß weise weis schusses schüsse schuss}
geschäftsflug	{hafen zeugen häfen zeuge hafens verkehr verkehrs}
eingreif	{truppe werte trupps mandat trupp kräfte verband ...}
	+{en t e er est et st}
exekutier	{t en ten te ung e ter er end est et st tet}
runtersch	{lucken iebt ubsen icken aute}

2. **Assess the stem candidates**
   - accept all candidates with lexicon words only
   - rank by average frequency of non-lexicon words
3. **Define threshold for ranked list**
   - 0.3 for German / English / Finnish, 0.6 for Turkish

Context-sensitive segmentation

1. **Generate all possible segmentations**
   - locally most probable suffix not necessarily globally best solution
   - less under-segmentation if "transitional prob. < 1" condition dropped
2. **Heuristic pruning**
   - remove all analyses that contain unknown segments
     if there is at least one analysis with only known segments
   - disprefer short unknown segments
3. **Ranking using language model**
   - bi-gram model trained on simple segmentations (bootstrapping)
   - divide probabilities by # of segments to reduce bias towards analyses with few segments
   - biased towards simple segmentation

Q: How to learn morphotactics? HMM? What units?

Context-sensitive segmentation

1. **Generate all possible segmentations**
   - locally most probable suffix not necessarily globally best solution
   - less under-segmentation if "transitional prob. < 1" condition dropped
2. **Heuristic pruning**
   - remove all analyses that contain unknown segments
     if there is at least one analysis with only known segments
   - disprefer short unknown segments
3. **Ranking using language model**
   - bi-gram model trained on simple segmentations (bootstrapping)
   - divide probabilities by # of segments to reduce bias towards analyses with few segments
   - biased towards simple segmentation

Q: How to learn morphotactics? HMM? What units?

Stem Variation Detection Method

1. **Clustering**

studentenaus	{schuß weise weis schusses schüsse schuss}
geschäftsflug	{hafen zeugen häfen zeuge hafens verkehr verkehrs}
eingreif	{truppe werte trupps mandat trupp kräfte verband ...}
	+{en t e er est et st}

2. **Edit Distance**
   - edit-dist(schuss – schüsse) = 3
     pattern: u $\rightarrow$ ü..e
   - edit-dist(hafen – häfen) = 2
     pattern: a $\rightarrow$ ä

3. **Ranking**
   count frequencies of patterns with small edit distance.

Stem Variation

freq.	diff.	examples
1682	a ä..e	sack-säcke, brach-bräche, stark-stärke
344	a ä	sahen-sähen, garten-gärten
321	u ü..e	flug-flüge, bund-bünde
289	ä a..s	verträge-vertrages, pässe-passes
189	o ö..e	chor-chöre, strom-ströme, ?röhre-rohr
175	t en	setzt-setzen, bringt-bringen
168	a u	laden-luden, *damm-dumm
160	ß ss	läßt-lässt, mißbrauch-missbrauch
[. . .]		
136	a en	firma-firmen, thema-themen
[. . .]		
2	ß g	*fließen-fliegen, *laßt-lagt
2	um o	*studiums-studios

Integration of Stem Variation Component

Future work:

Integrate stem variation information

- **affix acquisition**
  generate other forms using patterns
  and see whether those are contained in dictionary
- **word segmentation**
  generate equivalence sets for transitional probabilities
- **lemmatization**
  identify semantically related words

Q: What is an efficient way to generate the stem variations?

Integration of Stem Variation Component

Future work:

Integrate stem variation information

- **affix acquisition**
  generate other forms using patterns
  and see whether those are contained in dictionary
- **word segmentation**
  generate equivalence sets for transitional probabilities
- **lemmatization**
  identify semantically related words

Q: What is an efficient way to generate the stem variations?

Evaluation of effect of versions

Lang.	alg.version	F-Meas.	Prec.	Recall
Ger	original	59.2%	71.1%	50.7%
	stems	68.4%	68.1%	68.6%
	n-gram seg.	**68.9%**	73.7%	64.6%
Eng	original	**76.8%**	76.2%	77.4%
	stems	67.6%	62.9%	73.1%
	n-gram seg.	75.1%	74.4%	75.9%
Tur	original	54.2%	72.9%	43.1%
	stems	61.8%	65.9%	58.2%
	n-gram seg.	**64.2%**	65.2%	63.3%
Fin	original	47.1%	84.5%	32.6%
	stems	56.6%	74.1%	45.8%
	n-gram seg.	**58.9%**	76.1%	48.1%
	max-split*	**61.3%**	66.3%	56.9%

Comparison to other systems (German)

morphology	F-Meas.	Prec.	Recall
SMOR-disamb2	83.6%	87.1%	80.4%
ETI	79.5%	75.4%	84.1%
SMOR-disamb1	71.8%	95.4%	57.6%
RePortS-lm	68.8%	73.7%	64.6%
RePortS-stems	68.4%	68.1%	68.6%
Bernhard	63.5%	64.9%	62.1%
Bordag	61.4%	60.6%	62.3%
orig. RePortS	59.2%	71.1%	50.7%
Morfessor 1.0	52.6%	70.9%	41.8%

How does morphological information help grapheme-to-phoneme conversion?

Pronunciation of words is sensitive to morphological boundaries

- English example: *loophole*
  /ˈluːfəʊl/ vs. /ˈluːpˌhəʊl/
- *Sternanisöl*
  /ˈʃtəʀnʔaniːsʔœːl/ vs. /ʃtəʀˈnaːnizœl/
- *Röschen*
  /rœʃən/ vs. /rœːsçən/
- *vertikal* vs. *vertickern*
  /v/ vs. /f/
- *Weihungen* vs. *Gen*
  /ə/ vs. /eː/

Morphological Systems for g2p conversion

morphology	F-Measure	PER AWT
CELEX	100%	2.64%
ETI	79.5%	2.78%
SMOR-disamb2	83.0%	3.00%
SMOR-disamb1	71.8%	3.28%
RePortS-lm	68.8%	3.45%
no morphology		3.63%
orig. RePortS	59.2%	3.83%
Bernhard	63.5%	3.88%
RePortS-stem	68.4%	3.98%
Morfessor 1.0	52.6%	4.10%
Bordag	64.1%	4.38%

Table: Evaluation on manually annotated CELEX and a grapheme-to-phoneme conversion task using the Add-WordTree decision tree (Lucassen and Mercer [1984]).

Limitations (1)

Typical errors:

- **Over-segmentation of short words**
  - *ab-st-e-ig-e* vs. *ab-steig-e* ‘dismount’.
- **Under-segmentation of long words**
  - *Ab-ge-ordnet-e* ‘deputy’ vs. *Abgeordnet-en-haus-e* ‘assembly building’
  - *Ab-blend-licht* ‘dim light’ vs. *Abblendlicht-e*
- **Data sparseness in morphologically complex languages**
  - segmentation step does not look beyond unknown segments
  - sparse trees

Limitations (1)

Typical errors:

- **Over-segmentation of short words**
  - *ab-st-e-ig-e* vs. *ab-steig-e* 'dismount'.
- **Under-segmentation of long words**
  - *Ab-ge-ordnet-e* 'deputy' vs. *Abgeordnet-en-haus-e* 'assembly building'
  - *Ab-blend-licht* 'dim light' vs. *Abblendlicht-e*
- **Data sparseness in morphologically complex languages**
  - segmentation step does not look beyond unknown segments
  - sparse trees

Limitations (1)

Typical errors:

- **Over-segmentation of short words**
  - *ab-st-e-ig-e* vs. *ab-steig-e* ‘dismount’.
- **Under-segmentation of long words**
  - *Ab-ge-ordnet-e* ‘deputy’ vs. *Abgeordnet-en-haus-e* ‘assembly building’
  - *Ab-blend-licht* ‘dim light’ vs. *Abblendlicht-e*
- **Data sparseness in morphologically complex languages**
  - segmentation step does not look beyond unknown segments
  - sparse trees

Limitations (2)

- **Inner-word affixes**
  only affixes that occur at the edges of words are found
  $\rightarrow$ run step again on stem candidates
  (preliminary result: +2% f-score for Turkish)

- **Interleaving Prefixation and Suffixation Processes**
  - currently totally independent
  - if related can cope with circumfixes
  - capture info from co-occurrence of prefixes and suffixes

Q: Can you think a better way to find inner-word affixes?

Limitations (2)

- **Inner-word affixes**
  only affixes that occur at the edges of words are found
  $\rightarrow$ run step again on stem candidates
  (preliminary result: +2% f-score for Turkish)

- **Interleaving Prefixation and Suffixation Processes**
  - currently totally independent
  - if related can cope with circumfixes
  - capture info from co-occurrence of prefixes and suffixes

Q: Can you think a better way to find inner-word affixes?

Limitations (2)

- **Inner-word affixes**
  only affixes that occur at the edges of words are found
  $\rightarrow$ run step again on stem candidates
  (preliminary result: +2% f-score for Turkish)

- **Interleaving Prefixation and Suffixation Processes**
  - currently totally independent
  - if related can cope with circumfixes
  - capture info from co-occurrence of prefixes and suffixes

Q: Can you think a better way to find inner-word affixes?

Summary

I proposed:

- Stem candidate generation step
- Filter for segmentation
- Method for detecting stem variation

I found:

- Significant improvement in recall
- Good performance on German, English, Turkish
- Still low recall on Finnish
- Only method that beats no-morph. baseline on German g2p task

Questions?

Q: How to find and relate all affixes that signify e.g. past tense?

Q: How to learn morphotactics? HMM? What units?

Q: What is an efficient way to generate the stem variations?

Q: Can you think a better way to find inner-word affixes?

Q: Are the trees actually the kind of data structure we want? (Inherent bias for prefixes and suffixes?)

Questions?

Q: How to find and relate all affixes that signify e.g. past tense?

Q: How to learn morphotactics? HMM? What units?

Q: What is an efficient way to generate the stem variations?

Q: Can you think a better way to find inner-word affixes?

Q: Are the trees actually the kind of data structure we want? (Inherent bias for prefixes and suffixes?)

Bibliography

Delphine Bernhard. Unsupervised morphological segmentation based on segment predictability and word segments alignment. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 19–24, Venice, Italy, 2006.

Stefan Bordag. Two-step approach to unsupervised morpheme segmentation. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 25–29, Venice, Italy, 2006.

Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. In *ACM Transaction on Speech and Language Processing*, 2006.

John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 2001.

Margaret A. Hafer and Stephen F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval 10*, pages 371–385, 1974.

Zellig Harris. From phoneme to morpheme. *Language 31*, pages 190–222, 1955.

Samarth Keshava and Emily Pitler. A simpler, intuitive approach to morpheme induction. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 31–35, Venice, Italy, 2006.

J. Lucassen and R. Mercer. An information theoretic approach to the automatic determination of phonemic baseforms. In *ICASSP 9*, 1984.

Sylvain Neuvel and Sean Fulop. Unsupervised learning of morphology without morphemes. In *Proc. of the Wshp on Morphological and Phonological Learning, ACL Pub.*, 2002.

Jenny R. Saffran, Elissa L. Newport, and Richard N. Aslin. Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35:606–621, 1996.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the Computational Natural Language Learning Conference*, pages 67–72, Lisbon, 2000.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of inflectional morphologies. In *Proceedings of the North American chapter of the Association for Computational Linguistics (NAACL-2001)*, 2001.