

Using Hand-Written Rewrite Rules to Induce Underlying Morphology

Michael A. Tepper

University of Washington
Department of Linguistics

Unsupervised Morpheme Analysis – Morpho Challenge 2007

Outline

Introduction

Morphemes and Allomorphs
Examples from Challenge Languages

Procedure Overview

Rewrite Rules
Stage A :: Basic EM
Stage B :: Split Segments

Results

F-Measure Results

Summary

Outline

Introduction

Morphemes and Allomorphs
Examples from Challenge Languages

Procedure Overview

Rewrite Rules
Stage A :: Basic EM
Stage B :: Split Segments

Results

F-Measure Results

Summary

Definitions

We consider **morphemes** to be...

- ▶ basic units of grammar with no internal structure which may be composed together to form words
- ▶ realized as sequences of linguistic symbols (phones and/or letters)

Morphemes may be rendered differently in different contexts:

- ▶ lexical context: /s/ → en, as in *oxen*
- ▶ phonological/orthographic context: /s/ → es, as in *dresses*

Morphological *variants* are known as **allomorphs**

Definitions

We consider **morphemes** to be...

- ▶ basic units of grammar with no internal structure which may be composed together to form words
- ▶ realized as sequences of linguistic symbols (phones and/or letters)

Morphemes may be rendered differently in different contexts:

- ▶ lexical context: /s/ → en, as in *oxen*
- ▶ phonological/orthographic context: /s/ → es, as in *dresses*

Morphological *variants* are known as **allomorphs**

Definitions

We consider **morphemes** to be...

- ▶ basic units of grammar with no internal structure which may be composed together to form words
- ▶ realized as sequences of linguistic symbols (phones and/or letters)

Morphemes may be rendered differently in different contexts:

- ▶ lexical context: /s/ → en, as in *oxen*
- ▶ **phonological/orthographic context: /s/ → es, as in *dresses***

Morphological *variants* are known as **allomorphs**

Examples

Language	Type	Morpheme	Allomorphs
English	<i>stem</i>	/wake/	wake, wak
	<i>suffix</i>	/s/	s, es
Finnish	<i>stem</i>	/katto/ roof	katto, kato
	<i>suffix</i>	/ta/ partitive	a, ä, ta, tä
Turkish	<i>stem</i>	/kanad/ wing	kanad, kanat
	<i>suffix</i>	/dik/ nominalizer	dik, dük, dık, duk tik, tük, tık, tuk diğ, düğ, dığ, duğ tiğ, tüğ, tığ, tuğ

Examples

Language	Type	Morpheme	Allomorphs
English	<i>stem</i>	/wake/	wake, wak
	<i>suffix</i>	/s/	s, es
Finnish	<i>stem</i>	/katto/ roof	katto, kato
	<i>suffix</i>	/ta/ partitive	a, ä, ta, tä
Turkish	<i>stem</i>	/kanad/ wing	kanad, kanat
	<i>suffix</i>	/dik/ nominalizer	dik, dük, dık, duk tik, tük, tık, tuk diğ, düğ, dığ, duğ tiğ, tüğ, tığ, tuğ

Examples

Language	Type	Morpheme	Allomorphs
English	<i>stem</i>	/wake/	wake, wak
	<i>suffix</i>	/s/	s, es
Finnish	<i>stem</i>	/katto/ roof	katto, kato
	<i>suffix</i>	/ta/ partitive	a, ä, ta, tä
Turkish	<i>stem</i>	/kanad/ wing	kanad, kanat
	<i>suffix</i>	/dik/ nominalizer	dik, dük, dık, duk tik, tük, tık, tuk diğ, düğ, dığ, duğ tiğ, tüğ, tığ, tuğ

Examples

Language	Type	Morpheme	Allomorphs
English	<i>stem</i>	/wake/	wake, wak
	<i>suffix</i>	/s/	s, es
Finnish	<i>stem</i>	/katto/ roof	katto, kato
	<i>suffix</i>	/ta/ partitive	a, ä, ta, tä
Turkish	<i>stem</i>	/kanad/ wing	kanad, kanat
	<i>suffix</i>	/dik/ nominalizer	dik, dük, dık, duk tik, tük, tık, tuk diğ, düğ, dığ, duğ tiğ, tüğ, tığ, tuğ

Examples

Language	Type	Morpheme	Allomorphs
English	<i>stem</i>	/wake/	wake, wak
	<i>suffix</i>	/s/	s, es
Finnish	<i>stem</i>	/katto/ roof	katto, kato
	<i>suffix</i>	/ta/ partitive	a, ä, ta, tä
Turkish	<i>stem</i>	/kanad/ wing	kanad, kanat
	<i>suffix</i>	/dik/ nominalizer	dik, dük, dık, duk tik, tük, tık, tuk diğ, düğ, dığ, duğ tiğ, tüğ, tığ, tuğ

Examples

Language	Type	Morpheme	Allomorphs
English	<i>stem</i>	/wake/	wake, wak
	<i>suffix</i>	/s/	s, es
Finnish	<i>stem</i>	/katto/ roof	katto, kato
	<i>suffix</i>	/ta/ partitive	a, ä, ta, tä
Turkish	<i>stem</i>	/kanad/ wing	kanad, kanat
	<i>suffix</i>	/dik/ nominalizer	dik, dük, dık, duk tik, tük, tık, tuk diğ, düğ, dığ, duğ tiğ, tüğ, tığ, tuğ

Outline

Introduction

Morphemes and Allomorphs

Examples from Challenge Languages

Procedure Overview

Rewrite Rules

Stage A :: Basic EM

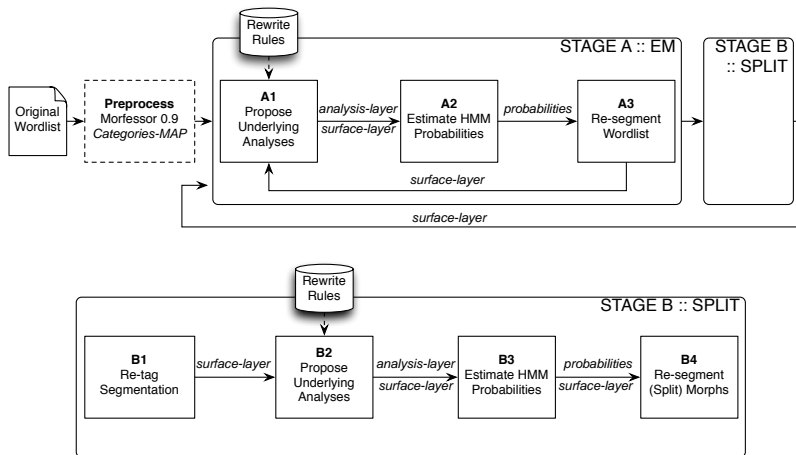
Stage B :: Split Segments

Results

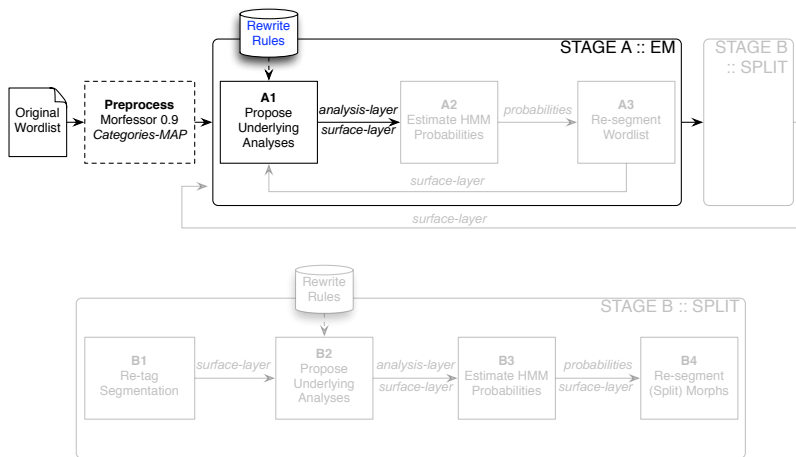
F-Measure Results

Summary

Flowchart



Flowchart



Analysis by Rewrite Rules

- ▶ Written as cascaded (ordered) rewrite rules and compiled into regular expressions.
- ▶ Rules are meant to be run in the analysis direction on a surface segmentation
- ▶ For efficiency, we only permit two types of analyses per segment s :
 - ▶ analyses where all the rules that could have applied, did. (u'')
 - ▶ analyses where no rules applied ($u' = s$)
- ▶ Example Rule capturing the fact that English suffix /s/ is written as es after sibilants (s, z, sh, ...):

$$\underset{\text{underlying}}{\emptyset} \rightarrow \underset{\text{surface}}{e} / [+SIB] + _s \quad (1)$$

Analysis by Rewrite Rules

- ▶ Written as cascaded (ordered) rewrite rules and compiled into regular expressions.
- ▶ Rules are meant to be run in the analysis direction on a surface segmentation
- ▶ For efficiency, we only permit two types of analyses per segment s :
 - ▶ analyses where all the rules that could have applied, did. (u'')
 - ▶ analyses where no rules applied ($u' = s$)
- ▶ Example Rule capturing the fact that English suffix /s/ is written as es after sibilants (s, z, sh, ...):

$$\underset{\text{underlying}}{\emptyset} \rightarrow \underset{\text{surface}}{e} / [+SIB] + _s \quad (1)$$

Analysis by Rewrite Rules

- ▶ Written as cascaded (ordered) rewrite rules and compiled into regular expressions.
- ▶ Rules are meant to be run in the analysis direction on a surface segmentation
- ▶ For efficiency, we only permit two types of analyses per segment s :
 - ▶ analyses where all the rules that could have applied, did. (u'')
 - ▶ analyses where no rules applied ($u' = s$)
- ▶ Example Rule capturing the fact that English suffix /s/ is written as es after sibilants (s, z, sh, ...):

$$\underset{\text{underlying}}{\emptyset} \rightarrow \underset{\text{surface}}{e} / [+SIB] + _s \quad (1)$$

Analysis by Rewrite Rules

- ▶ Written as cascaded (ordered) rewrite rules and compiled into regular expressions.
- ▶ Rules are meant to be run in the analysis direction on a surface segmentation
- ▶ For efficiency, we only permit two types of analyses per segment s :
 - ▶ analyses where all the rules that could have applied, did. (u'')
 - ▶ analyses where no rules applied ($u' = s$)
- ▶ Example Rule capturing the fact that English suffix $/s/$ is written as es after sibilants (s, z, sh, \dots):

$$\underset{\text{underlying}}{\emptyset} \rightarrow \underset{\text{surface}}{e} / [+SIB] + _s \quad (1)$$

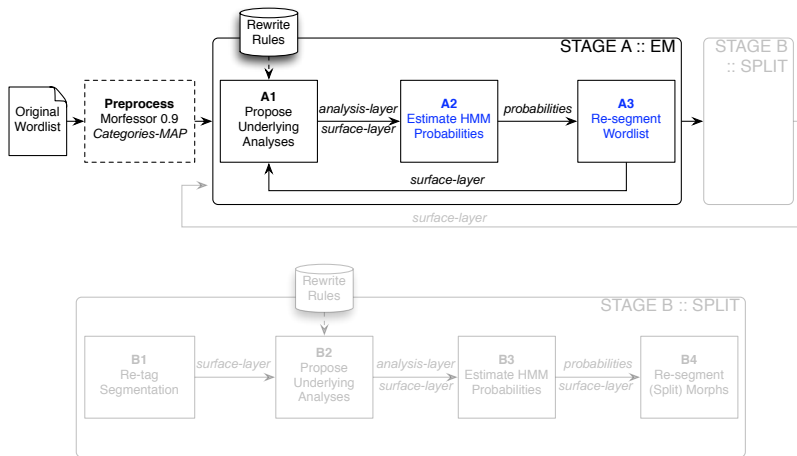
Analysis by Rewrite Rules

- ▶ Written as cascaded (ordered) rewrite rules and compiled into regular expressions.
- ▶ Rules are meant to be run in the analysis direction on a surface segmentation
- ▶ For efficiency, we only permit two types of analyses per segment s :
 - ▶ analyses where all the rules that could have applied, did. (u'')
 - ▶ analyses where no rules applied ($u' = s$)
- ▶ Example Rule capturing the fact that English suffix /s/ is written as es after sibilants (s, z, sh, ...):

$$\underset{\text{underlying}}{\emptyset} \rightarrow \underset{\text{surface}}{e} / [+SIB] + _s \quad (1)$$

Stage A :: Basic EM

Flowchart



Stage A :: Basic EM

- ▶ We **estimate** transition and emission probabilities of a morfeessor-style HMM via maximum likelihood.
- ▶ Emission probabilities are estimated by observing cooccurrences of segments s_i in the surface layer, u_i in the analysis layer, with tags t_i to estimate the probability $P(u_i|t_i)$ of emitting **underlying** morphemes:

$$P(u_i|t_i) = \sum_{s \in \text{allom.-of}(u_i)} P(u_i, s|t_i) \quad (2)$$

Where:

$$u_i = \begin{cases} u_i' & \text{if } u_i = s_i \\ u_i'' & \text{otherwise} \end{cases}$$

Stage A :: Basic EM

- ▶ We **estimate** transition and emission probabilities of a morfeessor-style HMM via maximum likelihood.
- ▶ Emission probabilities are estimated by observing cooccurrences of segments s_i in the surface layer, u_i in the analysis layer, with tags t_i to estimate the probability $P(u_i|t_i)$ of emitting **underlying** morphemes:

$$P(u_i|t_i) = \sum_{s \in \text{allom.-of}(u_i)} P(u_i, s|t_i) \quad (2)$$

Where:

$$u_i = \begin{cases} u'_i & \text{if } u_i = s_i \\ u''_i & \text{otherwise} \end{cases}$$

Stage A :: Basic EM

- ▶ We **estimate** transition and emission probabilities of a morfeessor-style HMM via maximum likelihood.
- ▶ Emission probabilities are estimated by observing cooccurrences of segments s_i in the surface layer, u_i in the analysis layer, with tags t_i to estimate the probability $P(u_i|t_i)$ of emitting **underlying** morphemes:

$$P(u_i|t_i) = \sum_{s \in \text{allom.-of}(u_i)} P(u_i, s|t_i) \quad (2)$$

Where:

$$u_i = \begin{cases} u'_i & \text{if } u_i = s_i \\ u''_i & \text{otherwise} \end{cases}$$

Stage A :: Basic EM

- ▶ We **estimate** transition and emission probabilities of a morfeessor-style HMM via maximum likelihood.
- ▶ Emission probabilities are estimated by observing cooccurrences of segments s_i in the surface layer, u_i in the analysis layer, with tags t_i to estimate the probability $P(u_i|t_i)$ of emitting **underlying** morphemes:

$$P(u_i|t_i) = \sum_{s \in \text{allom.-of}(u_i)} P(u_i, s|t_i) \quad (2)$$

Where:

$$u_i = \begin{cases} u_i' & \text{if } u_i = s_i \\ u_i'' & \text{otherwise} \end{cases}$$

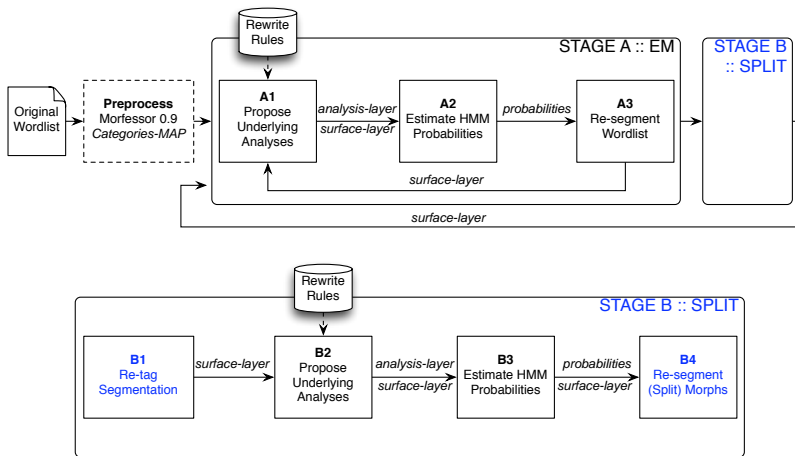
Stage A :: Basic EM

- ▶ Find the **maximum** probability segmentation of the wordlist by finding the argmax of the following equation for each word:

$$\operatorname{argmax}_{\mathbf{u}, \mathbf{t}} P(\mathbf{u}|\mathbf{t})P(\mathbf{t}) \approx \operatorname{argmax}_{\mathbf{u}, \mathbf{t}} \left[\prod_{i=1}^n P(u_i|t_i)P(t_i|t_{i-1}) \right] \quad (3)$$

Stage B :: Split Segments

Flowchart



Stage B :: Split Segments

- ▶ **Re-tag** the segmentation first, using Creutz and Lagus's 2004-2005 heuristic technique, such that only morphs exhibiting prototypical **affix-** or **stem-**distributional features are tagged as such.
- ▶ The remainder are tagged as **noise**; this makes them unavailable to be used in splitting.
- ▶ Key: Forcably **split** segments that are too frequent break under normal circumstances.



Stage B :: Split Segments

- ▶ **Re-tag** the segmentation first, using Creutz and Lagus's 2004-2005 heuristic technique, such that only morphs exhibiting prototypical **affix-** or **stem-**distributional features are tagged as such.
- ▶ The remainder are tagged as **noise**; this makes them unavailable to be used in splitting.
- ▶ Key: Forcably **split** segments that are too frequent break under normal circumstances.

Stage B :: Split Segments

- ▶ **Re-tag** the segmentation first, using Creutz and Lagus's 2004-2005 heuristic technique, such that only morphs exhibiting prototypical **affix-** or **stem-**distributional features are tagged as such.
- ▶ The remainder are tagged as **noise**; this makes them unavailable to be used in splitting.
- ▶ Key: Forcably **split** segments that are too frequent break under normal circumstances.

Outline

Introduction

Morphemes and Allomorphs

Examples from Challenge Languages

Procedure Overview

Rewrite Rules

Stage A :: Basic EM

Stage B :: Split Segments

Results

F-Measure Results

Summary

F-Measure Results

Language	Method	Precision	Recall	F-Measure
English	Morf.- <i>CatMAP</i>	82.17%	33.08%	47.17%
	Bernhard2	61.63%	60.01%	60.81%
	Tepper2-b300	75.62%	51.72%	61.43%
				1% impr.
Finnish	Morf.- <i>CatMAP</i>	76.83%	27.54%	40.55%
	Bernhard2	59.65%	40.44%	48.20%
	Tepper-b600	62.01%	46.20%	52.95%
				10% impr.
Turkish	Zeman	65.81%	18.79%	29.23%
	Morf.-CatMAP	76.36%	24.50%	37.10%
	Tepper-b100	61.15%	49.22%	54.54%
				47% impr.

F-Measure Results

Language	Method	Precision	Recall	F-Measure
English	Morf.- <i>CatMAP</i>	82.17%	33.08%	47.17%
	Bernhard2	61.63%	60.01%	60.81%
	Tepper2-b300	75.62%	51.72%	61.43% 1% impr.
Finnish	Morf.- <i>CatMAP</i>	76.83%	27.54%	40.55%
	Bernhard2	59.65%	40.44%	48.20%
	Tepper-b600	62.01%	46.20%	52.95% 10% impr.
Turkish	Zeman	65.81%	18.79%	29.23%
	Morf.-CatMAP	76.36%	24.50%	37.10%
	Tepper-b100	61.15%	49.22%	54.54% 47% impr.



Outline

Introduction

Morphemes and Allomorphs

Examples from Challenge Languages

Procedure Overview

Rewrite Rules

Stage A :: Basic EM

Stage B :: Split Segments

Results

F-Measure Results

Summary

Summary

- ▶ Our approach, which utilizes a small amount of knowledge in an otherwise unsupervised framework, is successful at learning underlying morphology.
- ▶ Learning improvements over unsupervised approaches are more dramatic for languages with more allomorphic effects, like Turkish (not surprising).
- ▶ There is hope that with a technique such as ours we can pinpoint generalizations about the most effective rules, which would be useful towards developing features for templates from which to **learn** rules.

Summary

- ▶ Our approach, which utilizes a small amount of knowledge in an otherwise unsupervised framework, is successful at learning underlying morphology.
- ▶ Learning improvements over unsupervised approaches are more dramatic for languages with more allomorphic effects, like Turkish (not surprising).
- ▶ There is hope that with a technique such as ours we can pinpoint generalizations about the most effective rules, which would be useful towards developing features for templates from which to **learn** rules.

Summary

- ▶ Our approach, which utilizes a small amount of knowledge in an otherwise unsupervised framework, is successful at learning underlying morphology.
- ▶ Learning improvements over unsupervised approaches are more dramatic for languages with more allomorphic effects, like Turkish (not surprising).
- ▶ There is hope that with a technique such as ours we can pinpoint generalizations about the most effective rules, which would be useful towards developing features for templates from which to **learn** rules.

Thank you!

Acknowledgements

Funding

- ▶ UW Simpson Center for the Humanities
- ▶ UW Graduate School

Thesis Committee

- ▶ Dr. Fei Xia
- ▶ Dr. Emily Bender

Friends and Colleagues

- ▶ Tia Ghose
- ▶ Jonathan North Washington

Special Thanks

Morpho Challenge Team

- ▶ Dr. Mikko Kurimo
- ▶ Dr. Mattias Creutz
- ▶ Matti Varjokallio
- ▶ Ville Turunen