Segmentation strategies for inflection class inference

Sacha Beniamine (LLF), Benoît Sagot (Alpage)
Université Paris Diderot

Décembrettes 9, Toulouse, 2015
Concept of Inflection Classes widely used to analyse inflectional systems

- The definition of IC is crucial for many linguistic and psycholinguistic studies, yet they are often taken for granted.
Quantitative typology of inflectional classification

- Concept of Inflection Classes widely used to analyse inflectional systems
  - The definition of IC is crucial for many linguistic and psycholinguistic studies, yet they are often taken for granted.
- No consensus on how to obtain the classification
Quantitative typology of inflectional classification

- Concept of Inflection Classes widely used to analyse inflectional systems
  - The definition of IC is crucial for many linguistic and psycholinguistic studies, yet they are often taken for granted.
- No consensus on how to obtain the classification
- We explore the concept through computational means: Brown and Evans, 2012; Lee and Goldsmith, 2013; Bonami, 2014
  - Formal definitions of the concept
  - Large datasets
  - Reproducible classifications
  - Commensurable across languages
  - Basis for theoretical and typological comparisons
## Inflection classes

Groups of lexemes that inflect alike.

<table>
<thead>
<tr>
<th></th>
<th>INF</th>
<th>PRES.3.SG</th>
<th>PRES.3.PL</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>tenir ‘hold’</td>
<td>təniʁ</td>
<td>tjɛ</td>
<td>tʃɛn</td>
<td>tʃɛn</td>
</tr>
<tr>
<td>finir ‘finish’</td>
<td>finiʁ</td>
<td>fin</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>haïr ‘hate’</td>
<td>aïʁ</td>
<td>e</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>peler ‘peel’</td>
<td>pəle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pɛle</td>
</tr>
<tr>
<td>laver ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>tasser ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
Groups of lexemes that inflect **alike**.

<table>
<thead>
<tr>
<th></th>
<th>INF</th>
<th>PRES.3.SG</th>
<th>PRES.3.PL</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>təniʁ</td>
<td>tjɛ̃</td>
<td>tjɛn</td>
<td>təny</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finiʁ</td>
<td>fini</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>HAÏR ‘hate’</td>
<td>aïʁ</td>
<td>e</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>pəle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pəle</td>
</tr>
<tr>
<td>LAYER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
What is needed to infer IC from paradigmatic data*

1. What form should an IC system take?
2. What Inflectional Realisations should we infer from the data?
3. How do we measure which lexemes inflect alike?
4. How do we find the best classes among all possible ones?
<table>
<thead>
<tr>
<th></th>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What form should an Inflection class (IC) system take?</td>
</tr>
<tr>
<td>2</td>
<td>What generalisations should we infer from the data?</td>
</tr>
<tr>
<td>3</td>
<td>How do we assess which lexemes inflect alike?</td>
</tr>
<tr>
<td>4</td>
<td>How do we find the best classes among all possible ones?</td>
</tr>
<tr>
<td>5</td>
<td>Results and discussion</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Inflection classes: Cohesive or distinctive?

- Insight from Canonical Typology (Corbett, 2009). An ideal inflection class system is a partition of the set of lexemes that is:
Insight from Canonical Typology (Corbett, 2009). An ideal inflection class system is a partition of the set of lexemes that is:

- **Cohesive**: Maximal homogeneity within classes
Inflection classes: Cohesive or distinctive?

- Insight from Canonical Typology (Corbett, 2009). An ideal inflection class system is a partition of the set of lexemes that is:
  - **Cohesive**: Maximal homogeneity within classes
  - **Distinctive**: Maximal heterogeneity between classes
Inflection classes: Cohesive or distinctive?

- Insight from Canonical Typology (Corbett, 2009). An ideal inflection class system is a partition of the set of lexemes that is:
  - **Cohesive**: Maximal homogeneity within classes
  - **Distinctive**: Maximal heterogeneity between classes

- In most languages, each of these criteria leads to different partitions:

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>tənɪk</td>
<td>tjë</td>
<td>tjɛn</td>
<td>təny</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finɪk</td>
<td>fini</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>HAÏR ‘hate’</td>
<td>ais</td>
<td>ɛ</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>pəle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pəle</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
**Inflection classes: Cohesive or distinctive?**

- Insight from Canonical Typology (Corbett, 2009). An ideal inflection class system is a partition of the set of lexemes that is:
  - **Cohesive**: Maximal homogeneity within classes
  - **Distinctive**: Maximal heterogeneity between classes

- In most languages, each of these criteria leads to different partitions:
  - favouring cohesion: *numerous small, similar classes*

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>tənɪʁ</td>
<td>tjɛ</td>
<td>tjɛn</td>
<td>tənỳ</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finiʁ</td>
<td>fini</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>HAİR ‘hate’</td>
<td>ais</td>
<td>ɛ</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>pɛle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pɛle</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
Insight from Canonical Typology (Corbett, 2009). An ideal inflection class system is a partition of the set of lexemes that is:

- **Cohesive**: Maximal homogeneity within classes
- **Distinctive**: Maximal heterogeneity between classes

In most languages, each of these criteria leads to different partitions:

- **Favouring cohesion**: numerous small, similar classes
- **Favouring distinction**: fewer large classes with exceptions

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>təniʁ</td>
<td>tjɛ</td>
<td>tjɛn</td>
<td>təny</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finiʁ</td>
<td>fini</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>HAİR ‘hate’</td>
<td>ais</td>
<td>ɛ</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>pəle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pəle</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
Inflection classes: Macro and microclasses?

- Dressler and Thornton’s terminology (1996):
  - **Micro-classes**
    - Numerous small, similar classes.
  - **Macro-classes**
    - Fewer large classes with exceptions.

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>tənɪʁ</td>
<td>tjɛn</td>
<td>təny</td>
<td></td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finiʁ</td>
<td>fini</td>
<td>fini</td>
<td></td>
</tr>
<tr>
<td>HAİİR ‘hate’</td>
<td>aiʁ</td>
<td>ɛ</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>pəle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pəle</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
Inflection classes: Macro and microclasses?

- Dressler and Thornton’s terminology (1996):
  - **Micro-classes**
    - Numerous small, similar classes.
  - **Macro-classes**
    - Fewer large classes with exceptions.
  - Combined in a hierarchy. (Corbett and Fraser, 1993; Dressler and Thornton, 1996; Brown and Evans, 2012)

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>tənɪr</td>
<td>tjɛ</td>
<td>tjɛn</td>
<td>təny</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finɪr</td>
<td>fini</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>HAİR ‘hate’</td>
<td>aiɾ</td>
<td>ɛ</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>pəle</td>
<td>pɛl</td>
<td>pɛl</td>
<td>pəle</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
The example of French verbal inflection

- School grammar (Bescherelle):

Class 3: others (~370)

- chapters

- pages

- footnotes
The example of French verbal inflection

- School grammar (Bescherelle)
- Kilani-Schoch and Dressler, 2005: different microclasses, some dropped, two macroclasses (dual route).
Inflection classes: Macro and microclasses?

- **Micro-classes**
  - Homogenous: *Numerous small, similar classes.*
  - Inventories vary across accounts.
  - Empirically motivated

- **Macro-classes**
  - Heterogenous: *Fewer large classes with ”exceptions”.*
  - High variation across accounts.
  - Empirical motivation in question:
Inflection classes: Macro and microclasses?

- **Micro-classes**
  - Homogenous: *Numerous small, similar classes.*
  - Inventories vary across accounts.
  - Empirically motivated

- **Macro-classes**
  - Heterogenous: *Fewer large classes with "exceptions".*
  - High variation across accounts.
  - Empirical motivation in question:

Are macroclasses a descriptive artefact?
Table of Contents

1. What form should an Inflection class (IC) system take?

2. What generalisations should we infer from the data?

3. How do we assess which lexemes inflect alike?

4. How do we find the best classes among all possible ones?

5. Results and discussion

6. Conclusion
Two strategies for the representation of Inflectional Realisations.

- Stem and exponents
  - Captures differences between cells under the assumption of a constant stem.
  - cf. (Blevins, 2006)’s notion of constructive approach.
Two strategies for the representation of Inflectional Realisations.

▶ Stem and exponents
  ▶ Captures differences between cells under the assumption of a constant stem.
  ▶ cf. (Blevins, 2006)’s notion of constructive approach.

▶ Binary alternation patterns
  ▶ Captures the implicative relation between each pair of cells.
  ▶ cf. (Blevins, 2006)’s notion of abstractive approach.
Two strategies for the representation of Inflectional Realisations.

 ► Stem and exponents
  ▶ Captures differences between cells under the assumption of a constant stem.
  ▶ cf. (Blevins, 2006)’s notion of constructive approach.

 ► Binary alternation patterns
  ▶ Captures the implicative relation between each pair of cells.
  ▶ cf. (Blevins, 2006)’s notion of abstractive approach.

 ► Both rely on a segmentation of forms.
  ▶ global segmentation over the whole paradigm.
  ▶ local segmentation over pairs of forms.
### Segmentation strategies

- **Global**: On the basis of a whole paradigm.
- **Local**: On each pair of cells.

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>təniʁ</td>
<td>tjɛ̃</td>
<td>tjen</td>
<td>təny</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>finiʁ</td>
<td>fini</td>
<td>finis</td>
<td>fini</td>
</tr>
<tr>
<td>HAïR ‘hate’</td>
<td>aiʁ</td>
<td>e</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELE ‘peel’</td>
<td>pəle</td>
<td>pel</td>
<td>pəl</td>
<td>pəle</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>lave</td>
<td>lav</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>tase</td>
<td>tas</td>
<td>tas</td>
<td>tase</td>
</tr>
</tbody>
</table>
### Segmentation strategies

- **Global**: On the basis of a whole paradigm.
- **Local**: On each pair of cells.

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF</th>
<th>PRS.3SG</th>
<th>PRS.3PL</th>
<th>PST.PTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>Xəniʁ</td>
<td>Xjē</td>
<td>Xjen</td>
<td>Xəny</td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>Xjə</td>
<td>X</td>
<td>Xs</td>
<td>X</td>
</tr>
<tr>
<td>HAİR ‘hate’</td>
<td>aiʁ</td>
<td>ɛ</td>
<td>ais</td>
<td>ai</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>X₁əX₂e</td>
<td>X₁ɛX₂</td>
<td>X₁ɛX₂</td>
<td>X₁əX₂e</td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>Xe</td>
<td>X</td>
<td>X</td>
<td>Xe</td>
</tr>
<tr>
<td>TASSER ‘press’</td>
<td>Xe</td>
<td>X</td>
<td>X</td>
<td>Xe</td>
</tr>
</tbody>
</table>
**Segmentation strategies**

- **Global**: On the basis of a whole paradigm.
- **Local**: On each pair of cells.

<table>
<thead>
<tr>
<th>Lexeme</th>
<th>INF⇔ PRS.3SG</th>
<th>INF⇔ PRS.3PL</th>
<th>INF⇔ PST.PTCP</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>TENIR ‘hold’</td>
<td>Xανιʁ ⇔ Xjē</td>
<td>Yανιʁ ⇔ Yjɛn</td>
<td>Zίʁ ⇔ Zy</td>
<td></td>
</tr>
<tr>
<td>FINIR ‘finish’</td>
<td>Xɛ ⇔ X</td>
<td>Yɛ ⇔ Ys</td>
<td>Zɛ ⇔ Z</td>
<td></td>
</tr>
<tr>
<td>HAIR ‘hate’</td>
<td>αιʁ ⇔ ε</td>
<td>Yɛ ⇔ Ys</td>
<td>Zɛ ⇔ Z</td>
<td>...</td>
</tr>
<tr>
<td>PELER ‘peel’</td>
<td>X₁ɛX₂e ⇔ X₁ɛX₂</td>
<td>Y₁ɛY₂e ⇔ Y₁ɛY₂</td>
<td>Z ⇔ Z</td>
<td></td>
</tr>
<tr>
<td>LAVER ‘wash’</td>
<td>Xɛ ⇔ X</td>
<td>Ye ⇔ Y</td>
<td>Z ⇔ Z</td>
<td></td>
</tr>
<tr>
<td>Tasser ‘press’</td>
<td>Xɛ ⇔ X</td>
<td>Ye ⇔ Y</td>
<td>Z ⇔ Z</td>
<td></td>
</tr>
</tbody>
</table>
A clustering problem

- In general, grouping elements into classes is a clustering problem.
- There are many well-known solutions in computer science to address such problems.
- All of them require two things:
  - A criterion to evaluate the quality of clusters (classes).
  - An algorithm to explore the search space of all possible groupings.
A clustering problem

- In general, grouping elements into classes is a clustering problem.
- There are many well-known solutions in computer science to address such problems.
- All of them require two things:
  - A criterion to evaluate the quality of clusters (classes).
    → Minimum description length
  - An algorithm to explore the search space of all possible groupings.
A clustering problem

- In general, grouping elements into classes is a clustering problem.
- There are many well-known solutions in computer science to address such problems.
- All of them require two things:
  - A criterion to evaluate the quality of clusters (classes).
    - Minimum description length
  - An algorithm to explore the search space of all possible groupings.
    - Greedy bottom-up algorithm
1. What form should an Inflection class (IC) system take?

2. What generalisations should we infer from the data?

3. How do we assess which lexemes inflect alike?

4. How do we find the best classes among all possible ones?

5. Results and discussion

6. Conclusion
**Description length**

- **Minimum description length** (Rissanen, 1984): Choose the model allowing for the shortest description of the data.
- A partition of the set of lexemes is better than another one if it leads to a more economical description of the system. (Sagot and Walther, 2011; Walther, 2013)

\[
DL(\text{system}) = \text{number of symbols} \times - \sum_{x \in \text{symbols}} P(x) \times \log_2 (P(x))
\]

| Entropy |
We break down the description length into four components:

Toy imaginary dataset with three cells A, B and D.
We break down the description length into four components:

- **m1**:
  - \( A-B = X-X \)
  - \( A-D = Xjo-Xi \)
  - \( B-D = Xjo-Xi \)

- **m2**: (same as m1)

- **m3**: (same as m1)

- **c1**: (same as m1)

- **c2**: (same as m1)

Toy imaginary dataset with three cells A, B and D.
Description length of a partition of the set of lexemes

- We break down the description length into four components:

Toy imaginary dataset with three cells A, B and D.
Description length of a partition of the set of lexemes

► We break down the description length into four components:

Toy imaginary dataset with three cells A, B and D.
We break down the description length into four components:

- **M**: mapping from lexemes to microcl.
- **C**: mapping from microcl. to clusters
- **P**: Lists of patterns in each cluster.

Toy imaginary dataset with three cells A, B and D.
Description length of a partition of the set of lexemes

- We break down the description length into four components:

\[ DL = M + C + P + R \]
Table of Contents

1. What form should an Inflection class (IC) system take?

2. What generalisations should we infer from the data?

3. How do we assess which lexemes inflect alike?

4. How do we find the best classes among all possible ones?

5. Results and discussion

6. Conclusion
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
Clustering algorithm, ex. on European Portuguese conjugation.

(a) Begin with a partition into microclasses.
(b) Merge the pair optimising DL to get a new partition.
(c) Repeat until there is only 1 class.
(d) Run several times, merge variations.
Defining macroclasses

- This allows for an intuitive formal definition of macroclasses
- **Macroclasses**: The partition that best optimises the description length.
  - As we merge clusters, we first expect the DL to decrease.
  - Macroclasses are reached when DL stops decreasing.
- It is an empirical issue whether a system has macroclasses or not.
  
  *We demonstrate their existence in French and European Portuguese conjugation systems.*
Table of Contents

1. What form should an Inflection class (IC) system take?

2. What generalisations should we infer from the data?

3. How do we assess which lexemes inflect alike?

4. How do we find the best classes among all possible ones?

5. Results and discussion

6. Conclusion
Datasets

- Paradigm tables contain phonemically transcribed forms.
- **French**: Flexique (Bonami, Caron, and Plancq, 2014) (5406 verbal entries).
- Comparing **local and global** segmentation strategies
Portuguese classification, global patterns

- **Global strategy (stem & exponents):** Produces scattered classes with no relationship to conventional knowledge of Portuguese verbal IC.
Local strategy (alternation patterns): finds generalisations that display interesting relationship with traditional accounts.
- Local strategy (alternation patterns): finds generalisations that display interesting relationship with traditional accounts.
Global strategy (stem & exponents): Produces scattered classes with no relationship to conventional knowledge of French verbal IC.
Local strategy (alternation patterns): finds generalisations that display interesting relationship with traditional accounts.
Local strategy (alternation patterns): finds generalisations that display interesting relationship with traditional accounts.
We do find macroclasses
Discussion

- We do find macroclasses
  - Not a bipartition (regular / irregular or productive/unproductive), contra Kilani-Schoch and Dressler, 2005
Discussion

- We do find macroclasses
  - Not a bipartition (regular / irregular or productive/unproductive), contra Kilani-Schoch and Dressler, 2005
  - The algorithm had no knowledge of previous accounts.
We do find macroclasses

- Not a bipartition (regular / irregular or productive/unproductive), contra Kilani-Schoch and Dressler, 2005
  - The algorithm had no knowledge of previous accounts.

- We find groupings that were overlooked:
We do find macroclasses
  ▶ Not a bipartition (regular / irregular or productive/unproductive), contra Kilani-Schoch and Dressler, 2005
  ▶ The algorithm had no knowledge of previous accounts.

We find groupings that were overlooked:
  ▶ French: -yer, -oir
We do find macroclasses
  - Not a bipartition (regular / irregular or productive/unproductive), contra Kilani-Schoch and Dressler, 2005
  - The algorithm had no knowledge of previous accounts.

We find groupings that were overlooked:
  - French: -yer, -oir
  - French: haïr, finir, -ure, uire
We do find macroclasses

- Not a bipartition (regular / irregular or productive/unproductive), contra Kilani-Schoch and Dressler, 2005
- The algorithm had no knowledge of previous accounts.

We find groupings that were overlooked:

- French: -yer, -oir
- French: haïr, finir, -ure, uire
- Portuguese: two “irregular” groups.
## Comparison to Other Works

<table>
<thead>
<tr>
<th></th>
<th>Generalisations</th>
<th>Criterion</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown and Evans, 2012</td>
<td>raw paradigms</td>
<td>Compression distance</td>
<td>CompLearn</td>
</tr>
<tr>
<td>Bonami, 2014</td>
<td></td>
<td></td>
<td>UPGMA</td>
</tr>
<tr>
<td>Bonami, 2014</td>
<td>Affixes</td>
<td>Edit distance</td>
<td>UPGMA</td>
</tr>
<tr>
<td>Lee and Goldsmith, 2013</td>
<td>Patterns</td>
<td>Hamming distance</td>
<td>greedy bottom-up</td>
</tr>
<tr>
<td>This work</td>
<td>Sets of characters</td>
<td></td>
<td>greedy bottom-up</td>
</tr>
<tr>
<td>This work</td>
<td>Local patterns</td>
<td></td>
<td>greedy bottom-up</td>
</tr>
<tr>
<td></td>
<td>Global patterns</td>
<td></td>
<td>greedy bottom-up</td>
</tr>
</tbody>
</table>

Features of our approach:

- Principled notion of Inflectional Realization.
- Using a measure that evaluates the quality of the system allows us to infer macroscopic generalisations.
- No parameters to adjust: Occam’s razor is the only criterion.
Table of Contents

1. What form should an Inflection class (IC) system take?

2. What generalisations should we infer from the data?

3. How do we assess which lexemes inflect alike?

4. How do we find the best classes among all possible ones?

5. Results and discussion

6. Conclusion
CONCLUSION

- **Main properties:**
  - Based on information-theoretic measures.
  - Relies on automatically inferred generalisations.
  - Aims at cross-linguistic applications.
  - Formal definition of macroclasses and microclasses.

- An analysis into macroclasses can be empirically motivated.
- **Local segmentation** better captures the structure in inflection systems than global segmentation.
  - Supports the relevance of local patterns of alternation in abstractive approaches (Blevins, 2006).
  - Complementary to work on information-theoretic modelling of implicative structure (Ackerman, Blevins, and Malouf, 2009; Ackerman and Malouf, 2013; Bonami and Beniamine, 2015)
Code available on my webpage:
http://www.llf.cnrs.fr/fr/Gens/Beniamine
Acknowledgments

References


Segmentation strategies

Both can be used in an abstractive approach:

\[ \text{Xo} \Leftrightarrow \text{Xa} \]

\[ /\text{gordo}/ \text{ M.SG} \Leftrightarrow /\text{gorda}/ \text{ F.SG} \]

\[ \text{Xo} \Leftrightarrow \text{Xos} \]

\[ /\text{gordos}/ \text{ M.PL} \Leftrightarrow /\text{gordas}/ \text{ F.PL} \]

\[ \text{Xas} \Leftrightarrow \text{Xos} \]

Global segmentation

Spanish adjective GORDO ‘fat’.
Segmentation strategies

Both can be used in an abstractive approach:

\[ X_o \rightleftarrows X_s \]

\[ X_o \rightleftarrows X_s \]

\[ /\text{gordo/ M.SG} \]

\[ /\text{gorda/ F.SG} \]

\[ /\text{gordos/ M.PL} \]

\[ /\text{gordas/ F.PL} \]

\[ X_o \rightleftarrows X_a \]

\[ X_o \rightleftarrows X_a \]

\[ X_a \rightleftarrows X_o \]

\[ X_a \rightleftarrows X_o \]

\[ X \rightleftarrows X_s \]

\[ X \rightleftarrows X_s \]

\[ X \rightleftarrows X_s \]

\[ X \rightleftarrows X_s \]

\[ X_o \rightleftarrows X_o \]

\[ X_o \rightleftarrows X_o \]

Global segmentation vs local segmentation

Spanish adjective GORDO ‘fat’.
Non Determinism
Local strategy (alternation patterns): finds generalisations that are in line with traditional accounts.
Local strategy (alternation patterns): finds generalisations that are in line with traditional accounts.