Distributional Learning
Theoretical Linguistics and Formal Learning Theory

Alexander Clark
Manchester

6 November 2013
Outline

Structuralist linguistics

Distributional Learning

Substitutable languages

Strong Learning

Lattice based approaches
Outline

Structuralist linguistics

Distributional Learning

Substitutable languages

Strong Learning

Lattice based approaches
American structuralism

Structuralist tradition

- Bloomfield
- Rulon Wells
- Zellig Harris
Reviewing a nonstructuralist book

the value of the book is vitiated, especially for the layman, by a major short-coming. This is the neglect of the method of structural analysis, i.e. of organized synchronic description. As a result, many of the facts about languages are misconstrued, and linguistic theory is distorted. It is the chief purpose of this review to show that an appreciation of linguistic structure is necessary for any interpretation of linguistics, and that its neglect leads to undesirable results in practice.

Rejecting historical approach and its focus on written language.
And its rejection . . .

Chomsky, 1965

The only proposals that are explicit enough to support serious study are those that have been developed within taxonomic linguistics. It seems to have been demonstrated beyond reasonable doubt that quite apart from any questions of feasibility, methods of the sort that have been studied in taxonomic linguistics are intrinsically incapable of yielding the systems of grammatical knowledge that must be attributed to the speaker of a language.
And its rejection . . .

Chomsky, p.c.
From the 50s, there has seemed to me no hope in distributional procedures.
Why was it rejected?

- Distributional learning was perceived as a discovery procedure
- No mathematically precise models (contra Harris)
- Problems of complexity unless the range of variation is finite
- A sequence of phrase markers – unlearnable because you only observe the last one (Katz and Postal, 1964)
- No way of dealing with structure-dependent movement
- No model of ambiguity or of syntactic structure
Discovery procedure
Used by linguists to automatically generate a grammar
But if a grammar is a theory, why do we need to automatically generate it?

Model of language acquisition
Rather than a linguist analyzing a corpus, we have a child processing the primary linguistic data
Kulagina school

Structuralist linguistics died out in America after Chomsky.

Oettinger, 1958

The latter paper (Kulagina 1957) has considerable expository merit, and it is clearer and more sensible than similar papers on set-theoretic concepts in language which have sprouted like ungainly weeds in the lawn of our information-retrieval literature. The work is along somewhat different lines, and of lesser extent but of caliber comparable to that of the excellent theoretical work of Chomsky in this country.

Solomon Marcus, Sestier, Kunze, …
Direct psycholinguistic evidence

Artificial Grammar Learning in infants
Saffran, Aslin, Newport (1996) ...

Human simulation experiments
Gillette, J. et al. (1999), Gleitman (1990), ...

Lexical acquisition experiments
Mintz, T. (2002), Childers and Tomasello (2001), ...
Children and adults do exploit distributional evidence.
Computational experiments

Natural language processing
Standard components of large NLP systems.

CHILDES Experiments
These experiments show that rich evidence is available in reasonably sized natural corpora.
Chomsky (1968/2006)

"The concept of "phrase structure grammar" was explicitly designed to express the richest system that could reasonably be expected to result from the application of Harris-type procedures to a corpus."
Outline

Structuralist linguistics

Distributional Learning

Substitutable languages

Strong Learning

Lattice based approaches
Distributional Learning
Zellig Harris (1949, 1951)

Here as throughout these procedures $X$ and $Y$ are substitutable if for every utterance which includes $X$ we can find (or gain native acceptance for) an utterance which is identical except for having $Y$ in the place of $X$
Example

Is ‘cat’ substitutable for ‘dog’?

- The cat is over there.
- I want a dog for Christmas.
- I want a Siamese cat for Christmas.
- Put a cat-flap in the door to the kitchen.
- An Alsatian is a breed of dog.
- He continues to dog my footsteps.
- I would rather have a dog than a cat as a pet.
Example

Is ’cat’ substitutable for ’dog’?

- The dog is over there.
- I want a cat for Christmas.
- I want a Siamese dog for Christmas.
- Put a dog-flap in the door to the kitchen.
- An Alsatian is a breed of cat.
- He continues to cat my footsteps.
- I would rather have a dog than a dog as a pet.
- I would rather have a cat than a dog as a pet.
Empirical work on Distributional Learning

Real corpora

- Sample is not just of grammatical sentences
- Also semantically well-formed
- Also “true” in some non-technical sense
- Empirical distribution is very complex

Distributional similarity in real corpora often reflects semantic relatedness
Various notions of context

The word ’has’ in the sentence: “If the candidate has an outstanding examination result”

Local syntactic context
Immediately preceding and following word (candidate, an)

Wide bag-of-words context
Skip stop words
Set of words occurring in the same sentence/discourse.
{ candidate, examination, outstanding, result }
Various notions of context

The word 'has' in the sentence: “If the candidate has an outstanding examination result”

Local syntactic context
Immediately preceding and following word
(candidate, an)

Wide bag-of-words context
Skip stop words
Set of words occurring in the same sentence/discourse.
{ candidate, examination, outstanding, result }

Full context
“If the candidate □ an outstanding examination result”
Example

Distribution of “cat” in English

Infinite set of full contexts that ‘cat’ can appear in:
“the □ is over there”
“I want a □ for Christmas”

... 

- We can observe the distribution simply by looking at positive examples.
- We can see a similarity in distribution of “cat” and “dog”.
- Distributional learning is based on this idea.
Context (or environment)
A context is just a pair of strings $(l, r) \in \Sigma^* \times \Sigma^*$, that we write $l \square r$

Context substring relation
$l \square r \odot u = lur$
Special context $\square$

Distribution of a string
Given a language $L$
$C_L(u) = \{ l \square r | lur \in L \}$
$\square \in C_L(u)$ iff $u \in L$
Trees

S
  └── NP
      └── the
  └── VP
      └── is over there

NP
cat
Trees

```
S
  NP
    the
cat

NP
  VP
    is over there
```
I gave a cake to my mother.
I gave a cake to my mother.
I gave a cake to my mother.

λ NP is over there

the cat
Yield of a non-terminal
$Y_G(NP)$ is the set of all strings $w$ such that $NP \Rightarrow^* w$
$Y_G(NP) = \{ \text{the cat, the dog, some blue boxes} \ldots \}$

Contexts of a non-terminal
$C_G(NP)$ is the set of all contexts $l\Box r$ such that $S \Rightarrow^* lNP r$
$C_G(NP) = \{ \Box \text{is over there, I want }\Box, \text{Put it on }\Box \ldots \}$

Any string in $Y(NP)$ can occur in any context in $C(NP)$
A difficult question

Suppose we have some string, say 'the cat', which is in $Y(NP)$

**Question**

What is the relationship between the distribution of 'the cat' $C_L(\text{the cat})$ and the contexts or distribution of NP: $C_G(NP)$?
Outline

Structuralist linguistics

Distributional Learning

Substitutable languages

Strong Learning

Lattice based approaches
Chomsky review of Greenberg, 1959

let us say that two units A and B are substitutable$_1$ if there are expressions X and Y such that XAY and XBY are sentences of L.; substitutable$_2$ if whenever XAY is a sentence of L then so is XBY and whenever XBY is a sentence of L so is XAY (i.e. A and B are completely mutually substitutable). These are the simplest and most basic notions.

Problem:
we need substitutability$_2$ but what we observe is substitutability$_1$
Old idea 2

John Myhill, 1950 commenting on Bar-Hillel

I shall call a system *regular* if the following holds for all expressions $\mu, \nu$ and all wffs $\phi, \psi$ each of which contains an occurrence of $\nu$: If the result of writing $\mu$ for some occurrence of $\nu$ in $\phi$ is a wff, so is the result of writing $\mu$ for any occurrence of $\nu$ in $\psi$. Nearly all formal systems so far constructed are regular; ordinary word-languages are conspicuously not so.

Clark and Eyraud, 2005/2007

A language is *substitutable* if $lur, lvr, l'ur' \in L$ means that $l'vr' \in L$. 
substitutable and reversible

Clark and Eyraud, 2005
A language is substitutable if $lur, lvr, l'ur' \in L$ means that $l'vr' \in L$.

Angluin, 1982
A language is reversible if $ur, vr, ur' \in L$ means that $vr' \in L$. 
A Bad Intuition

One context in common in enough

- The cat died
- The dog died

So “cat” and “dog” are congruent
A Bad Intuition
One context in common in enough

- The cat died
- The dog died

So “cat” and “dog” are congruent
- He is an Englishman
- He is thin

So “thin” and “an Englishman” are congruent.
Chomsky example

John is eager to please
John is easy to please
Polynomial result

The class of substitutable context free languages is polynomially identifiable in the limit from positive data only.

- Polynomial characteristic set
- Polynomial update time

Why the delay?
A Simple Algorithm
Non-technical description

- Given a sample of strings $W = \{w_1, \ldots, w_n\}$.
- Define a graph $G = \langle N, E \rangle$
  - $N$ is the set of all non-empty substrings (factors) of $W$.
  - $E = \{(u, v) | \exists l, r, lur \in W \land lvr \in W\}$.
- Define a grammar in Chomsky normal form
  - The set of non-terminals is the set of components of the graph $G$
  - Have productions for: $[a] \rightarrow a$
  - Add rules for non terminals: $[w] \rightarrow [u][v]$ iff $[w] = [uv]$. 

---

Structuralism  Distribution  Substitutable languages  Strong Learning  Lattice based approaches
Simple linguistically motivated example

the man who is hungry died.
the man ordered dinner.
the man died.
the man is hungry.
is the man hungry?
the man is ordering dinner.
is the man who is hungry ordering dinner?
*is the man who hungry is ordering dinner?
Substitution graph
Auxiliary fronting example
is the man hungry?

is the man hungry?

is the man

is the man

the man who is hungry

ordering dinner

hungry
Counterexample
Berwick, Coen and Niyogi, p.c

is (bob) well ?
is (the man john) well ?
does he think (well) ?
does he think (hitting is nice) ?
is (bob) (well) ?
is the man john hitting is nice ?
Substitution graph

Unlearnable languages
Substitutable languages

• Some very basic languages are not substitutable:
  • $L = \{a, aa\}$
  • $L = \{a^n b^n | n > 0\}$
  • Dyck language

• The very strict requirement for contexts to be disjoint is unrealistic.

• The test for congruence is way too weak.

• If we move to a probabilistic learning approach, or queries we can have a better test

• If the test is good, the hypothesis will never overgenerate
Outline

Structuralist linguistics

Distributional Learning

Substitutable languages

Strong Learning

Lattice based approaches
[Berwick et al.(2011)Berwick, Pietroski, Yankama, and Chomsky]

Put another way, language acquisition is not merely a matter of acquiring a capacity to associate word strings with interpretations. Much less is it a mere process of acquiring a (weak generative) capacity to produce just the valid word strings of a language. Idealizing, one can say that each child acquires a procedure that generates boundlessly many meaningful expressions, and that a single string of words can correspond to more than one expression.
Two options

- Learn from sound/meaning pairs
- Learn trees just from strings
# Alternative model

Two mathematically reasonable models

<table>
<thead>
<tr>
<th></th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak</td>
<td>strings</td>
<td>strings</td>
</tr>
<tr>
<td>Weak semantic</td>
<td>strings + meanings</td>
<td>strings + meanings</td>
</tr>
</tbody>
</table>
Alternative model

Two mathematically reasonable models

<table>
<thead>
<tr>
<th></th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak</td>
<td>strings</td>
<td>strings</td>
</tr>
<tr>
<td>Weak semantic</td>
<td>strings + meanings</td>
<td>strings + meanings</td>
</tr>
</tbody>
</table>

A mathematically unreasonable model

[Wexler and Culicover(1980)]

<table>
<thead>
<tr>
<th></th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong learning</td>
<td>strings</td>
<td>strings + trees</td>
</tr>
</tbody>
</table>
Target class of grammars

$\mathcal{G}$ is some set of context-free grammars. Pick some grammar $G_* \in \mathcal{G}$

Weak learning

We receive examples $w_1, \ldots, w_n, \ldots$
We produce a series of hypotheses $G_1, \ldots, G_n, \ldots$
We want $G_n$ to converge to some grammar $\hat{G}$ such that $L(\hat{G}) = L(G_*)$
Structural descriptions as derivation trees of CFGs

Target class of grammars

$\mathcal{G}$ is some set of context-free grammars.
Pick some grammar $G_* \in \mathcal{G}$

Strong learning

We receive examples $w_1, \ldots, w_n, \ldots$
We produce a series of hypotheses $G_1, \ldots, G_n, \ldots$
We want $G_n$ to converge to some grammar $\hat{G}$ such that $\hat{G} \equiv G_*$
Equivalence

Equality \( \hat{G} = G_\ast \)

Isomorphism \( \hat{G} \equiv G_\ast \): a bijection between the nonterminals so that the two grammars have the same productions.

Strong equivalence Isomorphic trees

Structural equivalence Same unlabelled parse trees

[Paull and Unger(1968)]

Weak equivalence \( L(\hat{G}) = L(G_\ast) \)
Claim
If we have two grammars $G_1, G_2 \in \mathcal{G}$ which are:
1. Weakly equivalent: $L(G_1) = L(G_2)$
2. Not strongly equivalent: $G_1 \not\equiv G_2$
then we can’t strong-learn $\mathcal{G}$. 

Canonical grammars
We can only have one 'distinct' grammar for each language in $\mathcal{G}$.
Contrast with: "a language does not uniquely determine the grammar that generates it" [Lewis(1975)]
Redundancy

Claim
If we have two grammars $G_1, G_2 \in \mathcal{G}$ which are:

1. Weakly equivalent: $L(G_1) = L(G_2)$
2. Not strongly equivalent: $G_1 \not\equiv G_2$

then we can’t strong-learn $\mathcal{G}$.

Canonical grammars
We can only have one ’distinct’ grammar for each language in $\mathcal{G}$.

Contrast with : “a language does not uniquely determine the grammar that generates it”[Lewis(1975)]
Weak to Strong learning

\[ L(\hat{G}_w) = L(G_*) \]

\[ \hat{G}_s \equiv G_* \]
Structuralism  Distribution  Substitutable languages  Strong Learning  Lattice based approaches

```
                NT11
               /   \
              NT0   NT8
             /     \
           NT13   NT15
          /     /               \
         open not hot close
```
<table>
<thead>
<tr>
<th>Condition</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>Implies danger</td>
</tr>
<tr>
<td>Cold</td>
<td>Implies danger</td>
</tr>
<tr>
<td>Hot</td>
<td>Implies danger</td>
</tr>
</tbody>
</table>

**Structuralism**

**Distribution**

**Substitutable languages**

**Strong Learning**

**Lattice based approaches**
Definition

A congruence class $X$ is composite if there are two congruence classes $Y, Z$ such that $X = YZ$.
(and neither $Y$ nor $Z$ is the class $[\lambda]$)

Definition

A congruence class $X$ is prime if it is not composite.

The whole is greater than the sum of the parts
Fundamental theorem of substitutable languages

Every congruence class $Q$ can be uniquely represented as a sequence of primes such that $Q = P_1 \ldots P_n$
Fundamental theorem of substitutable languages

Every congruence class $Q$ can be uniquely represented as a sequence of primes such that $Q = P_1 \ldots P_n$

Intuition

If $X = YZ$, and we have a rule $P \rightarrow QXR$, then we can change it to $P \rightarrow QYZR$
Restriction

- We only consider substitutable languages which have a finite number of primes.
- We define nonterminals only for these primes.

<table>
<thead>
<tr>
<th>Label</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>rain, cold, open rain and cold close</td>
</tr>
<tr>
<td>O</td>
<td>open</td>
</tr>
<tr>
<td>C</td>
<td>close</td>
</tr>
<tr>
<td>B</td>
<td>and, or, ...</td>
</tr>
<tr>
<td>N</td>
<td>not, hot or, cold and ...</td>
</tr>
</tbody>
</table>
We need non-binary rules.

Correct productions

\[ P_0 \rightarrow P_1 \ldots P_k \]
where \( P_0 \supseteq P_1 \ldots P_k \)

- \( N \rightarrow PB \)
- \( P \rightarrow ONSC \)
- Not \( P \rightarrow OPBPC \) – correct but too long.
- Not \( S \rightarrow ONONSCC \)

If there are \( n \) primes then there are at most \( n^2 \) valid productions.
Class of grammars

$G_{sc}$ is the class of canonical grammars for all substitutable languages with a finite number of primes.

Theorem

There is an algorithm which learns $G_{sc}$

- From positive examples
- Identification in the limit convergence
- Strongly
- Using polynomial time and data
Running example

(verbatim output from implementation)
Running example
(verbatim output from implementation)
Example 1

1. the man died
2. the dog died
3. the man is angry
4. it died
5. he died
6. the man kicked the dog
Example 1

```
S
/   \
|    |
|    |
NT4  NT1
/  \\
|    |
|    |
NT6  NT0  died
/    \\
|      \
|      |
the  man

S
/   \
|    |
|    |
NT4  NT1
/  \\
|    |
|    |
NT6  NT0
/    \\
|      \
|      |
the  man  is  angry
```
Example 1

S

NT4

NT6
  the

NT0
  dog

NT3
  kicked

NT4

NT6
  the

NT0
  man
Example 2

1. one
2. two
3. three
4. one and two
5. two times three
6. one and two times three
Example 2

```
S
  |__________________________|
  |                           |
  |                          |
  | S                         |
  |__________________________|
  |                           |
  |                          |
  | NT0                      |
  |__________________________|
  |                           |
  |                          |
  | NT0                      |
  |__________________________|
  |                           |
  | one                      |
  |__________________________|
  |                           |
  | NT0                      |
  |__________________________|
  |                           |
  | one                      |
  |__________________________|
  |                           |
  | NT1                      |
  |__________________________|
  |                           |
  | and                      |
  |__________________________|
  |                           |
  | NT0                      |
  |__________________________|
  |                           |
  | two                      |
```
Example 2

Three trees for this string

\[
S \\
\downarrow \\
NT0 \\
\downarrow \\
\begin{array}{c}
NT0 \\
\downarrow \\
one
\end{array} \quad \begin{array}{c}
NT1 \\
\downarrow \\
\text{and}
\end{array} \quad \begin{array}{c}
NT0 \\
\downarrow \\
\begin{array}{c}
NT0 \\
\downarrow \\
two
\end{array} \quad \begin{array}{c}
NT1 \\
\downarrow \\
times
\end{array} \quad \begin{array}{c}
NT0 \\
\downarrow \\
\text{three}
\end{array}
\end{array}
\]
Example 2
Three trees for this string

```
S
  /\   \   /\   /\  \\
NT0   NT1 NT0
     /\       |
    NT0 NT1 NT0
   /\     /\   |
  one  and two
```

one and two times three
Example 2
Three trees for this string

```
S
   /|
  / \
/    |
NT0   NT1
  /|
 / \
/    |
NT0   NT0   NT1
    /|
   / \
  /    |
one NT1 NT0 NT1
     /|
    / \
   /    |
  and two times
```
Result

Theorem: [Clark(2013b)]
This algorithm can learn all substitutable languages with a finite number of primes:
- Rapidly, efficiently
- Strongly
- From positive data alone
Outline

Structuralist linguistics

Distributional Learning

Substitutable languages

Strong Learning

Lattice based approaches
Limitations of congruential approach

- Not context sensitive
- Classes are too small: need many non-terminals
- Fails to capture generalisations
- No notion of feature
- Hard to tell very similar classes apart
- Lattice based approaches (Clark FG 2009, CoNLL 2010, ICGI 2010)
Example

“cat” in the sentence “There is a cat over there”

Smallest class
Set of all strings that can be substituted for “cat” in all contexts
Possibly only “cat”

Largest class
The set of all strings that can occur in a single context
“There is a □ over there”

• cat
• dog
• large cat
• large cat near here and a small dog
Multiple contexts

Better

The set of all strings that can occur in both contexts

1. “There is a □ over there”
2. “The □ just ran out”
Multiple contexts

Better
The set of all strings that can occur in both contexts

1. “There is a □ over there”
2. “The □ just ran out”

Contexts are often “ambiguous”
“He is □”

• boring
• a lawyer
• away on holiday
Finite context property

Example
I put the book on the table.

A context for the NP
I put the book on □.
Finite context property

Example
I put the book on the table.

A context for the NP
I put the book on □.

- I put the book on the table.
- I put the book on the floor.
- I put the book on the table and then went outside.
Finite context property

Example
I put the book over there.

A context for the NP
I put □ over there.
Finite context property

Example
I put the book over there.

A context for the NP
I put □ over there.

- I put the book over there.
- I put the wine over there.
- I put the book in the bin and then put the wine over there.
Finite context property

Two contexts for the NP
I put □ over there.
AND
I put the book on □.

- I put the table over there.
- I put the book on the table.
Two contexts for the NP
I put □ over there.
AND
I put the book on □.

• I put the book in the bin and then put the wine over there.
• *I put the book on the book in the bin and then put the wine.
Finite context property

Two contexts for the NP
I put □ over there.
AND
I put the book on □.

• *I put the table and then went outside over there.
• I put the book on the table and then went outside.
Notation

Suppose $C$ is a set of contexts. We write

$$C^{\triangleleft}$$

for the set of strings that can occur in all of the contexts of $C$. 
Suppose $C$ is a set of contexts. We write

$$C^\triangleleft$$

for the set of strings that can occur in all of the contexts of $C$.

**Finite Context Property**

A grammar has the FCP if for each category (nonterminal) we can find a finite set of contexts that picks out the strings in that category.

$$C^\triangleleft = \mathcal{L}(G, N)$$
Ambiguity in congruential models

Examples

- Can I have a can of beans?
- May I have a jar of beans?

- “can” and “may” are different distributionally
- “can” and “jar” are different distributionally
- The structural descriptions are thus completely distinct
Ambiguity in lattice models

can, jar, may

jar, can  can, may

jar  can  may
### Concepts

Maximal rectangles

<table>
<thead>
<tr>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
<th>f7</th>
<th>f8</th>
<th>f9</th>
<th>f10</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>k2</td>
<td>k3</td>
<td>k4</td>
<td>k5</td>
<td>k6</td>
<td>k7</td>
<td>k8</td>
<td>k9</td>
<td>k10</td>
</tr>
</tbody>
</table>

- **Structuralism**
- **Distribution**
- **Substitutable languages**
- **Strong Learning**
- **Lattice based approaches**
Concepts

Maximal rectangles

\[
\begin{array}{cccccccc}
 f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 & f_{10} \\
 k_8 & & & & & & & & & \\
k_7 & & & & & & & & & \\
k_6 & & & & & & & & & \\
k_5 & & & & & & & & & \\
k_4 & & & & & & & & & \\
k_3 & & & & & & & & & \\
k_2 & & & & & & & & & \\
k_1 & & & & & & & & & \\
\end{array}
\]

...
### Concepts

**Maximal rectangles**

<table>
<thead>
<tr>
<th></th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
<th>$f_6$</th>
<th>$f_7$</th>
<th>$f_8$</th>
<th>$f_9$</th>
<th>$f_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_4$</td>
<td></td>
<td></td>
<td></td>
<td><strong>1</strong></td>
<td><strong>2</strong></td>
<td><strong>3</strong></td>
<td><strong>4</strong></td>
<td><strong>5</strong></td>
<td><strong>6</strong></td>
<td><strong>7</strong></td>
</tr>
<tr>
<td>$k_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Concepts

Maximal rectangles

\[
\begin{array}{cccccccc}
& f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 & f_{10} \\
\hline
k_8 & & & & \checkmark & & & & & & \\
k_7 & & & & & & & & & & \\
k_6 & & & & & & \checkmark & & & & \\
k_5 & & & & \checkmark & \checkmark & \checkmark & & & & \\
k_4 & & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\
k_3 & & & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\
k_2 & & & & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\
k_1 & & & & & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\
\end{array}
\]
### Concepts

#### Maximal rectangles

<table>
<thead>
<tr>
<th></th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
<th>$f_6$</th>
<th>$f_7$</th>
<th>$f_8$</th>
<th>$f_9$</th>
<th>$f_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_2$</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_3$</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_4$</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_5$</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

- $f_1$ to $f_9$ are the features or attributes.
- $k_1$ to $k_8$ represent different concepts or categories.
## Concepts

### Maximal rectangles

<table>
<thead>
<tr>
<th>k_1</th>
<th>k_2</th>
<th>k_3</th>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
<th>f_4</th>
<th>f_5</th>
<th>f_6</th>
<th>f_7</th>
<th>f_8</th>
<th>f_9</th>
<th>f_10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **k_1**: The smallest k-value.
- **k_2**: The second smallest k-value.
- **k_3**: The largest k-value.

The table represents the maximal rectangles for different k-values, with each row indicating the presence or absence of a rectangle for the corresponding k-value.
### Concepts

#### Maximal rectangles

<table>
<thead>
<tr>
<th></th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
<th>$f_6$</th>
<th>$f_7$</th>
<th>$f_8$</th>
<th>$f_9$</th>
<th>$f_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{10}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{10}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $k_1$ to $k_8$ represent different concepts.
- $f_1$ to $f_{10}$ represent features or attributes.
Partial order
Top and bottom

\[
\begin{array}{cccccccc}
  f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 & f_{10} \\
  k_1 &  &  &  &  &  &  &  &  &  \\
  k_2 &  &  &  &  &  &  &  &  &  \\
  k_3 &  &  &  &  &  &  &  &  &  \\
  k_4 &  &  &  &  &  &  &  &  &  \\
  k_5 &  &  &  &  &  &  &  &  &  \\
  k_6 &  &  &  &  &  &  &  &  &  \\
  k_7 &  &  &  &  &  &  &  &  &  \\
  k_8 &  &  &  &  &  &  &  &  &  \\
\end{array}
\]
### Top and bottom

<table>
<thead>
<tr>
<th></th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
<th>$f_6$</th>
<th>$f_7$</th>
<th>$f_8$</th>
<th>$f_9$</th>
<th>$f_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Complete Lattice

Formal Concept Analysis
It is easy to define a focus-class embracing a large variety of sequence classes but characterized by only a few environments; it is also easy to define one characterized by a great many environments in which all its members occur but on the other hand poor in the number of diverse sequence-classes that it embraces. What is difficult, but far more important than either of the easy tasks, is to define focus-classes rich both in the number of environments characterizing them and at the same time in the diversity of sequence classes that they embrace.

- Concepts high up in the lattice have a few contexts, but lots of strings.
- Concepts low down have a larger number of contexts, but only a few strings.
### Lattice

**Palindrome language over** $a, b$

<table>
<thead>
<tr>
<th></th>
<th>$(a, \lambda)$</th>
<th>$(aa, \lambda)$</th>
<th>$(ba, \lambda)$</th>
<th>$(\lambda, a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\lambda, \lambda)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(ab, \lambda)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(b, \lambda)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(\lambda, b)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$aba$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$bb$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ba$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ab$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$aa$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lambda$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lattice

Many rectangles
Lattice

Palindromic language over $a, b$
Lattice
Linguistic concepts
Non-terminals

On the left hand side

NP  ->  D N
NP  ->  D Adj N
NP  ->  N_proper

On the right hand side

S  ->  NP VP
VP  ->  V_trans NP
Examples

On the left hand side: substrings: $Y(NP)$

```
NP       NP       NP
|        |        |
the     the     the
  cat    red     cat
  john
```

On the right hand side: contexts $C(NP)$

```
S
|       |
NP     VP
|       |
saw    a
  dog

S
|       |
john   VP
|       |
saw    NP
```
The term “context-free” means we can combine any context with any substring. Non-terminals are rectangles.
Minimal grammars

Theorem: [Clark(2013a)]
The smallest grammar that generates a given language will have categories that are *maximal rectangles* in the context substring table.
Context free grammar

Representation
Non-terminals correspond to syntactic concepts.
Context free grammar

Representation
Non-terminals correspond to syntactic concepts.

Rules

\[[X] \rightarrow [Y][Z]\]

If and only if \(X\subset Y\subset Z\)
Bibliography I


D. Lewis.
Languages and language.

Marvin C. Paull and Stephen H. Unger.
Structural equivalence of context-free grammars.
ISSN 0022-0000.
doi: 10.1016/S0022-0000(68)80037-6.

R. S. Wells.
Immediate constituents.
K. Wexler and P. W. Culicover.  
*Formal Principles of Language Acquisition.* 