Linguistic aggregation

The issue

Search techniques for aggregation

An application

WHAT IS AGGREGATION

Term "Aggregation" coined by Mann and Moore (1980)

***Structural aggregation* – Several logical assertions share information**

NL utterance with coordinated or omitted parts

"Foxes and wolves are animals" ***Conjunction reduction***

"Foxes are larger than birds and smaller than wolves" ***Ellipsis / Gapping***

Conceptual aggregation

NL utterance combining the meaning of several ones

"John hits Peter" and "Peter hits John" → "John and Peter fight"

The role of aggregation

Structural aggregation is the dominating factor

Aggregation relevant for virtually every application of NL generation

Not aggregating information is likely to produce awkward and redundant texts

AGGREGATION

Characterization

Relevant for all phases of generation

Integration in the overall process according to demands of application

Methods – in "historic" order

1980:	**(Mann, Moore)**	**Arbitrary rewrite rules covering a variety of issues**
		No control mechanism offered
1990:	**(Dalianis, Hovy)**	**Opportunistically applied rules for coordination issues**
		Discourse motivated ordering
2000:	**(Shaw)**	**Systematic procedure, mostly for syntactic aggregation**
		For applications with substantial aggregation needs
		Linear complexity, sacrificing optimality

WHEN TO APPLY AGGREGATION

Possible stages

Embedded in text planning

A dedicated stage within sentence planning

Part of grammatical processing (Categorial Grammar)

Advantages/disadvantages

Best phase depending on structural size and aggregation options

Early aggregation simplifies processing, but constitutes a commitment

When not to aggregate

Structures reaching across focus space boundaries (different rhetorical state)

Predicates that allow collective reading

“John goes to a party. Liz goes to a party” ≠ “John and Liz go to a party“.

BASICS OF SHAW'S ALGORITHM

Aggregation operations and examples

Propositions that differ in *one* argument

“Alice installed *Excel and Latex* for John on Monday”

Propositions that differ in *multiple* arguments

recurring elements deleted in forward or backward direction

“Cindy removed Access [for John] on Monday,
and Bob [removed] Latex for John on Tuesday”

SHAW'S ALGORITHM

4 stages

1. **Group propositions and order them according to similarities**

   Based on the number of distinct values for each argument.

2. **Determine recurring elements in the ordered propositions that will be combined**

   Done incrementally, starting with the first two propositions, etc.

3. **Create a sentence boundary when the combined clause reaches a given threshold**

4. **Determine which recurring elements are redundant and should be deleted**

   Requires grammatical knowledge about the target language

FUNCTIONALITY AND RESULTS (1)

Initial representation

p(install,Alice,Excel,John,Monday)

p(remove,Bob,Word,John,Tuesday)

p(install,Alice,Latex,John,Monday)

p(remove,Cindy,Access,John,Monday)

FUNCTIONALITY AND RESULTS (2)

Initial representation

p(install,Alice,Excel,John,Monday)
p(remove,Bob,Word,John,Tuesday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)

Representation after ordering (stage 1)

p(install,Alice,Excel,John,Monday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)
p(remove,Bob,Word,John,Tuesday)

FUNCTIONALITY AND RESULTS (3)

Initial representation

p(install,Alice,Excel,John,Monday)
p(remove,Bob,Word,John,Tuesday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)

Representation after ordering (stage 1)

p(install,Alice,Excel,John,Monday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)
p(remove,Bob,Word,John,Tuesday)

Recurrence markings performed (stage 2 & 3)

p(install^{1},Alice^{1},Excel,John^{1,2},Monday^{1,2})
p(install^{1},Alice^{1},Latex,John^{1,2},Monday^{1,2})
p(remove,Cindy,Access,John^{2},Monday^{2})
p(remove,Bob,Word,John,Tuesday)

FUNCTIONALITY AND RESULTS (4)

Initial representation

p(install,Alice,Excel,John,Monday)
p(remove,Bob,Word,John,Tuesday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)

Representation after ordering (stage 1)

p(install,Alice,Excel,John,Monday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)
p(remove,Bob,Word,John,Tuesday)

Recurrence markings performed (stage 2 & 3)

p(install^{1},Alice^{1},Excel,John^{1,2},Monday^{1,2})
p(install^{1},Alice^{1},Latex,John^{1,2},Monday^{1,2})
p(remove,Cindy,Access,John^{2},Monday^{2})
p(remove,Bob,Word,John,Tuesday)

Deletions and coordination done (stage 4)

{p(install,Alice,<Excel,
Latex>, – ,Monday),
p(remove,Cindy,Access,John, –)}
p(remove,Bob,Word,John,Tuesday)

FUNCTIONALITY AND RESULTS (5)

Initial representation

p(install,Alice,Excel,John,Monday)
p(remove,Bob,Word,John,Tuesday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)

Representation after ordering (stage 1)

p(install,Alice,Excel,John,Monday)
p(install,Alice,Latex,John,Monday)
p(remove,Cindy,Access,John,Monday)
p(remove,Bob,Word,John,Tuesday)

Recurrence markings performed (stage 2 & 3)

p(install^{1},Alice^{1},Excel,John^{1,2},Monday^{1,2})
p(install^{1},Alice^{1},Latex,John^{1,2},Monday^{1,2})
p(remove,Cindy,Access,John^{2},Monday^{2})
p(remove,Bob,Word,John,Tuesday)

Deletions and coordination done (stage 4)

{p(install,Alice,<Excel,
Latex>, – ,Monday),
p(remove,Cindy,Access,John, –)}
p(remove,Bob,Word,John,Tuesday)

“On Monday, Alice installed Excel and Latex and Cindy removed Access for John. Bob removed Word for John on Tuesday”

EXTENSIONS TO FORMULAS

Measures meeting particularities of formulas

Ordering criteria based on nesting, number of operators and variables

All combinations of the values of two slots , also for more than 2 slots

Multiple passes with different orderings

Application – Categorization of sets of residue classes and opertions as algebraic structures

Operations of residue classes modulo 5 (the quasi-groups)

***x*-y**	**2*x-y**	**2*x-(y+1)**	**3*x-(y+1)**	**2*x+(y+1)**	**3*x+(y+1)**	**(x-y)+1**	**(2*x)+(3*y)**
	3*x-y	**2*x-(y+2)**	**3*x-(y+2)**	**2*x+(y+2)**	**3*x+(y+2)**	**(x-y)+2**	**(3*x)+(2*y)**
	2*x+y	**2*x-(y+3)**	**3*x-(y+3)**	**2*x+(y+3)**	**3*x+(y+3)**	**(x-y)+3**	**2*(x-y)**
	3*x+y	**2*x-(y+4)**	**3*x-(y+4)**	**2*x+(y+4)**	**3*x+(y+4)**	**(x-y)+4**	**3*(x-y)**

Condensed formal expressions and formulas for the above set of operators

***Coord*(<2,3>,*,*x*,+,<3,2,*,*y*)**	***i**x+(5-*i*)*y,*i*=2,3**
***Coord*(*x*,-,*y*,+,<0,1,2,3,4>)**	***x*-*y*[+*i*], *i*=1,…,4**
***Permut*(<2,3>,*,*x*,<+,->,*y*)**	***i**(*x*±*y*), *i*=2,3**
***Permut*(<2,3>,*,*x*,<+,->,*y*,+,<0,1,2,3,4>)**	***i**x±y[+*j*], *i*=2,3, *j*=1…4**

APPLYING AGGREGATION TECHNIQUES

The task

Presentations in intelligent tutoring systems (Di Eugenio et al.)

Troubleshooting problems (home heating and circuitry)

The method

Syntactic aggregation (DIAG-NLP1)

Functional aggregation (DIAG-NLP2)

Achievement

Much better presentations

Significantly improved learning effects

CONSTRASTIVE TEXTS (1) DIAG-ORIG

The visual combustion check is igniting which is abnormal
(normal is combusting)
Oil Nozzle always
produces this abnormality when it fails.
Oil Supply Valve always
produces this abnormality when it fails.
Oil pump always
produces this abnormality when it fails.
Oil Filter always
produces this abnormality when it fails.
System Control Module sometimes
produces this abnormality when it fails.
Ignitor Assembly never
produces this abnormality when it fails.
Burner Motor always
produces this abnormality when it fa ils.

Facts to be communicated simply uttered in sequence

CONSTRASTIVE TEXTS (2) DIAG-1

The visual combustion check is igniting.
This is abnormal.
Normal is combusting.

Within the furnace system,
this is sometimes caused if
the System Control Module has failed.

Within the Oil Burner,
this is never caused if
the Ignitor Assembly has failed.
In contrast, this is always caused if
the Burner Motor, Oil Filter, Oil Punp,
Oil Supply Valve, or Oil Nozzel has failed.

Structural aggregation **(grouped into system modules) and** ***format improvements***

Some ***referring expressions*** **and a few** ***rhetorical relations***

CONSTRASTIVE TEXTS (3) DIAG-2

The combustion is abnormal.

In the Oil Burner, check the units along te path of the oil and the burner motor.

Empirically grounded content selection and presentation

Based on tutoring interactions between students and a human tutor

Human tutor has available the DIAG facts in tabular form

Tutor utterances coded for

- ***Domain ontology* – object classes and their states**
- ***Tutoring actions* – judgment, problem solving, domain knowledge**
- ***Aggregation* – functional (e.g., oil burner) and linguistic ones**
- ***Relation to DIAG's output* – included, excluded, or contradicted**

CONSTRASTIVE TEXTS (4) DIAG-ORIG

Water pump is a very poor suspect.

Some symptoms you have seen conflict with this theory.

Water pump sound was normal.

This normal indication never results when this unit fails.

Visual combustion check was igniting.

This abnormal indication never results when this unit fails.

Burner Motor RMP Gauge was 525.

This normal indication never results when this unit fails

CONTRASTIVE TEXTS (5) DIAG-1

The water pump is a very poor suspect.
Some symptoms you have seen conflict with this theory.

The following indicators never display normally
when this unit fails.
Within the furnace system,
the Burner Motor RMP Gauge is 525.
Within the water pump and safety cutoff valve,
the water pump sound indicator is normal.

The following indicators never display abnormally
when this unit fails.
Within the fire door sight hole,
the visual combustion check indicator is igniting.

Appropriate grouping and ordering **(normal, abnormal)**

Summary indicators **and** ***domain location*** **references added**

CONSTRASTIVE TEXTS (6) DIAG-2

The water pump is a very poor suspect since the water pump sound is ok.

You have seen that the combustion is abnormal.

Check the units along the path of the oil and the electrical devices.

Technicalities involved

Planning module - decides about content to convey according to query type

Only those units are mentioned which would result in the observed symptom

Functional aggregation invoked according to the number of components referred to

"Some …" if at least half, but not all components

Used if all components present

otherwise no functional aggregation

EVALUATION

Usability

General assessment – DIAG-1 and DIAG-2 preferred to DIAG-orig
(but no preference among DIAG-1 and DIAG-2)

Performance

DIAG-2 produces significantly better scores on posttests than DIAG-orig
(DIAG-1 does not)