GETARUNS
A System for Question-Answering and Summarization

Rodolfo Delmonte
Department of Language Sciences
Università Ca' Foscari
Ca' Garzoni-Moro - San Marco 3417 - 30124 VENEZIA
e-mail: delmont@unive.it website:project.cgm.unive.it

Outline

- Introduction
- INPUT TO THE QA MODULE
- RDF AND SEMANTIC WEB
- DISCOURSE MODEL
- ANALYSING QUESTIONS
- QUESTION ANSWERING
- CONCLUSIONS

Introduction

I. Question Answering on WWW is feasible START:manually…

II. Following the Semantic Web Initiative people use triples or ternary expressions

II. RDFs and ternary structures are insufficient to deal with natural language texts

Semantic Web and Inferencing

- For the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning.
- Artificial-intelligence researchers have studied such systems since long before the Web was developed. Knowledge representation, as this technology is often called, is currently in a state comparable to that of hypertext before the advent of the Web:
- it is clearly a good idea, and some very nice demonstrations exist, but it has not yet changed the world.
- It contains the seeds of important applications, but to realize its full potential it must be linked into a single global system.

Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American (May 2001).

Semantic Web and Inferencing

- The expressive power of the system made vast amounts of information available, and search engines (which would have seemed quite impractical a decade ago) now produce remarkably complete indices of a lot of the material out there.
- The challenge of the Semantic Web, therefore, is to provide a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge-representation system to be exported onto the Web.
- Adding logic to the Web—the means to use rules to make inferences, choose courses of action and answer questions—is the task before the Semantic Web community at the moment.

Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American (May 2001).

Semantic Web and Inferencing

- The logic must be powerful enough to describe complex properties of objects but not so powerful that agents can be tricked by being asked to consider a paradox. Fortunately, a large majority of the information we want to express is along the lines of "a hex-head bolt is a type of machine bolt," which is readily written in existing languages with a little extra vocabulary.
- Meaning is expressed by RDF, which encodes it in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence. These triples can be written using XML tags. In RDF, a document makes assertions that particular things (people, Web pages or whatever) have properties (such as "is a sister of," "is the author of") with certain values (another person, another Web page).

Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American (May 2001).

Semantic Web and Inferencing

- **This structure turns out to be a natural way to describe the vast majority of the data processed by machines. Subject and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a Web page. (URLs, Uniform Resource Locators, are the most common type of URI.) The verbs are also identified by URIs, which enables anyone to define a new concept, a new verb, just by defining a URI for it somewhere on the Web.**

Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American (May 2001).

Are RDFs sufficient for NLP?

SEMANTIC WEB AND RDFs

The RDF data model, as specified in[RDFMS], defines

a simple model

for describing interrelationships among resources in terms of named properties and values.

Too many roles...for RDFs

- as attributes of resources they correspond to traditional value-pairs
- RDF properties also represent relationships between resources
- as such the RDF data model can resemble any entity-relationship diagram
- however the RDFDM provides no mechanisms for declaring these properties
- nor does it provide any mechanisms for defining the relationships between these properties and other resources

FOPL, Inference Engine...

- **More succinctly, the RDF Schema mechanism provides a basic *type system* for use in RDF models.**
- **The typing system is specified in terms of the basic RDF data model - as resources and properties.**
- **The schema specification language is a declarative representation language influenced by ideas from knowledge representation (e.g. semantic nets, frames, predicate logic)**
- **The RDF schema specification language is less expressive, but much simpler to implement, than full predicate calculus languages such as CycL[CycL] and KIF [KIF].**

Ternary Expressions

Ternary expressions(T-expressions), <subject relation object>.
Certain other parameters (adjectives, possessive nouns, prepositional phrases, etc.) are used to create additional T-expressions in which prepositions and several special words may serve as relations. For instance, the following simple sentence

(1) Bill surprised Hillary with his answer

will produce two T-expressions:

(2) <<Bill surprise Hillary> with answer>

<answer related-to Bill>

Triples at CL

- The key step in the CL Research question-answering prototype was the analysis of the parse trees to extract semantic relation triples and populate the databases used to answer the question.

- A semantic relation triple consists of a discourse entity, a semantic relation which characterizes the entity's role in the sentence, and a governing word to which the entity stands in the semantic relation.

Predicates in Triples

- The semantic relations in which entities participate are intended to capture the semantic roles of the entities, as generally understood in linguistics.
- This includes such roles as agent, theme, location, manner, modifier, purpose, and time.
- ... surrogate place holders. ...included "SUBJ," "OBJ", "TIME," "NUM," "ADJMOD," and the prepositions heading prepositional phrases.

Grammatical Relations

- The governing word was generally the word in the sentence that the discourse entity stood in relation to.
- For "SUBJ," "OBJ," and "TIME," this was generally the main verb of the sentence. For prepositions, the governing word was generally the noun or verb that the prepositional phrase modified.
- For the adjectives and numbers, the governing word was generally the noun that was modified.

Introduction 2

I. The IR/IE BOWs approach suffers (at least) from Reversible Arguments Problem (Katz & Lin)

- What do frogs eat? vs
What eats frogs?

-The president of Russia visited the president of China. Who visited the president?

-John killed Tom. Tom was killed by a man. Who killed the man?

Problematic structures for BOWs & ternary expressions

Subject vs Object

- **Passivized structures**
- **Inchoatized structures**
- **Ergativized structures**

Control

- **Open Predicative structures (relatives, adjectival adjuncts, infinitives, participials, etc.**

Problematic structures for BOWs & ternary expressions

Factuality Prejudice

- **Negation**
- **Quantification**
- **Opaque contexts**
- **Modality**
- **Conditionals**

Anaphora Resolution

Some Proposals

1. **Large-scale indexing via partial parsing**

2. **Search Engines do IE by keywords**
2.1 **Then use top ten candidates to search answers and generate them from deep analysis**

3. **Apply deep analysis to the web and produce full-fledged knowledge representation of its linguistics contents**

GETARUNS: A HYBRID SYSTEM

SHALLOW COMPUTATION

STATISTICAL & WEAKLY LINGUISTIC

HYBRID COMPUTATION

SYMBOLIC & DEEP LINGUISTIC

DEEP COMPUTATION

GETARUNS: A HYBRID SYSTEM

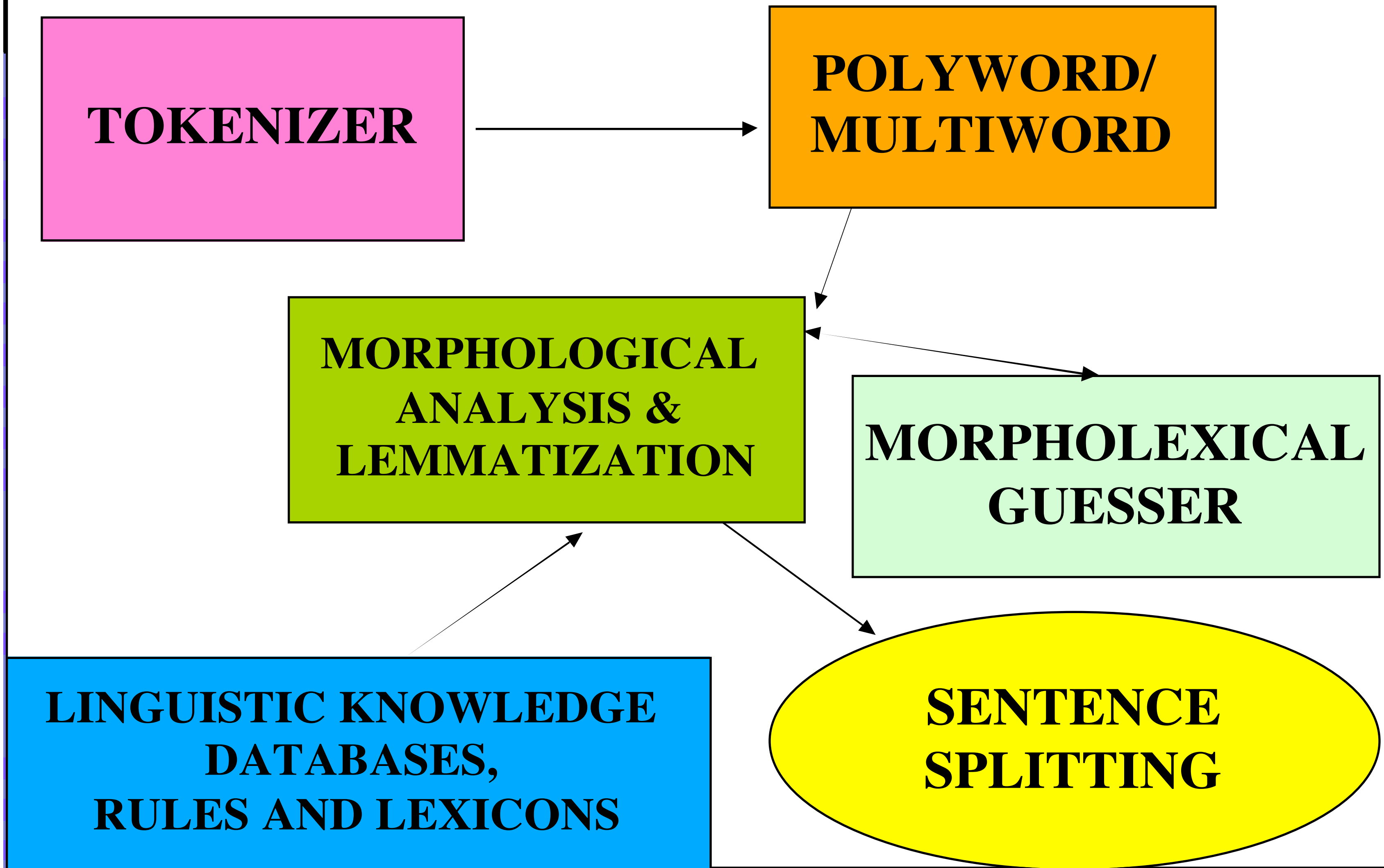
SHALLOW & COMPLETE

- Complete → **Complete Parsing & Semantics Deep Anaphora Resolution**
- Partial
- Shallow

Shallow & Partial Parsing... Semantics... Anaphora Resolution

- Chunks → **Shallow Parsing… No Semantics at Propositional Level… Shallow Anaphora Resolution**

GETARUNS' ARCHITECTURE



PARSING COMPLETE

Semantic Consistency Check
for every
Syntactic Constituent
Starting fromCP level

Deterministic
Policy:
Lookahead
WFSTs

Verb Guidance From
Subcategorization
Frames

Top-Down
DCG-based
Grammar Rules

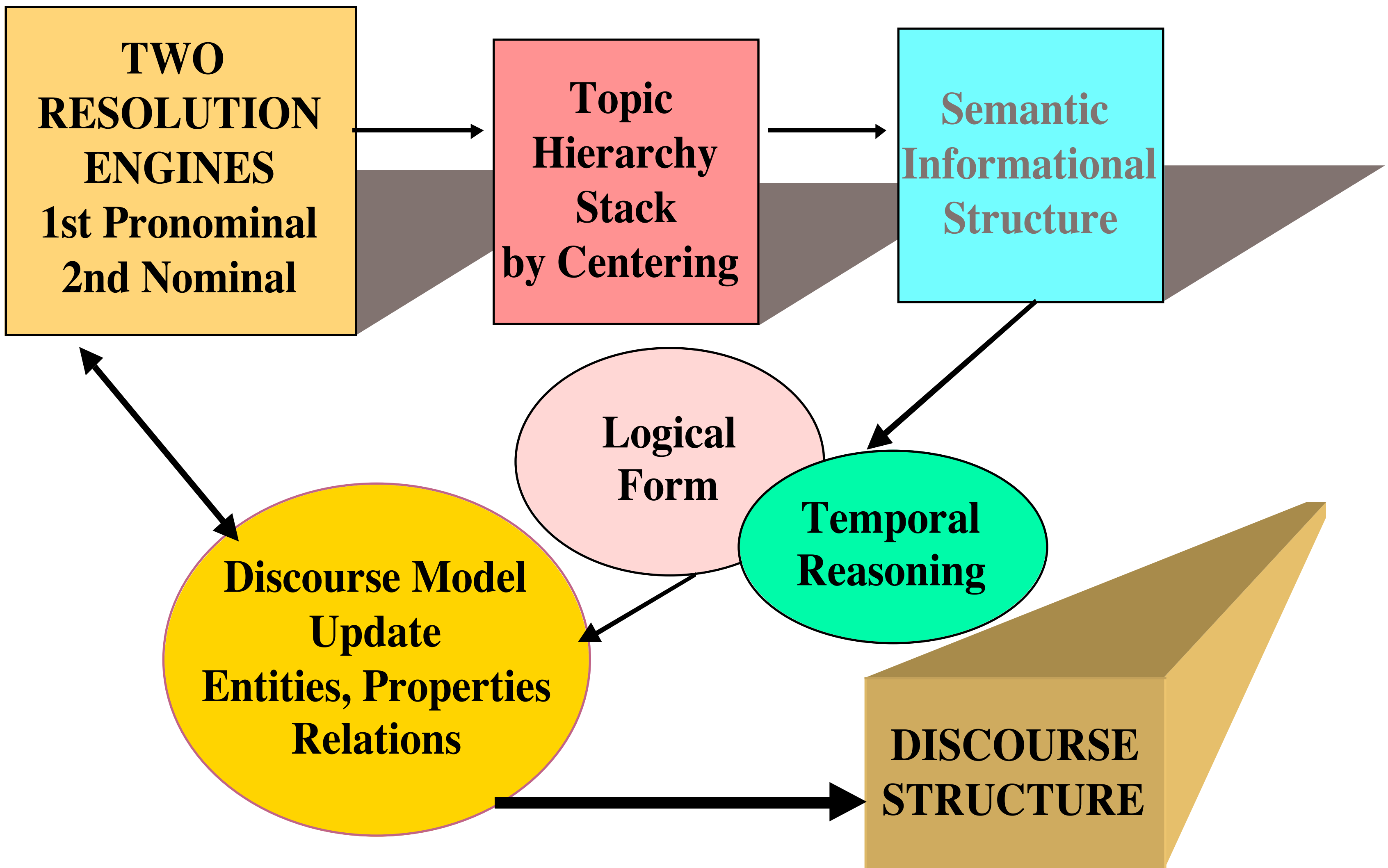
Phrase Structure Rules
==> F-structure
check for Completeness
Coherence, Uniqueness

Tense, Aspect and
Time Reference:
Time Relations and
Reference Interval

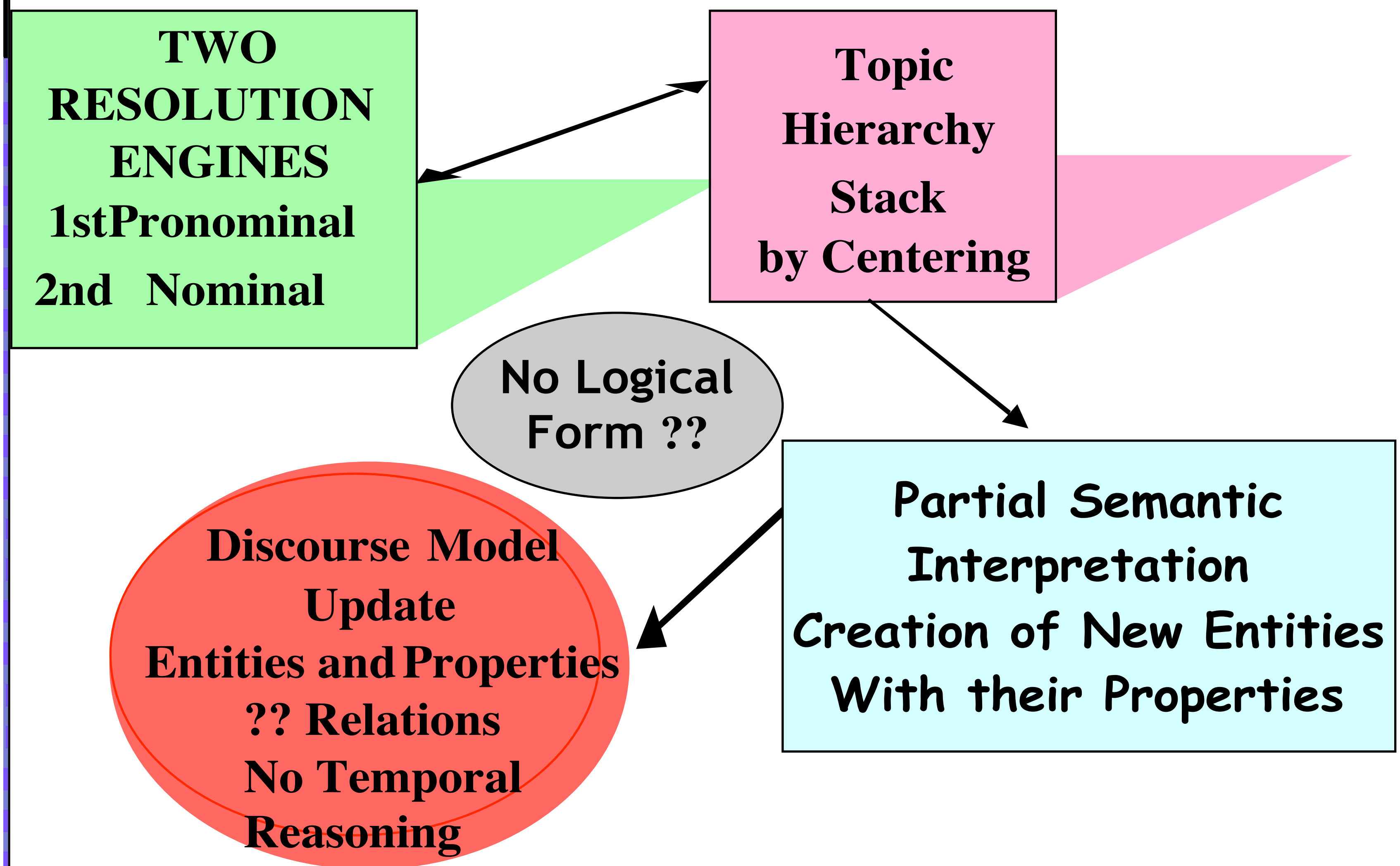
Quantifier Raising

Pronominal Binding at f-structure level

SYSTEM ARCHITECTURE II°



HIGH SYSTEM SHALLOW



Discourse Model

John went into a restaurant

1. <John go restaurant>
2. GO(SUBJ(John), OBL(restaurant))

FACT is an
Infon(Index,
Relation(Property),
List of Arguments - with Semantic Roles,
Polarity - 1 affirmative, 0 negation,
Temporal Location Index,
Spatial Location Index)

Arguments have a Semantic Identifier

Propositional Facts have a Semantic Identifier

Discourse Model

loc(infon2, id1, [arg:main_tloc, arg:tr(f1_r01)])

loc(infon3, id2, [arg:main_sloc, arg:restaurant])

ind(infon4, id3)

fact(infon5, inst_of, [ind:id3, class:man], 1, univ, univ)

fact(infon6, name, [john, id3], 1, univ, univ)

ind(infon7, id4)

fact(infon8, isa, [ind:id4, class:restaurant], 1, id1, id2)

fact(infon9, inst_of, [ind:id4, class:place], 1, univ, univ)

fact(id5, go, [agent:id3, locat:id4], 1, tes(f1_r01), id2)

fact(infon12, isa, [arg:id5, arg:ev], 1, tes(f1_r01), id2)

fact(infon13, isa, [arg:id6, arg:tloc], 1, tes(f1_r01), id2)

fact(infon14, past, [arg:id6], 1, tes(f1_r01), id2)

fact(infon15, time, [arg:id5, arg:id6], 1, tes(f1_r01), id2)

DM: Partial & Chunks

ind(infon4, id3)

fact(infon5, inst_of, [ind:id3, class:man], 1, univ, univ)

fact(infon6, name, [john, id3], 1, univ, univ)

ind(infon7, id4)

fact(infon8, isa, [ind:id4, class:restaurant], 1, id1, id2)

fact(infon9, inst_of, [ind:id4, class:place], 1, univ, univ)

fact(id5, go, [agent:id3, locat:id4], 1, univ, id2)

ind(infon4, id3)

fact(infon5, inst_of, [ind:id3, class:man], 1, univ, univ)

fact(infon6, name, [john, id3], 1, univ, univ)

ind(infon7, id4)

fact(infon8, isa, [ind:id4, class:restaurant], 1, id1, id2)

fact(infon9, inst_of, [ind:id4, class:place], 1, univ, univ)

An Example

Text 1. At the restaurant

John went into a restaurant. There was a table in the corner. The waiter took the order. The atmosphere was warm and friendly. He began to read his book.

from Sanford and Garrod
"Psychological Atmosphere Anaphora Resolution"

```
ds(new(1-1), 1-1, [expected:id3:john], go([id3:john,
id4:restaurant], 1, id2), undef(tes(f1_r01), tes(f1_r01)),
narration, 1-[1])]).
```

Discourse Structures

ds(new(1-1), 1-1, [expected:id3:john], go([id3:john, id4:restaurant], 1, id2), undef(tes(f1_r01), tes(f1_r01)), narration, 1-[1])]).

ds(down(1-1), 1-2, [main:id3:john], there_be([id7:table, id10:corner], 1, id2), during(tes(f2_r02), tes(f1_r01)), explanation, 1-[2]),

ds(level(2-2), 2-3, [expected:id13:waiter, secondary:id3:john], take([id13:waiter, id3:exist], 1, id2), after(tes(f1_r03), tes(f1_r01)), narration, 1-[2,3]),

ds(down(2-3), 2-4, [main:id3:john], be([infon59:[friendly, warm]], 1, id2), during(tes(f1_r04), tes(f1_r01)), explanation, 1-[2,3,4]),

ds(to(1-1), 1-5, [main:id3:john, secondary:id16:atmosphere], begin([id3:john, id20:read], 1, id2), after(tes(f1_r05), tes(f1_r01)), inception, 1-[5]),

Temporal Reasoning

```
fact(infon66, poss, [john, id3, id19], 1, id1, id2)
ind(infon67, id19)
fact(infon68, inst_of, [ind:id19, class:thing], 1, univ, univ)
fact(infon69, isa, [ind:id19, class:book], 1, id1, id2)
fact(id20, read, [agent:id3, actor:id19], 1, tes(finf1_r05), id2)
fact(infon73, isa, [arg:id20, arg:ev], 1, tes(finf1_r05), id2)
fact(infon74, isa, [arg:id21, arg:tloc], 1, tes(finf1_r05), id2)
fact(infon75, pres, [arg:id21], 1, tes(finf1_r05), id2)
fact(infon76, time, [arg:id20, arg:id21], 1, tes(f1_r05), id2)
fact(id22, begin, [actor:id3, prop:id20], 1, tes(f1_r05), id2)
fact(infon77, isa, [arg:id22, arg:ev], 1, tes(f1_r05), id2)
fact(infon78, isa, [arg:id23, arg:tloc], 1, tes(f1_r05), id2)
fact(infon79, past, [arg:id23], 1, tes(f1_r05), id2)
fact(infon80, time, [arg:id22, arg:id23], 1, tes(f1_r05), id2)

includes(tr(f1_r05), id1)
after(tes(f1_r05), tes(f1_r01))
```

Partial Analysis

This unit has been manufactured to assure your personal safety, but improper use can result in potential electrical shock or fire hazards.

refs(r0007, id5, subj, unit, [this, unit])
refs(r0008, id8, mod, electrical_shock, [electrical, shock, electrical_shock])
refs(r0009, id4, obj, safety, [your, personal, safety])
refs(r0010, id6, subj, use, [improper, use])

refs(r0011, id7, obl, fire, [in, potential, electrical_shock, or, fire])
refs(r0012, 4-4, id6, subj, [can, result], [improper, use])
refs(r0013, 4-4, id7, obl, [can, result], [in, potential, electrical_shock, or, fire])
refs(r0016, 4-1, id5, subj, [has, been, manufactured], [this, unit])
refs(r0017, 4-1, id9, vcomp, [has, been, manufactured], [to, assure])
refs(r0015, 4-2, id9, obj, [to,assure], [your,personal,safety])

Factoring out facts

```
entity(ind,id5,4,facts([
fact(infon16, inst_of, [ind:id5, class:institution], 1, univ, univ),
fact(infon17, isa, [ind:id5, class:unit], 1, univ, univ),
fact(infon24, this, [nil:id5], 1, univ, univ),
fact(id11, manufacture, [theme_aff:id5, prop:id9], 1, univ, univ),
```

```
entity(ind,id4,7,facts([
fact(infon12, inst_of, [ind:id4, class:place], 1, univ, univ),
fact(infon13, isa, [ind:id4, class:safety], 1, univ, univ),
fact(infon14, your, [nil:id4], 1, univ, univ),
fact(infon25, personal, [nil:id4], 1, univ, univ),
fact(id9, assure, [actor:arb, theme:id4], 1, univ, univ),
```

A more complex example

How Maple Syrup is Made

Maple syrup comes from sugar maple trees. At one time, maple syrup was used to make sugar. This is why the tree is called a "sugar" maple tree.

Sugar maple trees make sap. Farmers collect the sap. The best time to collect sap is in February and March. The nights must be cold and the days warm.

The farmer drills a few small holes in each tree. He puts a spout in each hole. Then he hangs a bucket on the end of each spout. The bucket has a cover to keep rain and snow out. The sap drips into the bucket. About 10 gallons of sap come from each hole.

A more complex example

How Maple Syrup is Made

Maple syrup comes from sugar maple trees. At one time, maple syrup was used to make sugar. This is why the tree is called a "sugar" maple tree.

Sugar maple trees make sap. Farmers collect the sap. The best time to collect sap is in February and March. The nights must be cold and the days warm.

The farmer drills a few small holes in each tree. He puts a spout in each hole. Then he hangs a bucket on the end of each spout. The bucket has a cover to keep rain and snow out. The sap drips into the bucket. About 10 gallons of sap come from each hole.

4.This is why the tree is called a "sugar" maple tree

```
ent(infon61, id18)
fact(infon62, prop, [arg:id18, disc_set:[id16:use:[theme_aff:id2, result:id14]]], 1, univ, id7)
ind(infon63, id19)
fact(infon64, tree, [nil:id19], 1, univ, id7)
fact(infon65, inst_of, [ind:id19, class:plant_life], 1, univ, univ)
fact(infon66, isa, [ind:id19, class:tree], 1, univ, id7)
in(infon67, id19, id8)
ind(infon68, id20)
fact(infon69, inst_of, [ind:id20, class:thing], 1, univ, univ)
fact(infon70, isa, [ind:id20, class:reason], 1, univ, id7)
fact(infon72, reason, [nil:id18], 1, univ, id7)
fact(id21, be, [prop:infon72], 1, tes(f15_es4), id7)
fact(id22, call, [actor:id19, theme:id10, prop:infon72], 1, tes(f15_es4), id7)
fact(infon73, isa, [arg:id22, arg:st], 1, tes(f15_es4), id7)
fact(infon74, isa, [arg:id23, arg:tloc], 1, tes(f15_es4), id7)
fact(infon75, pres, [arg:id23], 1, tes(f15_es4), id7)
fact(infon82, time, [arg:id22, arg:id23], 1, tes(f5_es4), id7)
during(tes(f15_es4), tes(f2_es3))
includes(tr(f15_es4), univ)
```

7.The best time to collect sap is in February and March

ind(infon110, id32)
fact(infon111, best, [ind:id32], 1, univ, id7)
fact(infon112, inst_of, [ind:id32, class:time], 1, univ, univ)
fact(infon113, isa, [ind:id32, class:time], 1, univ, id7)
set(infon114, id33)
card(infon115, 2)
fact(infon116, inst_of, [ind:id33, class:time], 1, univ, univ)
fact(infon117, isa, [ind:id33, class:[march, February]], 1, univ, id7)
fact(id35, collect, [agent:id28, theme_aff:id24], 1, tes(finf1_es7), id7)
fact(infon118, isa, [arg:id35, arg:ev], 1, tes(finf1_es7), id7)
fact(infon119, isa, [arg:id36, arg:tloc], 1, tes(finf1_es7), id7)
fact(infon120, nil, [arg:id36], 1, tes(finf1_es7), id7)
fact(infon121, [march, February], [arg:id32], 1, univ, id7)
fact(id37, be, [prop:id35, prop:infon130], 1, tes(f1_es7), id7)
fact(infon122, isa, [arg:id37, arg:st], 1, tes(f1_es7), id7)
fact(infon123, isa, [arg:id38, arg:tloc], 1, tes(f1_es7), id7)
fact(infon124, pres, [arg:id38], 1, tes(f1_es7), id7)
fact(infon125, time, [arg:id37, arg:id38], 1, tes(f1_es6), id7)
during(tes(f1_es7), tes(f1_es6))
includes(tr(f1_es7), univ)

1. Who collects maple sap?

q_loc(infon3, id1, [arg:main_tloc, arg:tr(f1_uq_1)])
q_ent(infon4, id2)
q_fact(infon5, isa, [ind:id2, class:who], 1, id1, univ)
q_fact(infon6, inst_of, [ind:id2, class:man], 1, univ, univ)
q_class(infon7, id3)
q_fact(infon8, inst_of, [ind:id3, class:coll], 1, univ, univ)
q_fact(infon9, isa, [ind:id3, class:sap], 1, id1, univ)
q_fact(infon10, focus, [arg:id2], 1, id1, univ)
q_fact(infon11, maple, [ind:id3], 1, id1, univ)
q_fact(id4, collect, [agent:id2, theme_aff:id3], 1, tes(f1_uq_1), univ)
q_fact(infon13, isa, [arg:id4, arg:pr], 1, tes(f1_uq_1), univ)
q_fact(infon14, isa, [arg:id5, arg:tloc], 1, tes(f1_uq_1), univ)
q_fact(infon15, pres, [arg:id5], 1, tes(f1_uq_1), univ)

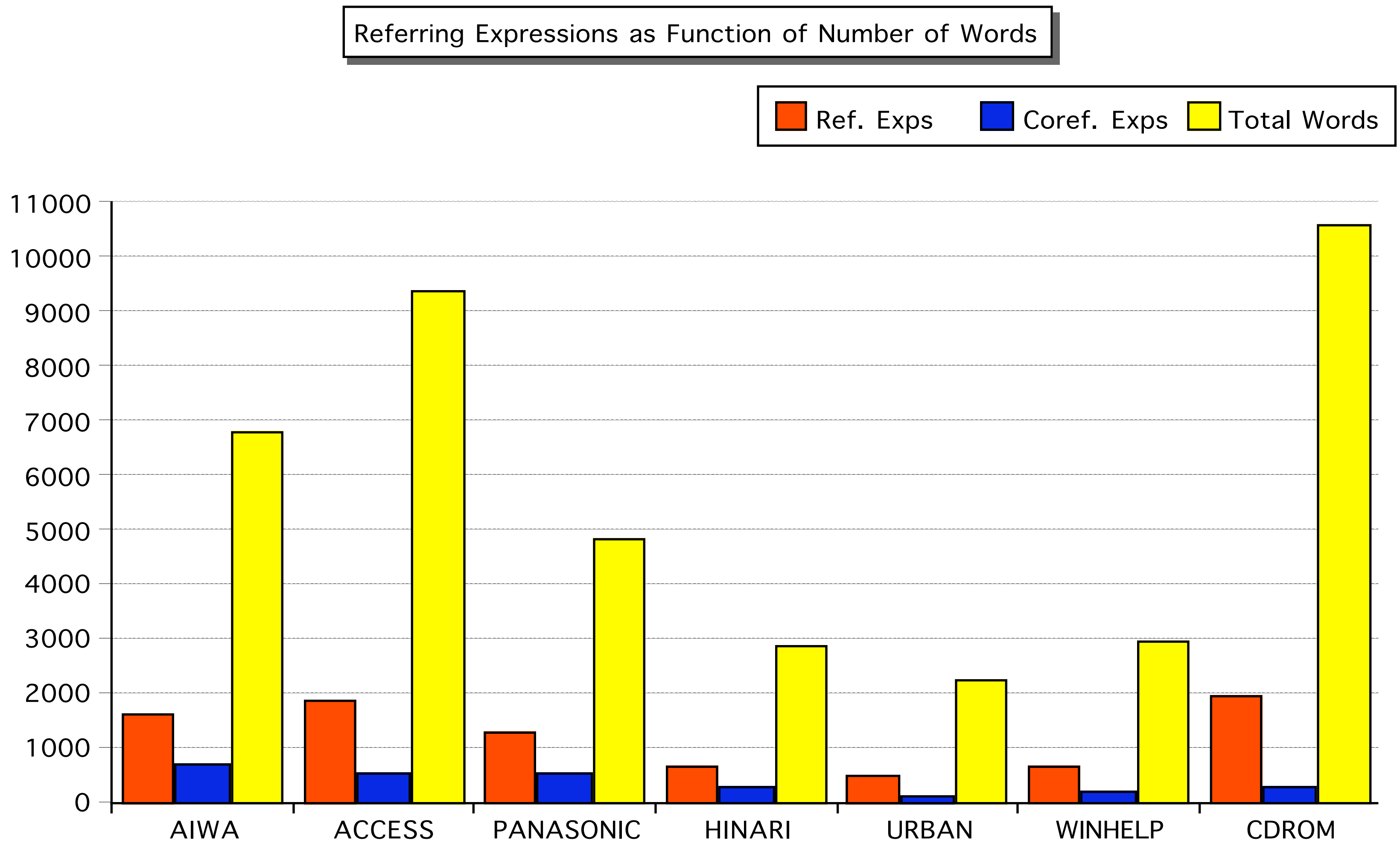
1. Farmers collects maple sap.

udm_loc(infon3, id1, [arg:main_tloc, arg:tr(f1_ua_1)])
udm_ent(infon4, id2)
udm_fact(infon5, isa, [ind:id2, class:farmer], 1, id1, univ)
udm_fact(infon6, inst_of, [ind:id2, class:man], 1, univ, univ)
udm_class(infon7, id3)
udm_fact(infon8, inst_of, [ind:id3, class:coll], 1, univ, univ)
udm_fact(infon9, isa, [ind:id3, class:sap], 1, id1, univ)
udm_fact(infon11, maple, [ind:id3], 1, id1, univ)
udm_fact(id4, collect, [agent:id2, theme_aff:id3], 1, tes(f1_ua_1), univ)
udm_fact(infon13, isa, [arg:id4, arg:pr], 1, tes(f1_ua_1), univ)
udm_fact(infon14, isa, [arg:id5, arg:tloc], 1, tes(f1_ua_1), univ)
udm_fact(infon15, pres, [arg:id5], 1, tes(f1_ua_1), univ)

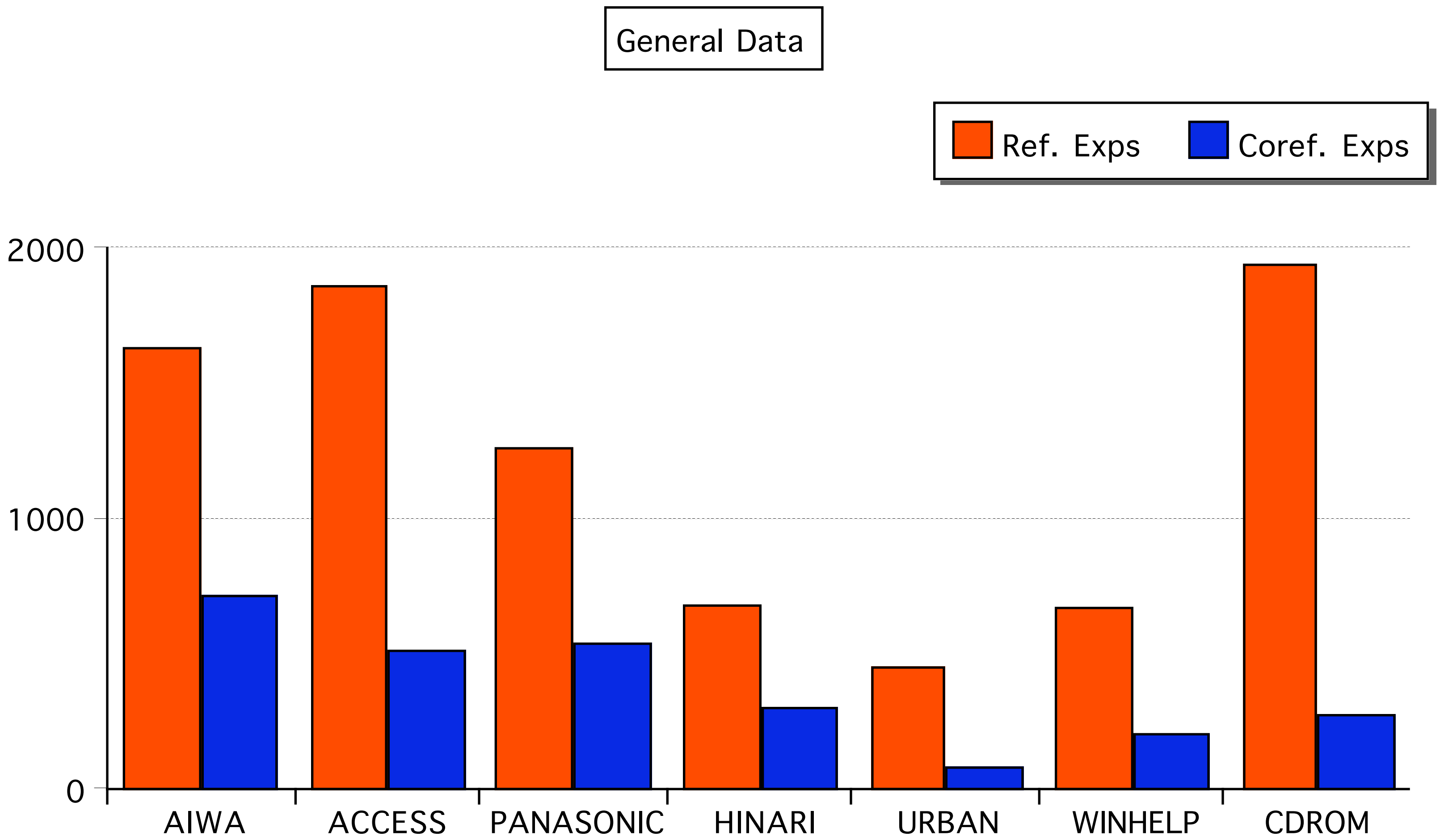
Evaluation: Coreference

We downloaded the only freely available corpus annotated with anaphoric relations, i.e. Wolverhampton's Manual Corpus made available by Prof. Ruslan Mitkov on his website. The corpus contains text from Manuals at the following address, <u>http://clg.wlv.ac.uk/resources/corpus.html</u> which is described by the authors as "Coreferentially annotated corpora", last updated: September 07 2002, made up of seven archives, where each archive contains the original (unannotated) file and the file containing the SGML annotation. The annotators adopted the MUC-7 annotation scheme and they operated by annotating identity-of-reference direct nominal anaphora, "which includes relationships such as specialisation, generalisation and synonymy, but excludes part-of and set membership relations that are considered instances of indirect anaphora."(Mitkov et al, available on the webpage).

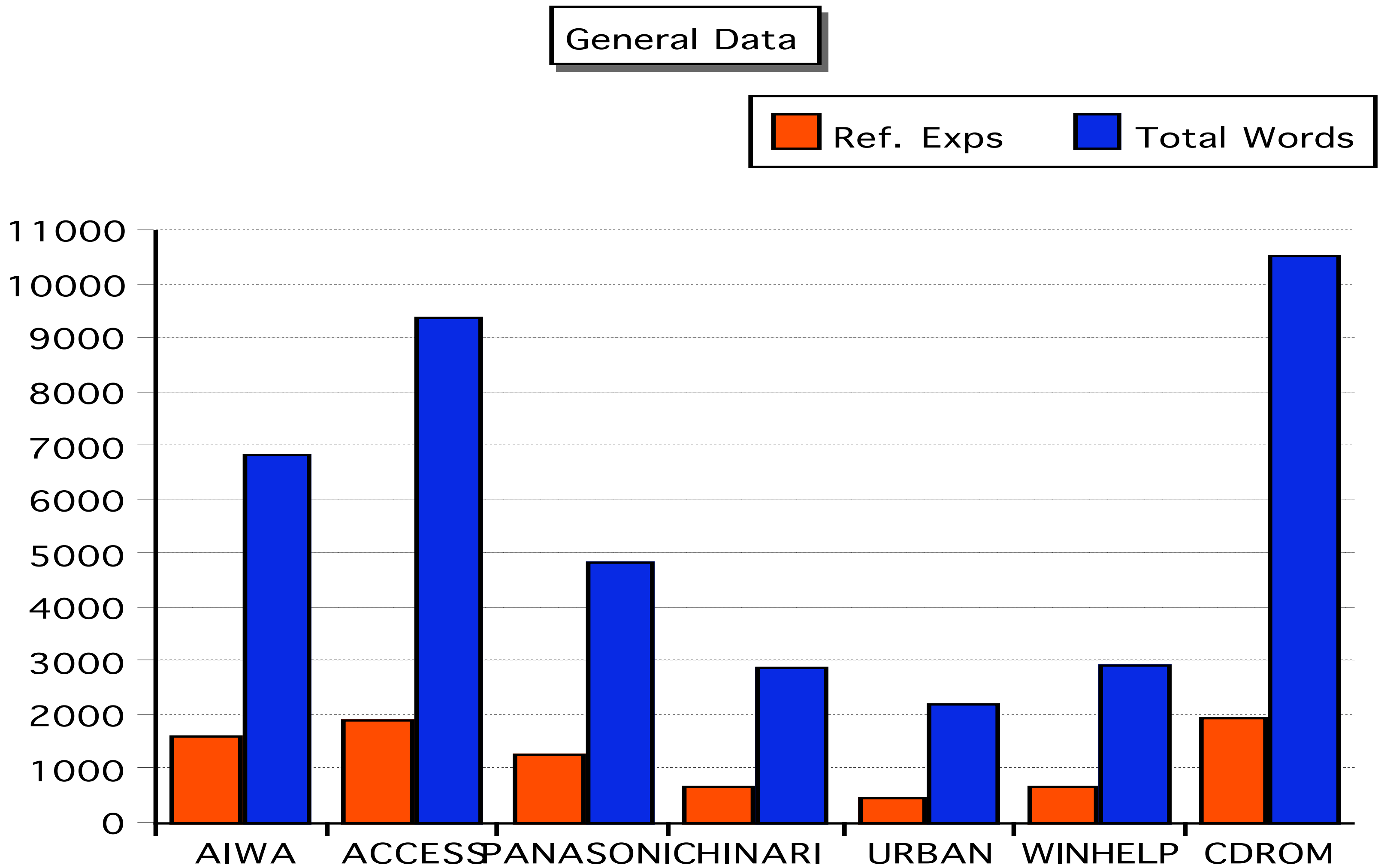
Evaluation


Referring Expressions as Function of Number of Words
Ref. Exps
Coref. Exps
Total Words
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
AIWA
ACCESS
PANASONIC
HINARI
URBAN
WINHELP
CDROM

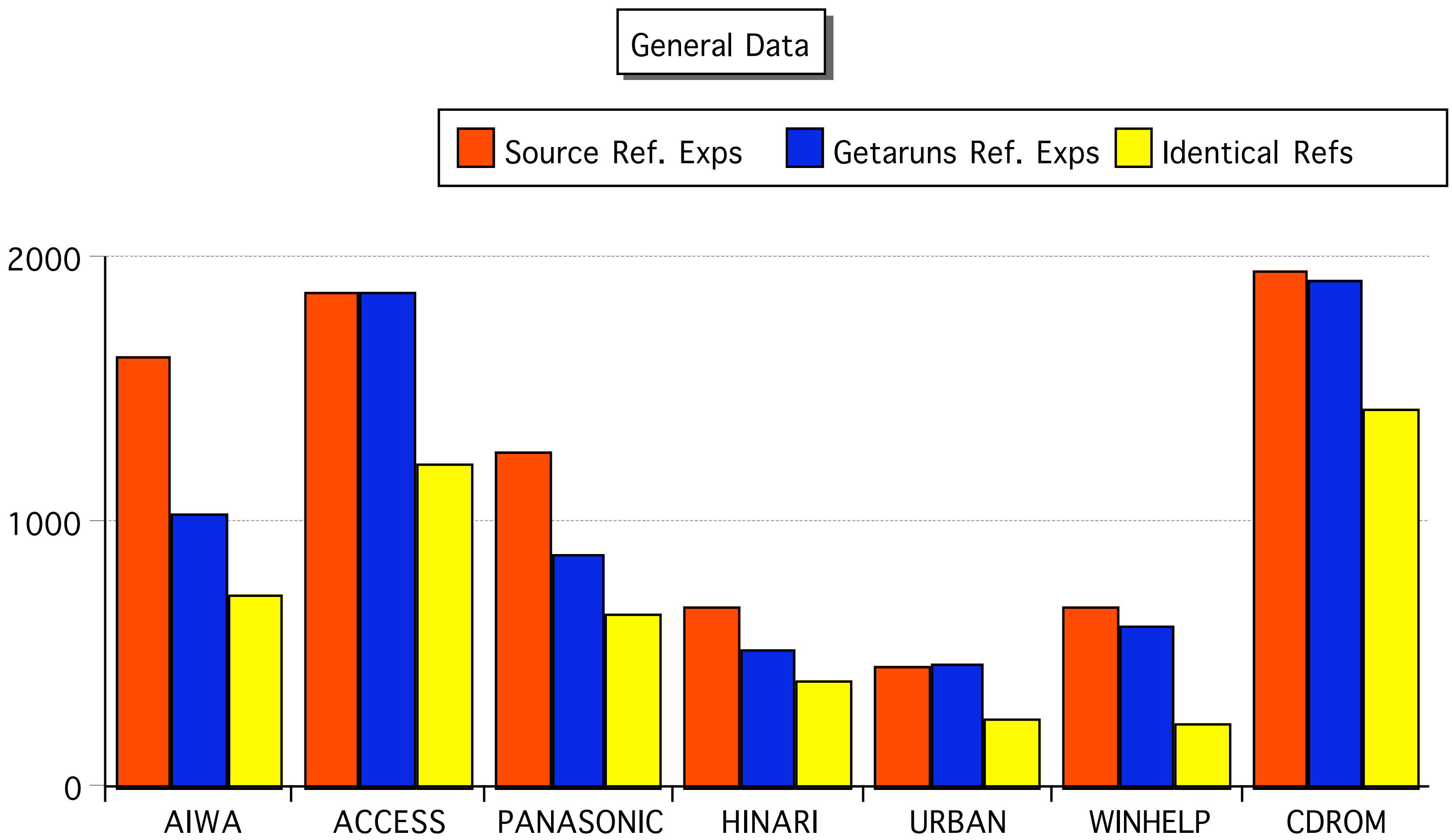
Evaluation - Coreference Corpus



Evaluation - Coreference Corpus



Comparison


General Data
Source Ref. Exps
Getaruns Ref. Exps
Identical Refs
2000
1000
0
AIWA
ACCESS
PANASONIC
HINARI
URBAN
WINHELP
CDROM

Precision & Recall