

# Tractor

## Toward Deep Understanding of Short Intelligence Messages

Stuart C. Shapiro

Department of Computer Science and Engineering  
Center for Cognitive Science,  
Center for Multisource Information Fusion  
University at Buffalo, The State University of New York  
Buffalo, New York 14260  
shapiro@buffalo.edu



# Outline

- 1 Acknowledgements
- 2 Introduction
- 3 Syntactic Processing
- 4 Human Coreference Editing
- 5 From Annotations to Syntactic Propositional Graph
- 6 CBIR
- 7 SNePS 3 and its GUI
- 8 Syntax-Semantics Mapping
- 9 Statistics
- 10 Performance on Motivational Example
- 11 Summary
- 12 Further Reading

# Acknowledgements

Daniel R. Schlegel, James M. Dobler  
Michael Prentice, Paul Bunter, Michael W. Kandefer

Rakesh Nagi, Kedar Sambhoos, Geoff A. Gross, Greg Tauer

Multidisciplinary University Research Initiative (MURI) Grant  
(Number W911NF-09-1-0392)  
for “Unified Research on Network-based Hard/Soft Information Fusion”  
Issued by the US Army Research Office (ARO)  
under the program management of Dr. John Lavery.

# Tractor I/O

**Input:** A short English message

- Mostly 1–3 sentences.
- In Counter-insurgency domain.
- Written by human informant or intelligence gatherer.
- Not necessarily “grammatical” English.

**Output:** Semantic Propositional Graph

- Representing information in the message.
- Nodes for Entities, Events, Actions, Categories, Properties, Property Values, Propositions, ...
- Edges represent non-conceptual relations.

# Tractor I/O

**Input:** A short English message

- Mostly 1–3 sentences.
- In Counter-insurgency domain.
- Written by human informant or intelligence gatherer.
- Not necessarily “grammatical” English.

**Output:** Semantic Propositional Graph

- Representing information in the message.
- Nodes for Entities, Events, Actions, Categories, Properties, Property Values, Propositions, ...
- Edges represent non-conceptual relations.

# Important Assumption

- A single message is written by a single person at a single time.
- Different messages might be written by different people at different times, without the author of one message being aware of the contents of previous messages.
- Therefore,
  - NLP techniques are appropriate for intra-message coreference resolution,
  - inter-message coreference resolution must be based on semantic descriptions of the various entities.
- Tractor must supply those semantic descriptions.

# Important Assumption

- A single message is written by a single person at a single time.
- Different messages might be written by different people at different times, without the author of one message being aware of the contents of previous messages.
- Therefore,
  - NLP techniques are appropriate for intra-message coreference resolution,
  - inter-message coreference resolution must be based on semantic descriptions of the various entities.
- Tractor must supply those semantic descriptions.

# Important Assumption

- A single message is written by a single person at a single time.
- Different messages might be written by different people at different times, without the author of one message being aware of the contents of previous messages.
- Therefore,
  - NLP techniques are appropriate for intra-message coreference resolution,
  - inter-message coreference resolution must be based on semantic descriptions of the various entities.
- Tractor must supply those semantic descriptions.

# Important Assumption

- A single message is written by a single person at a single time.
- Different messages might be written by different people at different times, without the author of one message being aware of the contents of previous messages.
- Therefore,
  - NLP techniques are appropriate for intra-message coreference resolution,
  - inter-message coreference resolution must be based on semantic descriptions of the various entities.
- Tractor must supply those semantic descriptions.

# Important Assumption

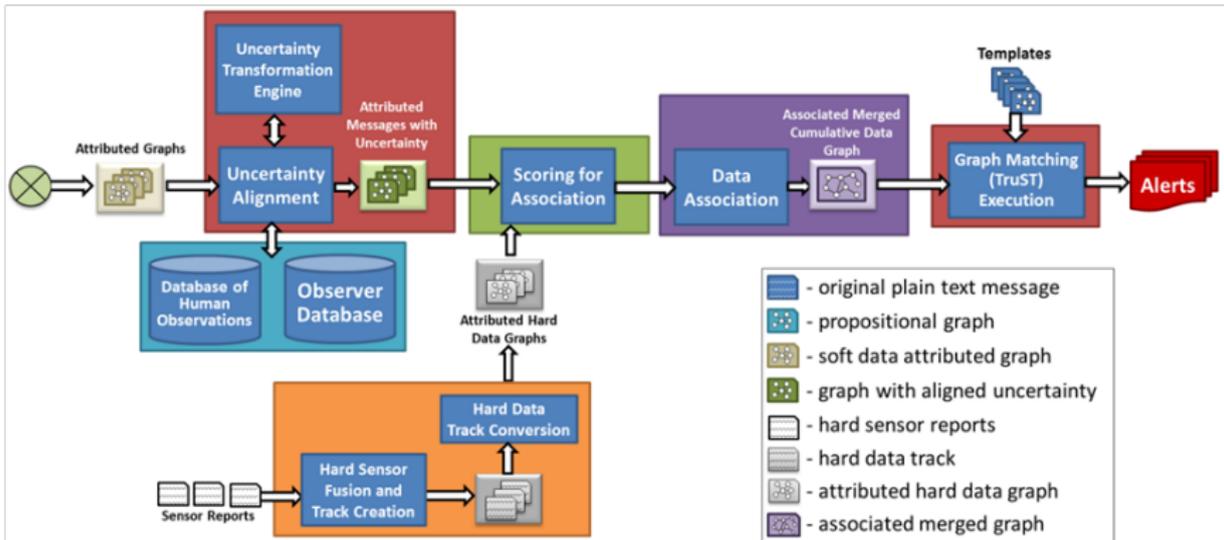
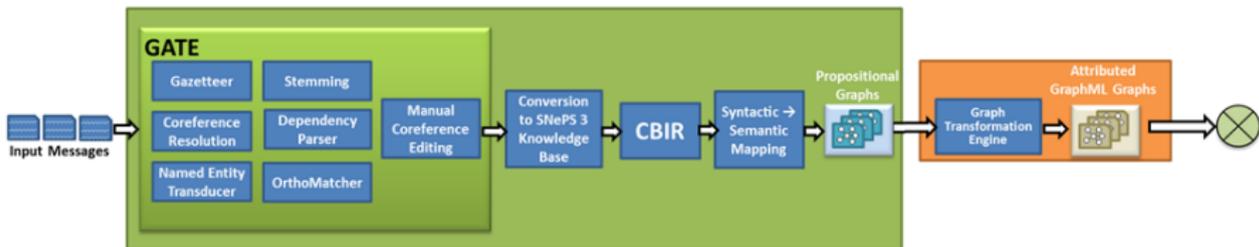
- A single message is written by a single person at a single time.
- Different messages might be written by different people at different times, without the author of one message being aware of the contents of previous messages.
- Therefore,
  - NLP techniques are appropriate for intra-message coreference resolution,
  - inter-message coreference resolution must be based on semantic descriptions of the various entities.
- Tractor must supply those semantic descriptions.

# Why Shortness Matters

Since the messages are short,  
no need to be concerned with

- rhetorical relations
- topic shifts
- etc.

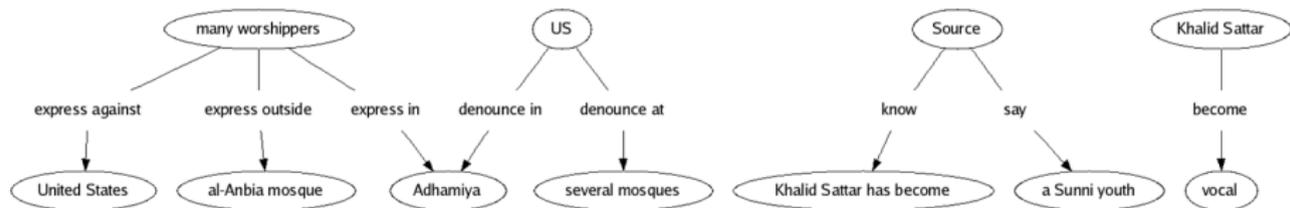
# Hard+Soft Information Fusion Architecture



## Motivational Example: STEF Messages

- 01/05/07 - Increased hostile sentiment being expressed against U.S. troops by many worshippers outside the al-Anbia mosque in Adhamiya.
- 01/06/07 - Source said a Sunni youth he knows, Khalid Sattar, has become increasingly vocal in denouncing the U.S. at several mosques in Adhamiya.

# Motivational Example: STEF Graphs



[K. Sambhoos, J. Llinas, & E. Little, Graphical Methods for Real-Time Fusion and Estimation with Soft Message Data, 11th International Conference on Information Fusion, 2008.]

## Questions:

- Relation between “US” and “United States”?
- What was expressed?
- Who denounced whom?
- Does Source know Khalid Sattar?
- Is Khalid Sattar a Sunni youth?
- What did Source say?
- When did these events occur?

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
  - Human Coreference Editing
  - XML Syntactic File of Annotations
  - Convert XML to SNePS 3
  - SNePS 3 Syntactic Propositional Graph
  - Add Contextually Relevant Background/Ontological Information
  - Syntax-Semantics Mapping
  - SNePS 3 Semantic Propositional Graph
  - Express as GraphML
  - Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

# Tractor Processing Stream

- Short English Message
- Syntactic Processing
- Human Coreference Editing
- XML Syntactic File of Annotations
- Convert XML to SNePS 3
- SNePS 3 Syntactic Propositional Graph
- Add Contextually Relevant Background/Ontological Information
- Syntax-Semantics Mapping
- SNePS 3 Semantic Propositional Graph
- Express as GraphML
- Semantic Propositional Graph in GraphML Format

## Example Message

1. 01/31/2010, 0700 hrs. – Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

# GATE (General Architecture for Text Engineering)

- 1 Character Sequence to English Text
  - 1 ANNIE (a Nearly-New Information Extraction System) Tokenizer
  - 2 ANNIE Sentence Splitter
- 2 Morphology
  - 1 ANNIE POS Tagger
  - 2 English Stemmer
- 3 Named Entity Recognition
  - 1 List-Based "Gazetteer"
  - 2 ANNIE Rule-Base NE Transducer
- 4 Coreferencers
  - 1 ANNIE OrthoMatcher
  - 2 ANNIE Pronominal Coreferencer
- 5 Stanford Dependency Parser

# GATE (General Architecture for Text Engineering)

- 1 Character Sequence to English Text
  - 1 ANNIE (a Nearly-New Information Extraction System) Tokenizer
  - 2 ANNIE Sentence Splitter
- 2 Morphology
  - 1 ANNIE POS Tagger
  - 2 English Stemmer
- 3 Named Entity Recognition
  - 1 List-Based "Gazetteer"
  - 2 ANNIE Rule-Base NE Transducer
- 4 Coreferencers
  - 1 ANNIE OrthoMatcher
  - 2 ANNIE Pronominal Coreferencer
- 5 Stanford Dependency Parser

# GATE (General Architecture for Text Engineering)

- 1 Character Sequence to English Text
  - 1 ANNIE (a Nearly-New Information Extraction System) Tokenizer
  - 2 ANNIE Sentence Splitter
- 2 Morphology
  - 1 ANNIE POS Tagger
  - 2 English Stemmer
- 3 Named Entity Recognition
  - 1 List-Based “Gazetteer”
  - 2 ANNIE Rule-Base NE Transducer
- 4 Coreferencers
  - 1 ANNIE OrthoMatcher
  - 2 ANNIE Pronominal Coreferencer
- 5 Stanford Dependency Parser

# GATE (General Architecture for Text Engineering)

- ① Character Sequence to English Text
  - ① ANNIE (a Nearly-New Information Extraction System) Tokenizer
  - ② ANNIE Sentence Splitter
- ② Morphology
  - ① ANNIE POS Tagger
  - ② English Stemmer
- ③ Named Entity Recognition
  - ① List-Based “Gazetteer”
  - ② ANNIE Rule-Base NE Transducer
- ④ Coreferencers
  - ① ANNIE OrthoMatcher
  - ② ANNIE Pronominal Coreferencer
- ⑤ Stanford Dependency Parser

# GATE (General Architecture for Text Engineering)

- ① Character Sequence to English Text
  - ① ANNIE (a Nearly-New Information Extraction System) Tokenizer
  - ② ANNIE Sentence Splitter
- ② Morphology
  - ① ANNIE POS Tagger
  - ② English Stemmer
- ③ Named Entity Recognition
  - ① List-Based “Gazetteer”
  - ② ANNIE Rule-Base NE Transducer
- ④ Coreferencers
  - ① ANNIE OrthoMatcher
  - ② ANNIE Pronominal Coreferencer
- ⑤ Stanford Dependency Parser

# Results of Tokenizer

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Snowball Stemmer
  - Stanford Parser
  - ANNIE Pronominal Corefer
  - ANNIE Nominal Coreferen
  - ANNIE OrthoMatcher
  - ANNIE NE Transducer
  - ANNIE POS Tagger
  - ANNIE Sentence Splitter
  - Syncoin Gazetteer
  - ANNIE English Tokeniser

Messages Tractor App bbs1.txt\_0001B

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1, 01/31/2010, 0700 hrs. - Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

- Date
- Dependency
- FirstPerson
- JobTitle
- Location
- Lookup
- Organization
- Person
- Sentence
- SpaceToken
- Split
- SyntaxTreeNode
- Token
- ▶ Original markups

C MatchesAnnots {null-[599, 6

C Mime type text/plain

C docNewLineType LF

C gate.SourceURL file:/shared/pr

C

New

# Results of Sentence Splitter

The screenshot shows the GATE Developer 7.0 build 4195 interface. The main window displays the text from the file 'bbs1.txt\_0001B'. The text is as follows:

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

The text is annotated with colored boxes representing different annotation sets. The 'Sentence' annotation set is checked in the right-hand pane, and its results are visible as blue highlights around the two sentences in the text area. Other annotation sets like 'Date', 'Dependency', 'FirstPerson', etc., are listed but not checked.

At the bottom of the window, there is a table with the following data:

C	MatchesAnnots	{null-[599, 6
C	MimeType	text/plain
C	docNewLineType	LF
C	gate.SourceURL	file:/shared/pr
C		

# Results of POS Tagger & Stemmer

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Snowball Stemmer
  - ANNIE POS Tagger
  - ANNIE Sentence Splitter
  - Syncoin Gazetteer
  - ANNIE English Tokeniser
  - Document Reset PR
- Datastores

Messages Tractor App bbs1.txt\_0001B

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. - Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

- Lookup
- Sentence
- SpaceToks
- Split
- Token
- ▶ Original m

Type	Set	Start	End	Id	Featu
Token	94	100	1304	{category=NN, kind=word, length=6, orth=lowercase, stem=policy, string=policy}	
Token	100	101	1305	{category=., kind=punctuation, length=1, stem=., string=}	
Token	102	107	1307	{category=JJ, kind=word, length=5, orth=lowercase, stem=local, string=local}	
Token	108	120	1309	{category=JJ, kind=word, length=12, orth=lowercase, stem=presidential, string=pr	
Token	121	130	1311	{category=NN, kind=word, length=9, orth=lowercase, stem=candidate, string=can	
Token	131	135	1313	{category=NNP, kind=word, length=4, orth=upperInitial, stem=azam, string=Azam	
Token	136	139	1315	{category=NNP, kind=word, length=3, orth=upperInitial, stem=al-, string=Al-}	
Token	139	144	1316	{category=NNP, kind=word, length=5, orth=upperInitial, stem=azhar, string=Azhar	
Token	145	148	1318	{category=VBZ, kind=word, length=3, orth=lowercase, stem=has, string=has}	
Token	149	155	1320	{category=VBN, kind=word, length=6, orth=lowercase, stem=call, string=called}	
Token	156	159	1322	{category=IN, kind=word, length=3, orth=lowercase, stem=for, string=for}	
Token	160	161	1324	{category=DT, kind=word, length=1, orth=lowercase, stem=a, string=a}	

73 Annotations (1 selected). Select:

MatchesAnnots {null-[]}

MimeType text/plain

docNewLineType LF

gate.SourceURL file:sharec

New

# Results of "Gazetteer-Based" Named Entity Recognition

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Syncoin Gazetteer
  - ANNIE English Tokeniser
  - Document Reset PR
- Datastores

Messages bbs1.txt\_0001B Tractor App

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al Azhar has called for a protest at the Second District Courthouse. Al Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

Lookup  
 SpaceToken  
 Token  
 ▶ Original markings

Type	Set	Start	End	Id
Lookup		9	13	1562 {majorType=year}
Lookup		28	36	1563 {majorType=organization, minorType=newspaper}
Lookup		31	36	1564 {majorType=person first, minorType=female}
Lookup		121	130	1565 {majorType=jobtitle}
Lookup		131	144	1566 {majorType=person_full}
Lookup		136	138	1567 {majorType=person first, minorType=male}
Lookup		139	144	1568 {majorType=person first, minorType=male}
Lookup		177	203	1569 {majorType=facility, minorType=building}
Lookup		184	192	1570 {majorType=loc key, minorType=post}
Lookup		205	207	1571 {majorType=person first, minorType=male}
Lookup		208	213	1572 {majorType=person first, minorType=male}
Lookup		246	250	1573 {majorType=time modifier}
Lookup		275	284	1574 {majorType=jobtitle pl}
Lookup		310	315	1575 {majorType=color}
Lookup		316	319	1576 {majorType=vehicle}

C MatchesAnnots {null=□}  
 C MimeType text/plain  
 C docNewLineType LF  
 C gate.SourceURL file/sharec

# Results of Rule-Based Named Entity Recognition

The screenshot displays the GATE Developer 7.0 build 4195 interface. The main window shows the 'Tractor App' processing the file 'bbs1.txt\_0001B'. The text in the central pane is as follows:

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

The text is annotated with colored boxes representing named entities: '01/31/2010' (Date), '0700 hrs.' (Time), 'Al Sabah' (Organization), 'candidate' (Person), 'Azam Al-Azhar' (Person), 'Second District Courthouse' (Location), 'Al-Azhar' (Person), 'he' (Person), 'his' (Person), and '1800 hrs.' (Time).

On the right side, a legend titled 'Original markups' lists various entity types with corresponding colored checkboxes:

- Date
- Dependency
- FirstPerson
- JobTitle
- Location
- Lookup
- Organization
- Person
- Sentence
- SpaceToken
- Split
- SyntaxTreeNode
- Token
- ▶ Original markups

The left-hand pane shows the 'GATE' tree structure with 'Tractor App' selected under 'Applications'. The bottom status bar shows configuration details for the current document:

- MatchesAnnots: {null-[1 75}
- MimeType: text/plain
- docNewLineType: LF
- gate.SourceURL: file://sharec

# Results of OrthoMatcher

The screenshot displays the GATE Developer interface. The main window shows the text from 'bbs1.txt\_0001B' with the following content:

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate **Azam Al-Azhar** has called for a protest at the Second District Courthouse. **Al-Azhar** said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

The **Co-reference Editor** panel on the right shows the following configuration:

- Sets: Default
- Types: Token
- Co-reference Data:
  - Default
    - Azam Al-Azhar

The bottom left panel shows the following properties:

C	MatchesAnnots	{null-[127
C	MimeType	text/plain
C	docNewLineType	LF
C	gate.SourceURL	file:/sharec
C		

# After Pronominal Coreferencer

GATE Developer 7.0 build 4195

File Options Tools Help

Messages Tractor App bbs1.txt\_0001B

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate **Azam Al-Azhar** has called for a protest at the Second District Courthouse. **Al-Azhar** said **he** would personally attend this protest, and that local residents should expect to see **his** black SUV arrive at the courthouse at around 1800 hrs.

Sets: Default

Types: Token Show

Co-reference Data

Default

Azam Al-Azhar

MatchesAnnots {null-[599,  
 Mime type text/plain  
 docNewLineType LF  
 gate.SourceURL file:/shared/p

# Result of Dependency Parse

GATE Developer 7.0 build 4195

File Options Tools Help

Messages bbs1.txt\_0001B Tractor App

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

Type	Set	Start	End	Id
Dependency		28	46	2910 { args=[2622, 2618], kind=nn }
Dependency		31	46	2911 { args=[2622, 2620], kind=nn }
Dependency		37	54	2912 { args=[2624, 2622], kind=nsobj }
Dependency		47	155	2928 { args=[2624, 2656], kind=ccomp }
Dependency		55	155	2913 { args=[2656, 2626], kind=complm }
Dependency		60	155	2914 { args=[2656, 2628], kind=prep }
Dependency		60	71	2915 { args=[2628, 2630], kind=pobj }
Dependency		63	74	2916 { args=[2630, 2632], kind=prep }
Dependency		72	100	2920 { args=[2632, 2640], kind=pobj }
Dependency		75	100	2917 { args=[2640, 2634], kind=det }
Dependency		79	100	2918 { args=[2640, 2636], kind=amod }
Dependency		83	100	2919 { args=[2640, 2638], kind=nn }
Dependency		102	144	2921 { args=[2652, 2643], kind=amod }
Dependency		108	144	2922 { args=[2652, 2645], kind=amod }
Dependency		121	144	2923 { args=[2652, 2647], kind=nn }
Dependency		131	144	2924 { args=[2652, 2649], kind=nn }
Dependency		136	144	2925 { args=[2652, 2651], kind=nn }
Dependency		139	155	2926 { args=[2656, 2652], kind=nsobj }
Dependency		145	155	2927 { args=[2656, 2654], kind=aux }
Dependency		149	159	2929 { args=[2656, 2658], kind=prep }
Dependency		156	169	2931 { args=[2658, 2662], kind=pobj }
Dependency		160	169	2930 { args=[2662, 2660], kind=det }
Dependency		162	172	2932 { args=[2662, 2664], kind=prep }
Dependency		170	202	2933 { args=[2664, 2673], kind=pobj }

54 Annotations (0 selected). Select:

- Date
- Dependency
- FirstPerson
- JobTitle
- Location
- Lookup
- Organization
- Person
- Sentence
- SpaceToken
- Split
- SyntaxTreeNode
- Token
- Original markups

MatchesAnnots { null--[[277

MimeType text/plain

docNewLineType LF

gate.SourceURL file://share

# Co-reference Editor

## Initial Chain

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Snowball Stemmer
  - Stanford Parser
  - ANNIE Pronominal Coreferencer
  - ANNIE Nominal Coreferencer
  - ANNIE OrthoMatcher
  - ANNIE NE Transducer
  - ANNIE POS Tagger
  - ANNIE Sentence Splitter
  - Syncoin Gazetteer
  - ANNIE English Tokeniser

Messages bbs1.txt\_0001B Tractor App

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

Sets: Default

Types: Token Show

Co-reference Data

- Default
  - Azam Al-Azhar

MatchesAnnots {null-[I27

MimeType text/plain

docNewLineType LF

gate.SourceURL file://sharec

# Co-reference Editor

## Adding a Chain

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Snowball Stemmer
  - Stanford Parser
  - ANNIE Pronominal Core
  - ANNIE Nominal Corefer
  - ANNIE OrthoMatcher
  - ANNIE NE Transducer
  - ANNIE POS Tagger
  - ANNIE Sentence Splitte
  - Syncoin Gazetteer
  - ANNIE English Tokenise

Messages bbs1.txt\_0001B Tractor App

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

Sets: Default

Types: Lookup Show

Co-reference Data

- Default
  - Azam Al-Azhar

Add "Second District Courthouse" to [New Chain]

OK Cancel

MatchesAnnots {null-[277

MimeType text/plain

# Co-reference Editor

## Adding to a Chain

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Snowball Stemmer
  - Stanford Parser
  - ANNIE Pronominal Coreferencer
  - ANNIE Nominal Coreferencer
  - ANNIE OrthoMatcher
  - ANNIE NE Transducer
  - ANNIE POS Tagger
  - ANNIE Sentence Splitter
  - Syncoin Gazetteer
  - ANNIE English Tokeniser

Messages bbs1.txt\_0001B Tractor App

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1: 01/31/2010, 0700 hrs. Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

Sets: Default

Types: Token Show

Co-reference Data

- Default
  - Azam Al-Azhar
  - Second District Courthouse
  - protest

Add "courthouse" to

Second District Courthouse

OK Cancel

MatchesAnnots {null-[277

MimeType text/plain

# Co-reference Editor

## Final Chains

GATE Developer 7.0 build 4195

File Options Tools Help

GATE

- Applications
  - Tractor App
- Language Resources
  - bbs1.txt\_0001B
  - BombBuster
- Processing Resources
  - Snowball Stemmer
  - Stanford Parser
  - ANNIE Pronominal Coreferencer
  - ANNIE Nominal Coreferencer
  - ANNIE OrthoMatcher
  - ANNIE NE Transducer
  - ANNIE POS Tagger
  - ANNIE Sentence Splitter
  - Syncoin Gazetteer
  - ANNIE English Tokeniser

Messages bbs1.txt\_0001B Tractor App

Annotation Sets Annotations List Annotations Stack Co-reference Editor Text

1. 01/31/2010, 0700 hrs. -- Al Sabah newspaper reports that in response to the new government policy, local presidential candidate Azam Al-Azhar has called for a protest at the Second District Courthouse. Al-Azhar said he would personally attend this protest, and that local residents should expect to see his black SUV arrive at the courthouse at around 1800 hrs.

Sets: Default

Types: Token Show

Co-reference Data

- Default
  - Azam Al-Azhar
  - Second District Courthouse
  - protest

MatchesAnnots {null-[I277}

MimeType text/plain

docNewLineType LF

gate.SourceURL file://sharec

## XML to SNePS 3

```

XML <Annotation Id="510" Type="Token" StartNode="251" EndNode="258">
  <Feature>
    <Name className="java.lang.String">string</Name>
    <Value className="java.lang.String">protest</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">category</Name>
    <Value className="java.lang.String">NN</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">dependencies</Name>
    <Value className="java.util.ArrayList" itemClassName="java.lang.String">
      det (569)</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">matches</Name>
    <Value className="java.util.ArrayList" itemClassName="java.lang.Integer">
      482;510</Value>
  </Feature>
</Annotation>

```

```

SNePS 3 (assert `(token-start-pos 510 251))
(assert `(token-end-pos 510 258))
(assert `(TextOf |protest| 510))
(assert `(SyntacticCategoryOf "NN" 510))
(assert `(det 510 569))
(assert `(Equiv (setof 482 510)))

```

# xml2sneps3

Besides translation, `xml2sneps3`

- combines annotations that cover same substring into one token,
- adjusts message times to GMT,
- adjusts message dates as necessary given time change,
- converts message dates to ISO format.

# Adding Contextually Relevant Background/Ontological Information

For each noun in the graph

- Finds the Cyc concept from ResearchCyc
- Loads the ontology above it in OpenCyc into the graph

# SNePS 3

SNePS 3 is the latest member of the SNePS Family of KRR systems.

It is still being implemented.

The SNePS 3 KB can be thought of as simultaneously being:

- Logic based,
- Frame based, and
- Graph based.

We have created a user interface which uses all three:

- Assertions and queries of a KB are handled using logic or frames.
- Visualization and inspection is done using propositional graphs.

## SNePS 3 GUI

File Graph SNePS Help

REPL

Graph View

Add CF Graph Find Mouse: Picking Lens Collapse Zoom: + - Reset

Ahmed

My brother sends greetings.

Source

Theme

Location

Resident

member

class

Communication

Communicator

Resident

member

Location

class

Person

Adhamiya

# Graph View

# Frames

Logical Interaction

wft3!: (LocationOf Sufian Adhamiya)  
(assert '(Isa Ziyad Person))

wft4!: (Isa Ziyad Person)  
(assert '(SourceOf Ahmed (LocationOf Ziyad Adhamiya))

wft6!: (SourceOf Ahmed (LocationOf Ziyad Ramadi))

Semantic Types

- Semantic Types
  - Entity
    - Act
    - Policy
    - Proposition
  - Thing
    - Action

Casesframes

- Casesframes
  - and
  - andor
  - if
  - Isa
  - LocationOf

Contexts

- Contexts
  - BaseCT
    - DefaultCT

S. C. Shapiro (UB) Tractor 28 March 2012 29 / 70

# KB as set of Logical Expressions

The SNePS 3 KB is a set of logical expressions:

- Atomic terms
  - Individual constants denoting entities in domain including some relations
- Arbitrary and indefinite terms [Shapiro, KR2004]
- Functional terms including
  - terms denoting atomic propositions
  - terms denoting non-atomic propositions

Use CLIF syntax.

Every logical expression is a term.

Allows propositions about propositions without leaving First-Order.

Internal name of functional terms: *wft/ [!]*

for “well-formed term”.

# KB as set of Logical Expressions

The SNePS 3 KB is a set of logical expressions:

- Atomic terms
  - Individual constants denoting entities in domain including some relations
- Arbitrary and indefinite terms [Shapiro, KR2004]
- Functional terms including
  - terms denoting atomic propositions
  - terms denoting non-atomic propositions

Use CLIF syntax.

Every logical expression is a term.

Allows propositions about propositions without leaving First-Order.

Internal name of functional terms: *wft/ [!]*

for “well-formed term”.

# KB as set of Logical Expressions

The SNePS 3 KB is a set of logical expressions:

- Atomic terms
  - Individual constants denoting entities in domain including some relations
- Arbitrary and indefinite terms [Shapiro, KR2004]
- Functional terms including
  - terms denoting atomic propositions
  - terms denoting non-atomic propositions

Use CLIF syntax.

Every logical expression is a term.

Allows propositions about propositions without leaving First-Order.

Internal name of functional terms: *wft/ [!]*

for “well-formed term”.

# KB as set of Logical Expressions

The SNePS 3 KB is a set of logical expressions:

- Atomic terms
  - Individual constants denoting entities in domain including some relations
- Arbitrary and indefinite terms [Shapiro, KR2004]
- Functional terms including
  - terms denoting atomic propositions
  - terms denoting non-atomic propositions

Use CLIF syntax.

Every logical expression is a term.

Allows propositions about propositions without leaving First-Order.

Internal name of functional terms: *wft/ [!]*

for “well-formed term”.

# KB as set of Logical Expressions

The SNePS 3 KB is a set of logical expressions:

- Atomic terms
  - Individual constants denoting entities in domain including some relations
- Arbitrary and indefinite terms [Shapiro, KR2004]
- Functional terms including
  - terms denoting atomic propositions
  - terms denoting non-atomic propositions

Use CLIF syntax.

Every logical expression is a term.

Allows propositions about propositions without leaving First-Order.

Internal name of functional terms: `wft/ [!]`

for “well-formed term”.

# KB as set of Logical Expressions

The SNePS 3 KB is a set of logical expressions:

- Atomic terms
  - Individual constants denoting entities in domain including some relations
- Arbitrary and indefinite terms [Shapiro, KR2004]
- Functional terms including
  - terms denoting atomic propositions
  - terms denoting non-atomic propositions

Use CLIF syntax.

Every logical expression is a term.

Allows propositions about propositions without leaving First-Order.

Internal name of functional terms: `wfti [!]`

for “well-formed term”.

# Caseframes

- Based on “The Case for Case” [Fillmore, 1968] and The Berkeley FrameNet Project

[Baker, Fillmore, & Lowe, 1998; Ruppenhofer *et al.*, 2010]

- Frame
  - schematic representation of a situation with a set of participants and conceptual roles.
- Eliminates syntactic differences.
- E.g.
  - Sufian called Ziyad.
  - Ziyad was called by Sufian.
  - a call from Sufian to Ziyad
- We will use “caseframe” for their “frame”
- and use “frame” for an instantiated caseframe.

# Caseframes

- Based on “The Case for Case” [Fillmore, 1968] and The Berkeley FrameNet Project

[Baker, Fillmore, & Lowe, 1998; Ruppenhofer *et al.*, 2010]

- Frame
  - schematic representation of a situation with a set of participants and conceptual roles.
- Eliminates syntactic differences.
- E.g.
  - Sufian called Ziyad.
  - Ziyad was called by Sufian.
  - a call from Sufian to Ziyad
- We will use “caseframe” for their “frame”
- and use “frame” for an instantiated caseframe.

# Caseframes

- Based on “The Case for Case” [Fillmore, 1968] and The Berkeley FrameNet Project

[Baker, Fillmore, & Lowe, 1998; Ruppenhofer *et al.*, 2010]

- Frame
  - schematic representation of a situation with a set of participants and conceptual roles.
- Eliminates syntactic differences.
- E.g.
  - Sufian called Ziyad.
  - Ziyad was called by Sufian.
  - a call from Sufian to Ziyad
- We will use “caseframe” for their “frame”
- and use “frame” for an instantiated caseframe.

# Caseframes

- Based on “The Case for Case” [Fillmore, 1968] and The Berkeley FrameNet Project

[Baker, Fillmore, & Lowe, 1998; Ruppenhofer *et al.*, 2010]

- Frame
  - schematic representation of a situation with a set of participants and conceptual roles.
- Eliminates syntactic differences.
- E.g.
  - Sufian called Ziyad.
  - Ziyad was called by Sufian.
  - a call from Sufian to Ziyad
- We will use “caseframe” for their “frame”
- and use “frame” for an instantiated caseframe.

# Components of Caseframes

## Definition

A caseframe has

- A name
- A sort
- An ordered list of slots
- ...

# Slots

Slots are defined globally independently of the caseframes that use them.

## Definition

A slot has

- A name
- Minimum and maximum number of fillers
- The sort of its fillers
- ...

# Examples of Caseframes

## Example

**Isa** is a caseframe of type `Proposition`  
with slots `member` and `class`.

## Example

**Call** is a caseframe of type `Proposition`  
with slots `Communicator`, `Addressee`, and `Communication`.

# Frames vs. Logical Terms

- A *frame* is an instance of a caseframe.
- The logical term  $(F \ x_1, \dots, x_n)$  is represented by an instance of the caseframe named  $F$  whose slots,  $s_1, \dots, s_n$  are filled by the representations of  $x_1, \dots, x_n$ , respectively.

## Frames vs. Logical Terms: Example

```
(assert ' (Call Sufian Ziyad  
          "My brother sends greetings"))
```

creates an instance of the `Call` caseframe

whose `Communicator` slot contains the filler `Sufian`,

whose `Addressee` slot contains `Ziyad`,

and whose `Communication` slot

contains `"My brother sends greetings"`.

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,  
to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
    - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti$  [!])
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti[!]$ )
- Edges drawn
  - from the node corresponding to the frame,  
to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

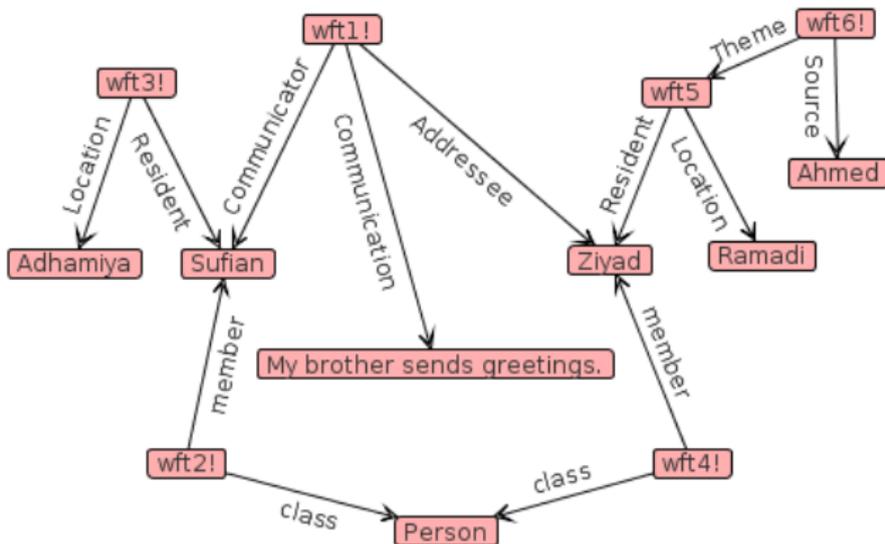
- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti[!]$ )
- Edges drawn
  - from the node corresponding to the frame,
  - to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Propositional Graphs

A way of visualizing and traversing the frames.

- Directed Acyclic Graph
- Every term is a node.
  - Individual constants
  - Functional terms (frames)
  - Proposition-denoting functional terms
- Node ID is
  - symbol
  - frame name ( $wfti[!]$ )
- Edges drawn  
from the node corresponding to the frame,  
to the nodes corresponding to the slot fillers
- Edges labeled by slot names

# Example Propositional Graph



*“Sufian, a person in Adhamiya, called Ziyad, a person who, according to Ahmed, is in Ramadi, saying ‘My brother sends greetings.’”*

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Graph View as Visualization

- Visualized graph is for human comprehension.
- Visualized graph need not be isomorphic to implementation of KB.
- Usefulness of `wft` nodes:
  - Functional term with more than two arguments (slots).
  - Functional term with more than one filler in a slot.
  - Functional term shown as argument of another (filler in a slot).
- Can show a binary relation with no arc coming into it as a labeled arc (“collapsed arc”).

# Visualizing a Collapsed Arc

- Slots in a frame are ordered.
- Order of slots = order of arguments of functional term.
- Draw collapsed arc from first argument to second argument.
- Name of caseframe = function symbol.
- Label collapsed arc with function symbol.
- Different style of arrow head  
so user knows it's a collapsed arc.

# Visualizing a Collapsed Arc

- Slots in a frame are ordered.
- Order of slots = order of arguments of functional term.
- Draw collapsed arc from first argument to second argument.
- Name of caseframe = function symbol.
- Label collapsed arc with function symbol.
- Different style of arrow head  
so user knows it's a collapsed arc.

# Visualizing a Collapsed Arc

- Slots in a frame are ordered.
- Order of slots = order of arguments of functional term.
- Draw collapsed arc from first argument to second argument.
- Name of caseframe = function symbol.
- Label collapsed arc with function symbol.
- Different style of arrow head  
so user knows it's a collapsed arc.

# Visualizing a Collapsed Arc

- Slots in a frame are ordered.
- Order of slots = order of arguments of functional term.
- Draw collapsed arc from first argument to second argument.
- Name of caseframe = function symbol.
- Label collapsed arc with function symbol.
- Different style of arrow head  
so user knows it's a collapsed arc.

# Visualizing a Collapsed Arc

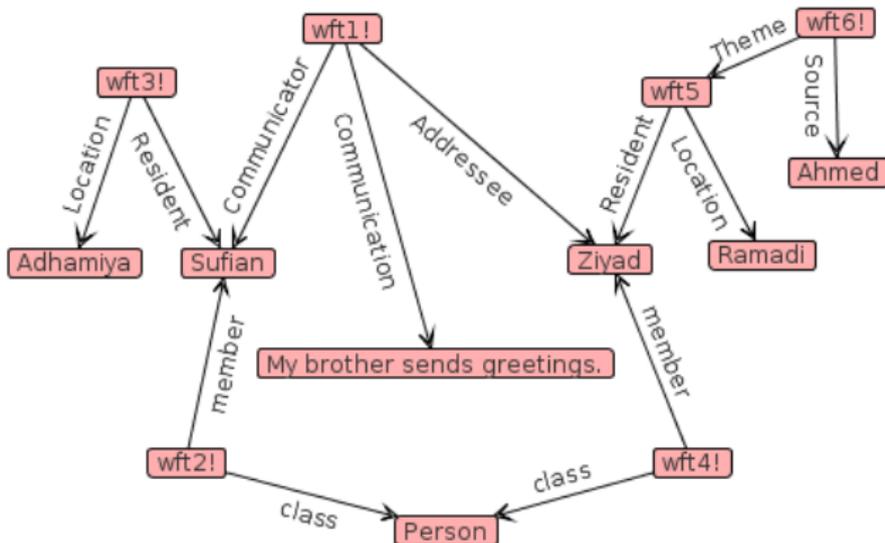
- Slots in a frame are ordered.
- Order of slots = order of arguments of functional term.
- Draw collapsed arc from first argument to second argument.
- Name of caseframe = function symbol.
- Label collapsed arc with function symbol.
- Different style of arrow head  
so user knows it's a collapsed arc.

## Visualizing a Collapsed Arc

- Slots in a frame are ordered.
- Order of slots = order of arguments of functional term.
- Draw collapsed arc from first argument to second argument.
- Name of caseframe = function symbol.
- Label collapsed arc with function symbol.
- Different style of arrow head  
so user knows it's a collapsed arc.

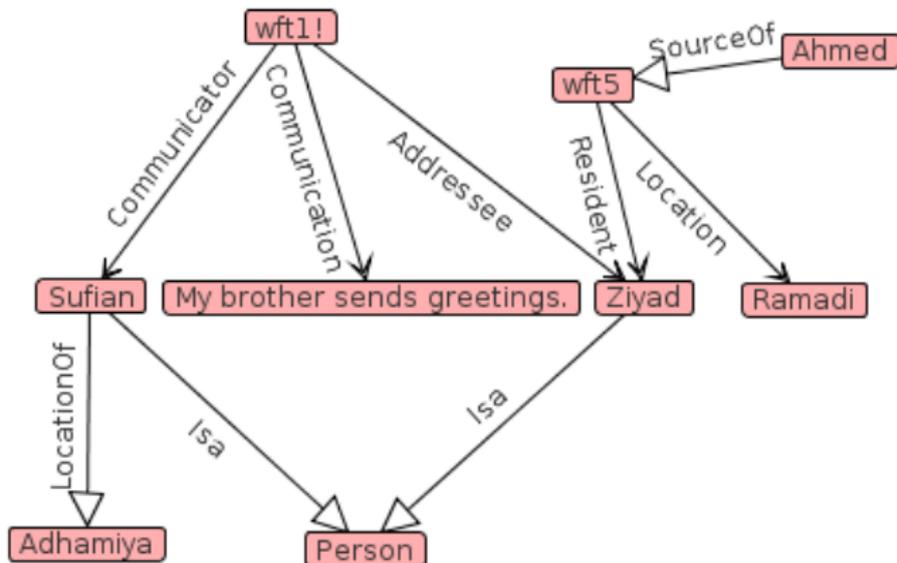
# Example of Collapsed Graph: Before

The uncollapsed version of Suifian calling Ziyad example:

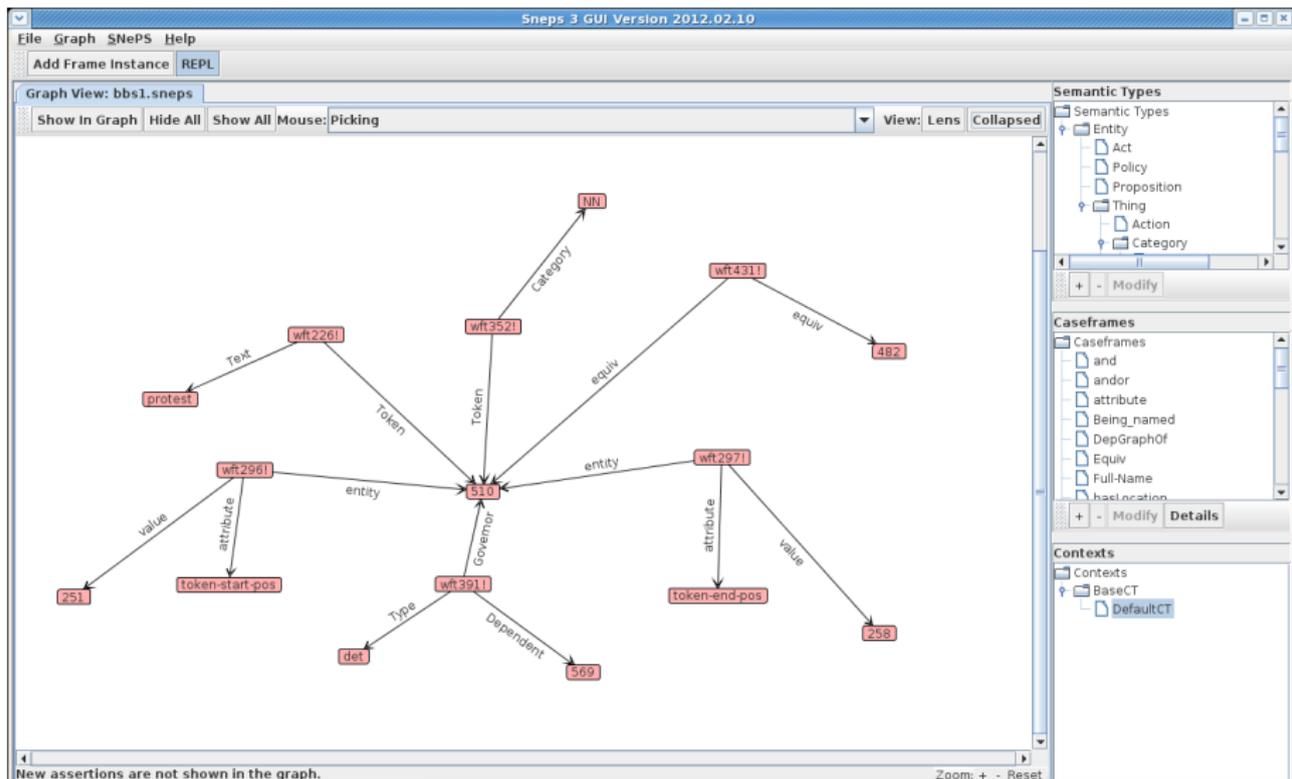


## Example of Collapsed Graph: After

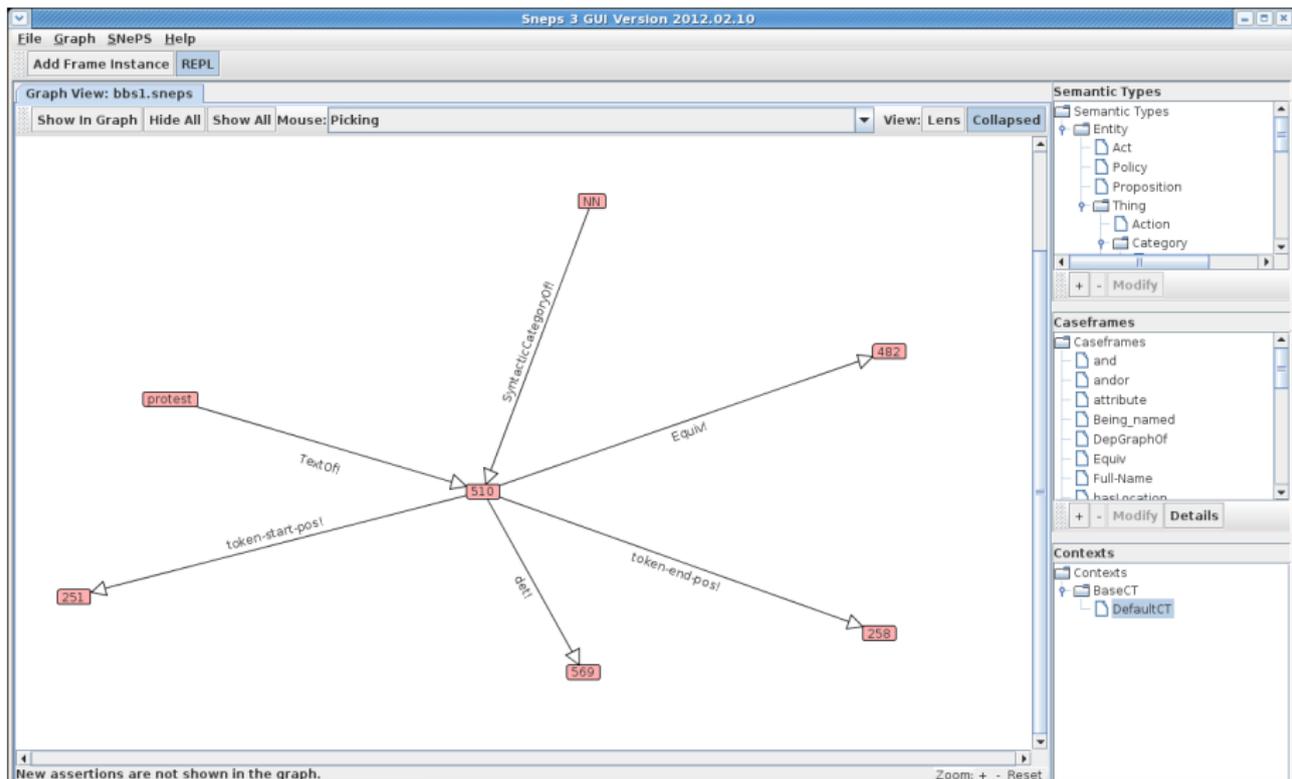
The collapsed version of Suifian calling Ziyad example:



# SNePS 3 Syntactic Graph Visualized (Uncollapsed)



# SNePS 3 Syntactic Graph Visualized (Collapsed)



# Mapping Rule: properNounToName

Sneps 3 GUI Version 2012.02.10

File Graph SNePS Help  
Add Frame Instance REPL

Graph View  
Show In Graph Hide All Show All Mouse: Picking View: Lens Collapsed

Semantic Types

- Entity
  - Act
  - Policy
  - Proposition
  - Thing
    - Action
    - Category

Caseframes

- and
- andor
- attribute
- Being\_named
- DepGraphOf
- Equiv
- Full-Name
- hasLocation

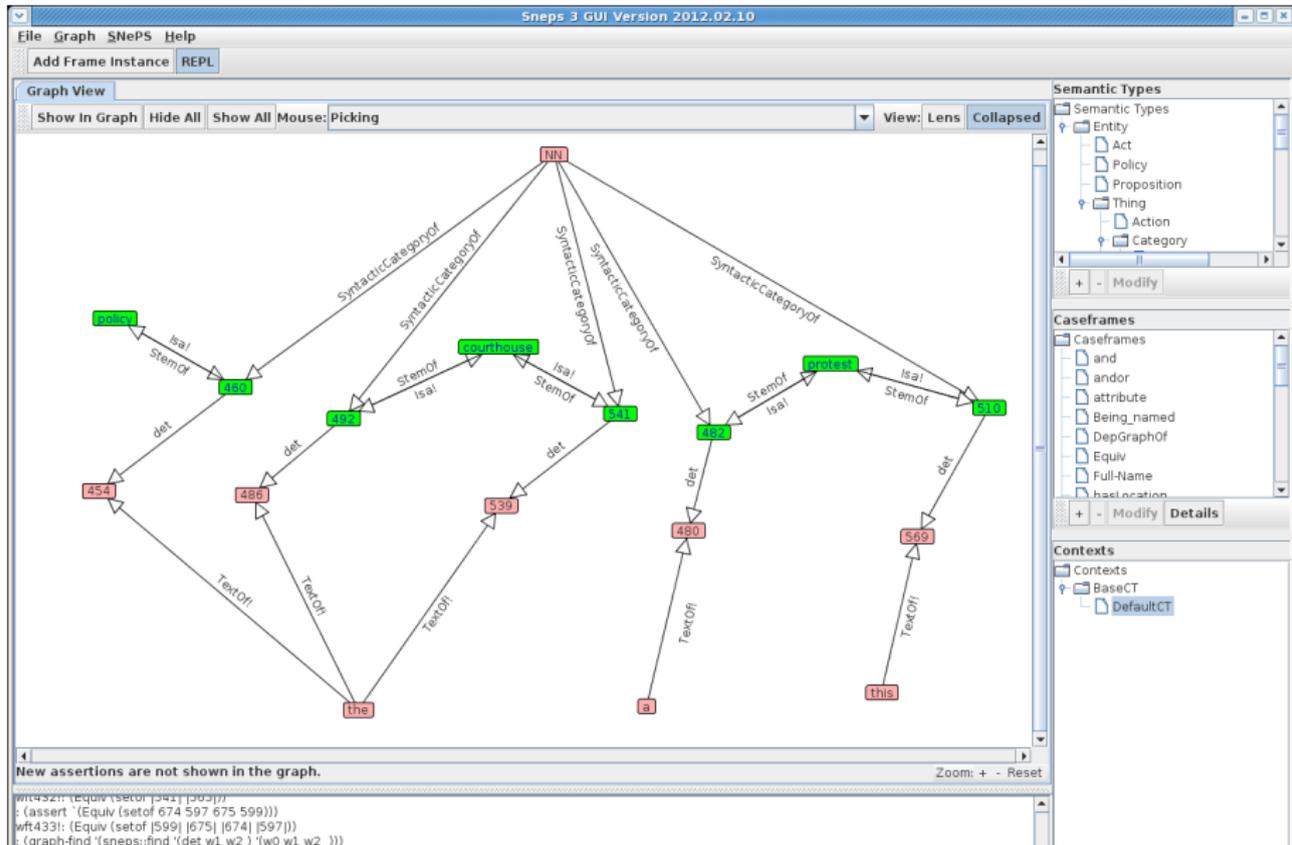
Contexts

- BaseCT
  - DefaultCT

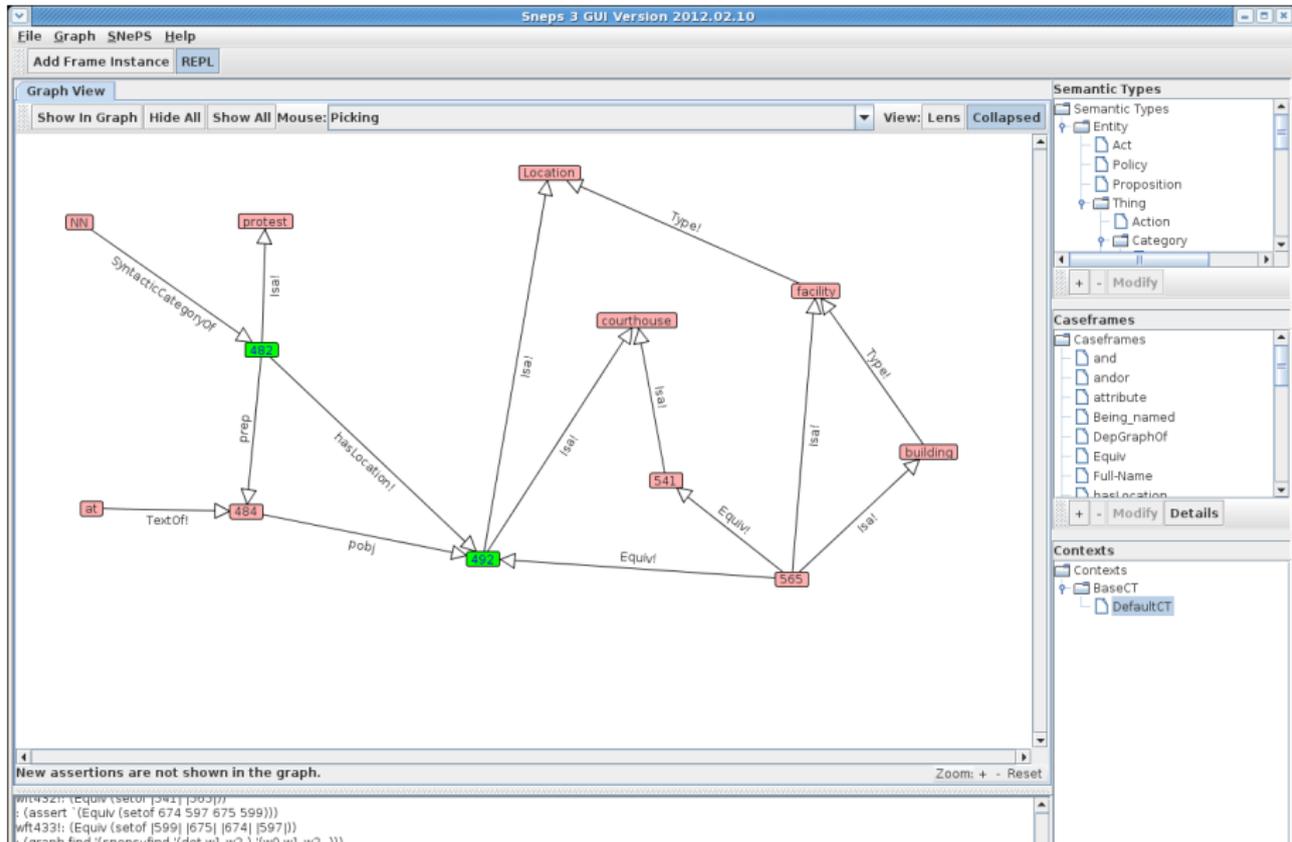
New assertions are not shown in the graph. Zoom: + - Reset

```
wft421: (Equiv (setof [482] [110] /))
(assert `(Equiv (setof [541] [565])))
wft432: (Equiv (setof [541] [565] /))
(assert `(Equiv (setof [674] [597] [675] [599])))
```

# Mapping Rule: nounPhraseToInstance



# Mapping Rule: atLocation



# Mapping Rule: colorProperty

Sneps 3 GUI Version 2012.02.10

File Graph SNePS Help

Add Frame Instance REPL

Graph View

Show In Graph Hide All Show All Mouse: Picking View: Lens Collapsed

Semantic Types

- Entity
  - Act
  - Policy
  - Proposition
  - Thing
    - Action
    - Category

Caseframes

- and
- andor
- attribute
- Being\_named
- DepGraphOf
- Equiv
- Full-Name
- has\_location

Contexts

- Contexts
  - BaseCT
    - DefaultCT

New assertions are not shown in the graph. Zoom: + - Reset

```

wft432: (Equiv (setof [w2] [color]))
: (assert `(Equiv (setof 674 597 675 599)))
wft433: (Equiv (setof [599] [675] [674] [597]))
: (graph-find '(sneps:find '(det w) w2) '(w0 w) w2 ))
  
```

# Mapping Rule: distribVerbOverConj

Sneps 3 GUI Version 2012.02.10

File Graph SNePS Help  
Add Frame Instance REPL

Graph View: bbs1.sneps  
Show In Graph Hide All Show All Mouse: Picking View: Lens Collapsed

Semantic Types

- Semantic Types
  - Entity
    - Act
    - Policy
  - Proposition
    - categorization-Proposi
    - molecular-Proposition
  - Thing

Caseframes

- Caseframes
  - and
  - and/or
  - attribute
  - Being\_named
  - DepGraphOf
  - Equiv
  - Full-Name
  - hasLocation

Contexts

- Contexts
  - BaseCT
    - DefaultCT

New assertions are not shown in the graph. Zoom: + - Reset

```
wft431: (equiv (setof [210] [582]))
(assert (Equiv (setof [541] [565])))
wft432: (Equiv (setof [541] [565]))
(assert (Equiv (setof [674] [597] [675] [599])))
wft433: (Equiv (setof [674] [675] [597] [599]))
```

# Mapping Rule: distribSubjOverXcomp

Sneps 3 GUI Version 2012.02.10

File Graph SNePS Help  
Add Frame Instance REPL

Graph View  
Show In Graph Hide All Show All Mouse: Picking View: Lens Collapsed

Semantic Types

- Semantic Types
  - Entity
    - Act
    - Policy
    - Proposition
  - Thing
    - Action
  - Category

Caseframes

- and
- andor
- attribute
- Being\_named
- DepGraphOf
- Equip
- Full-Name
- hasLocation
- hasPart

Contexts

- Contexts
  - BaseCT
    - DefaultCT

New assertions are not shown in the graph. Zoom: + - Reset

```

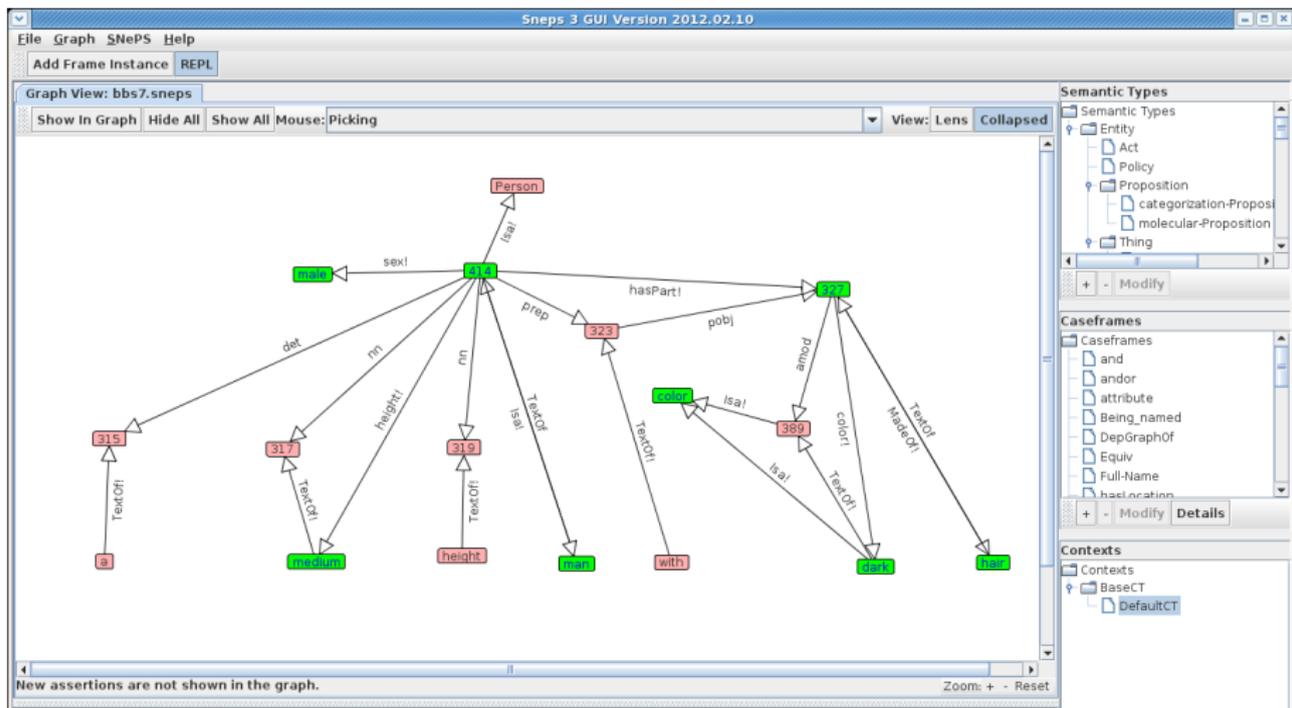
wft432: (Equip (setof [674] [675] [676]))
: (assert (Equip (setof 674 597 675 599)))
wft433: (Equip (setof [599] [675] [674] [597]))
: (graph-find '(sneps:find '(det w1 w2) '(w0 w1 w2 )))
  
```

## Another Example

7. 01/31/2010, 1817 hrs. – Blue team reports that **a medium height man with dark hair** just entered a blue car by the Second District Courthouse. The man was wearing a tan jacket. They are not sure where he came from.

# Mapping Rules:

nounPhraseToInstance; madeOfSubstance;  
colorProperty; hasDimensionValue; withTypicalPart



# CBIR Productivity

Over the 7 messages of the Bomber Buster data set

Noun types looked up: 80

Noun tokens: 165

Assertions added to graphs: 11,860

# Mapping Rule Usage

Rule	bbs1	bbs2	bbs3	bbs4	bbs5	bbs6	bbs7	Total
addRelevantOntology	1	1	1	1	1	1	2	8
substringCoreference	10	4	12	2	6	6	6	46
distribModOverConj	1	0	1	1	1	0	0	4
properNounToName	13	7	15	3	6	4	4	52
nounPhraseToInstance	5	9	4	12	4	2	5	41
madeOfSubstance	0	0	0	0	0	0	1	1
atLocation	1	0	0	0	1	1	0	3
modLocation	0	1	0	0	0	0	0	1
colorProperty	1	2	2	3	2	0	3	13
hasDimensionValue	0	1	0	0	0	0	1	2
withTypicalPart	0	0	0	0	0	0	1	1
distribVerbOverConj	1	0	1	0	1	0	0	3
passiveToActive	0	0	0	0	0	0	0	0
distribSubjOverXcomp	2	0	0	1	0	0	0	3
isalsEquiv	0	2	0	1	0	0	0	3
prepToRelation	1	3	0	3	0	0	0	7
removeRedundantStems	56	59	37	73	28	17	40	310
removeTextWhenIsa	6	12	4	13	4	2	5	46
removeTextWhenMadeOf	0	0	0	0	0	0	1	1
Total	98	101	77	113	54	33	69	545

# Timings

Average running time per message

Over the 7 messages of the Bomber Buster data set

On a Core i7 2600K @ 3.4ghz, with 16GB RAM

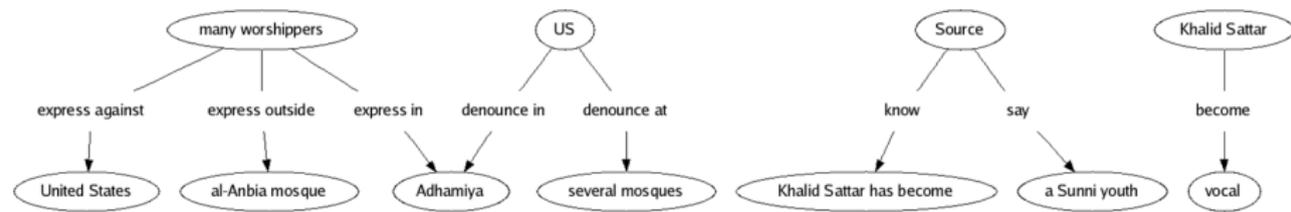
GATE (without human coreference editing):	2.642 secs
xml2sneps3:	0.978 secs
Propositionalizer (without CBIR):	0.596 secs
Propositionalizer (with CBIR):	74.547 secs

(Significant start-up time, so steady-state is faster.)

## Motivational Example: STEF Messages

- 01/05/07 - Increased hostile sentiment being expressed against U.S. troops by many worshippers outside the al-Anbia mosque in Adhamiya.
- 01/06/07 - Source said a Sunni youth he knows, Khalid Sattar, has become increasingly vocal in denouncing the U.S. at several mosques in Adhamiya.

# Motivational Example: STEF Graphs



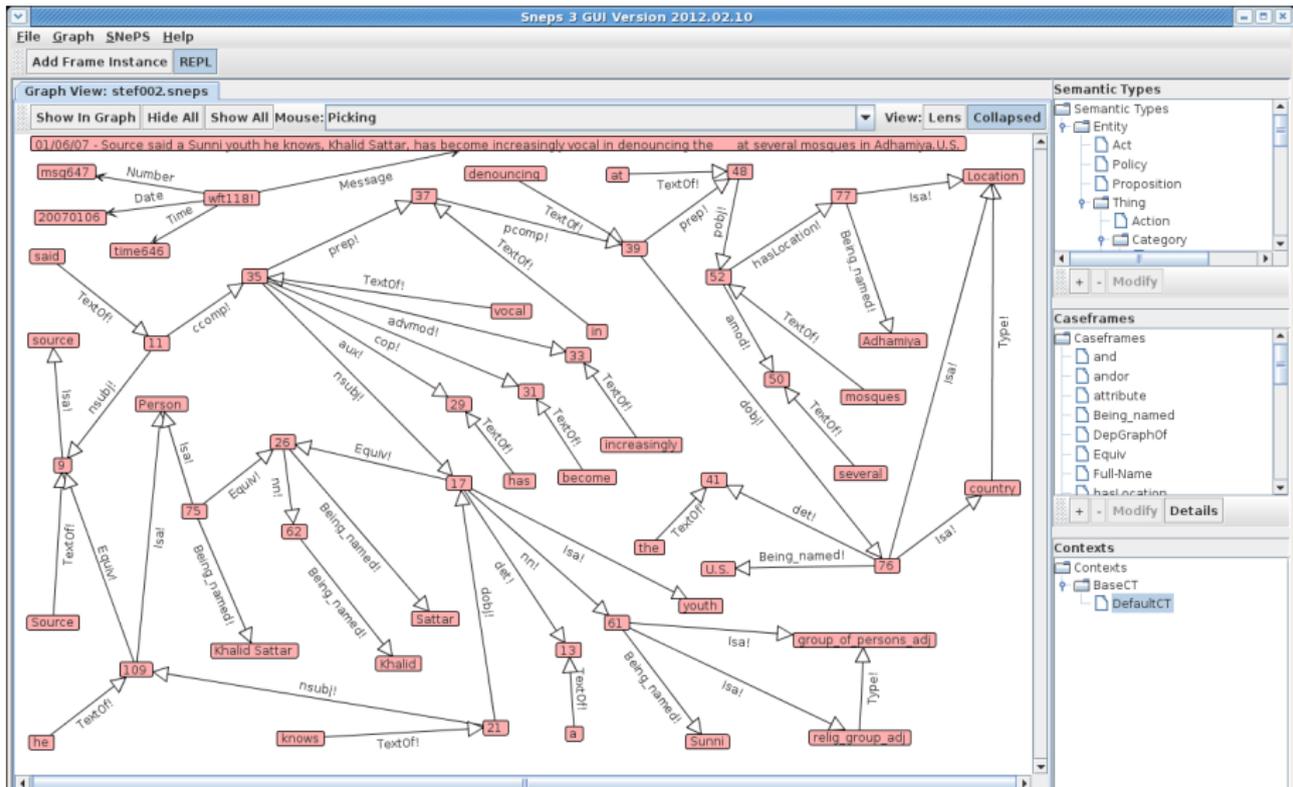
[K. Sambhoos, J. Llinas, & E. Little, Graphical Methods for Real-Time Fusion and Estimation with Soft Message Data, 11th International Conference on Information Fusion, 2008.]

## Questions:

- Relation between "US" and "United States"?
- What was expressed?
- Who denounced whom?
- Does Source know Khalid Sattar?
- Is Khalid Sattar a Sunni youth?
- What did Source say?
- When did these events occur?

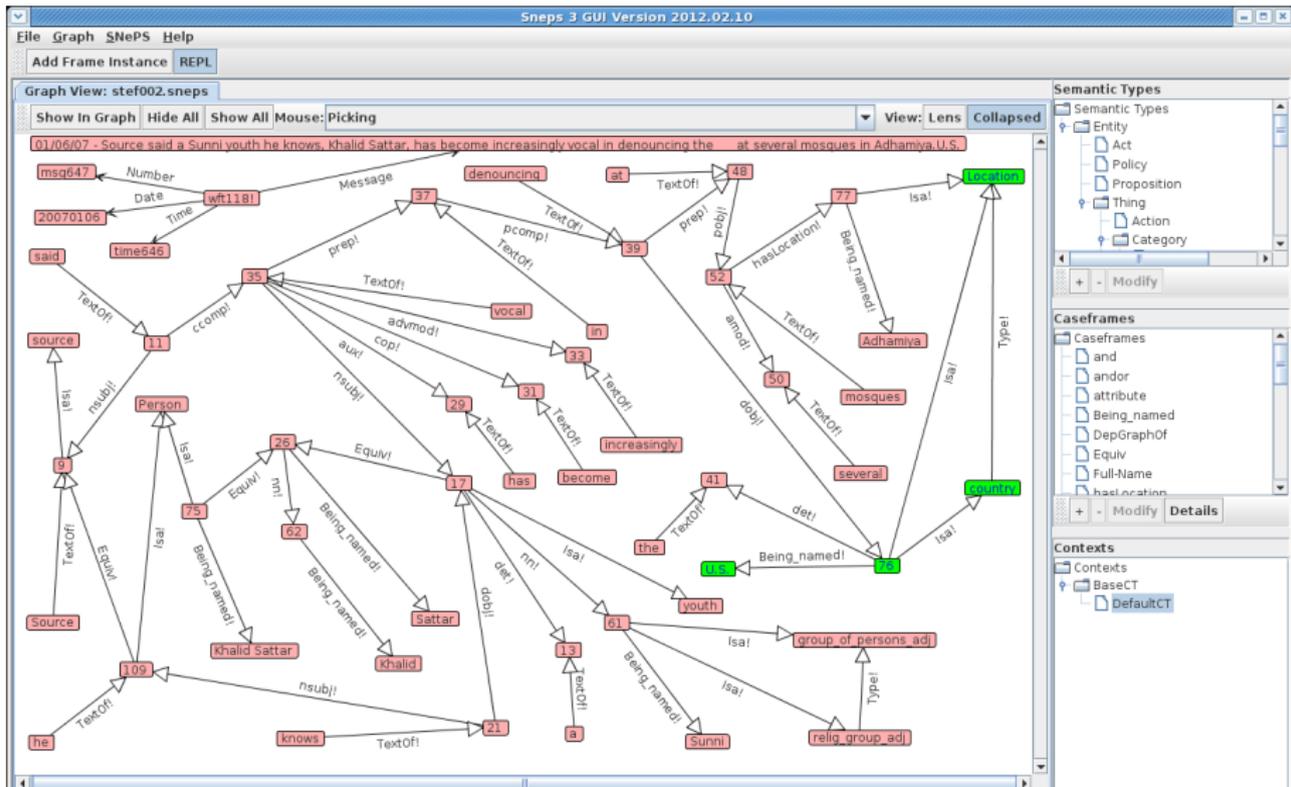


# Tractor on STEF002

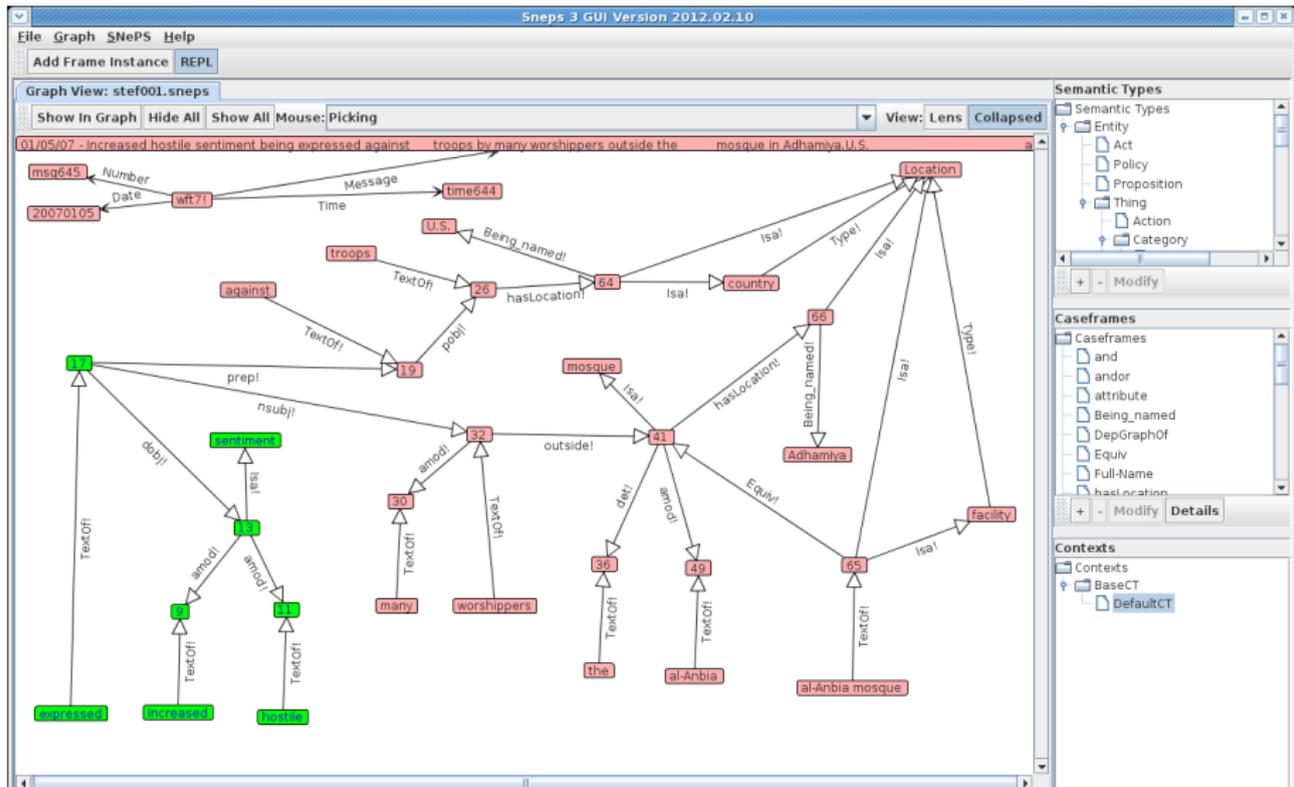




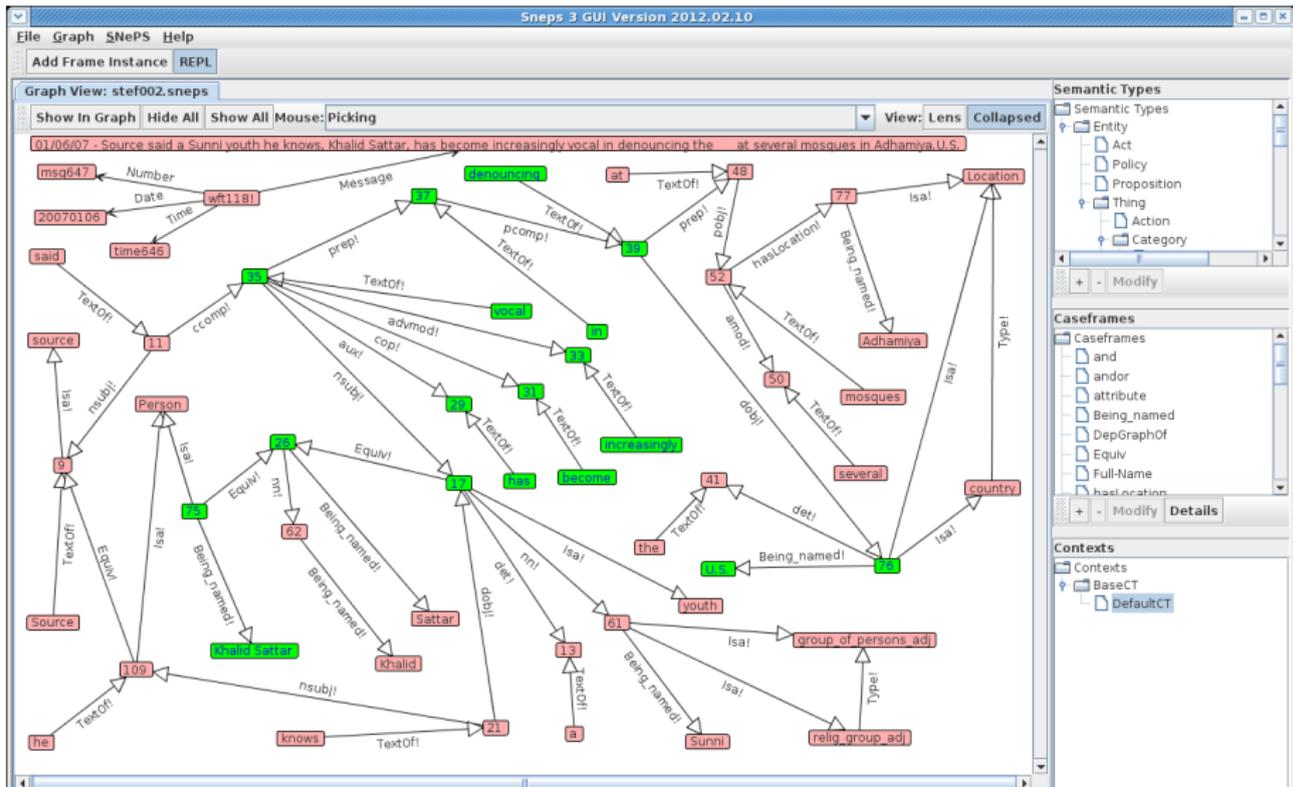
# STEF002: "US"



# STEF001: What was Expressed?



# STEF002: Who denounced whom?

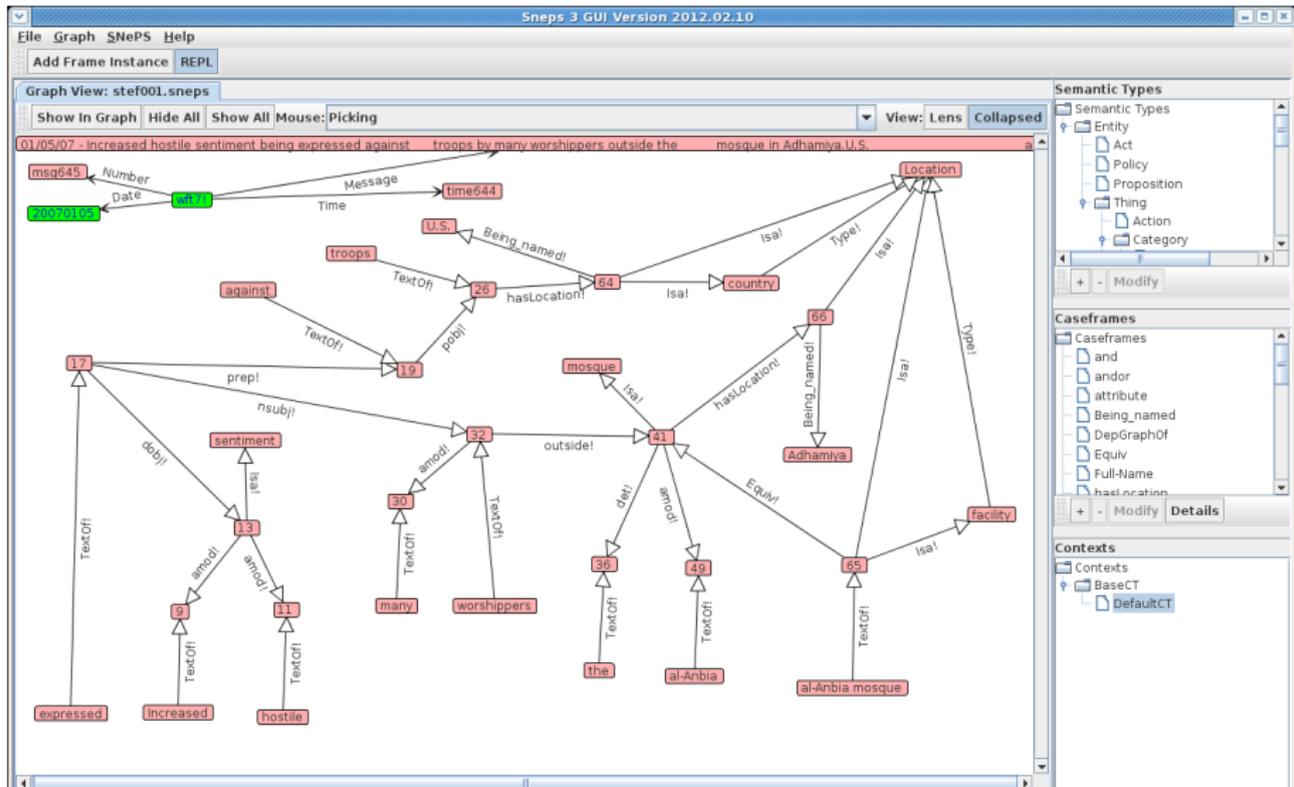








# STEF001: When did these events occur?





# Summary

- **Short English message, not necessarily “grammatical”.**
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

# Summary

- Short English message, not necessarily “grammatical”.
- Syntactic processing within GATE.
- Named Entity Recognition, List-Based and Rule-Based.
- Automatic Coreferencers.
- Human Coreference Editing.
- Dependency Parser.
- Syntactic Propositional Graph: Syntactic Relations.
- Syntax-Semantics Mapping Rules.
- Semantic Propositional Graph reflects information in message.

## Further Reading

- Michael Prentice, Michael Kandefer, & Stuart C. Shapiro, Tractor: A Framework for Soft Information Fusion, *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, 2010, Th3.2.2, 8 pages, unpaginated.
- Michael Prentice and Stuart C. Shapiro, Using Propositional Graphs for Soft Information Fusion, *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, 2011, 522–528.
- Daniel R. Schlegel and Stuart C. Shapiro, Visually Interacting with a Knowledge Base Using Frames, Logic, and Propositional Graphs. In M. Croitoru, S. Rudolph, N. Wilson, J. Howse, and O. Corby, Eds., *Graph Structures for Knowledge Representation and Reasoning, Lecture Notes in Artificial Intelligence 7205*, Springer-Verlag, Berlin, 2012, in press.