Using Propositional Graphs for Soft Information Fusion

Michael Prentice and Stuart C. Shapiro
Department of Computer Science and Engineering
and Center for Multisource Information Fusion
State University of New York at Buffalo
201 Bell Hall, Buffalo, NY 14260
{mjp44,shapiro}@buffalo.edu

***Abstract*—Soft information is information contained in natural language messages written by human informants or human intelligence gatherers. Tractor is a system that automatically processes natural language messages and represents the information extracted from them as propositional graphs. Propositional graphs have several benefits as a knowledge representation formalism for information fusion: n-ary relations may be represented as simply as binary relations; meta-information and pedigree may be represented in the same format as object-level information; they are amenable to graph matching techniques and fusion with information from other sources; they may be used by reasoning systems to draw inferences from explicitly conveyed information and relevant background information. The propositional graphs produced by Tractor are based on the FrameNet system of deep lexical semantics.**

Keywords: hard/soft data fusion, ontologies, propositional graphs.

I. Introduction

Soft information fusion is the process of associating and combining data and information from multiple soft data sensors, such as humans in the field providing data in natural language form. Soft information processing involves a number of challenges to information fusion [1], [2]. A system that accepts and uses natural language information must solve the problem of natural language understanding, in which it must represent the meaning of a message in a formalism more amenable to automated processing than the text input.

Tractor is an architecture for soft information fusion in development at the University at Buffalo as part of ongoing research into using information fusion for counter-insurgency (COIN) [3]. Tractor is composed of a syntactic processing stage, in which it extracts syntactic information such as named entities and word/phrase dependencies; a propositionalizer stage, in which it produces a propositional graph; and a context-based information retrieval stage, in which new information is incorporated into a world state estimate. This paper focuses on the propositionalizer stage.

First semantic frames as a formalism for meaning are introduced. Next, propositional graphs are demonstrated and discussed. The pipeline for the propositionalizer is shown and discussed. Finally, the use of propositional graphs for soft information fusion is discussed. Some benefits of using propositional graphs are demonstrated including how they can be used to help solve current problems in soft information fusion, and unsolved problems in their use are presented.

II. Caseframes for Semantics

In “The Case for Case,” Charles Fillmore argued for a frame-based approach to the semantics of a sentence [4]. A frame is a schematic representation of a situation with a set of participants and conceptual roles. The Berkeley FrameNet Project [5] is an implementation of the frame-based approach, providing a large database of lexical entries and their associated semantic frames.

Every frame has an associated set of frame elements, which are the semantic roles to be filled by entities of certain types. Frame elements are further divided into core and non-core, corresponding roughly to how central they are to the semantics of the frame. A core frame element is a “conceptually necessary component of a frame,” but a non-core, or peripheral, frame element does not “introduce additional, independent or distinct events” [6].

As an example, the *Communication* frame represents the act of a communicator conveying a message. The core frame elements are a *Communicator* of semantic type *Sentient*, a *Message* of semantic type *Message*, a *Medium*, and a *Topic* (with no associated semantic types in the FrameNet database). Non-core frame elements include an *Addressee* (*Sentient*), *Manner*, *Place*, *Time*, and so on. A frame can be thought of as a structure with named slots for core elements. Participants are fillers of those slots.

Frame semantics are compositional. The filler of a frame element can be another frame. The semantics of the frame is composed from the semantics of its frame elements and their relation to each other in the frame. To continue with the *Communication* frame example, suppose that the filler of *Message* is the frame *Commerce buy*, which has the frame elements *Buyer* and *Goods* and represents the act of a *Buyer* purchasing *Goods*. The semantics of such a *Communication* are the semantics of the *Commerce buy* frame in the *Message* frame element composed with the information that the *Communicator* conveyed that *Message*. To re-phrase, the *Communicator* conveyed the *Message* that the *Buyer* purchased *Goods*.

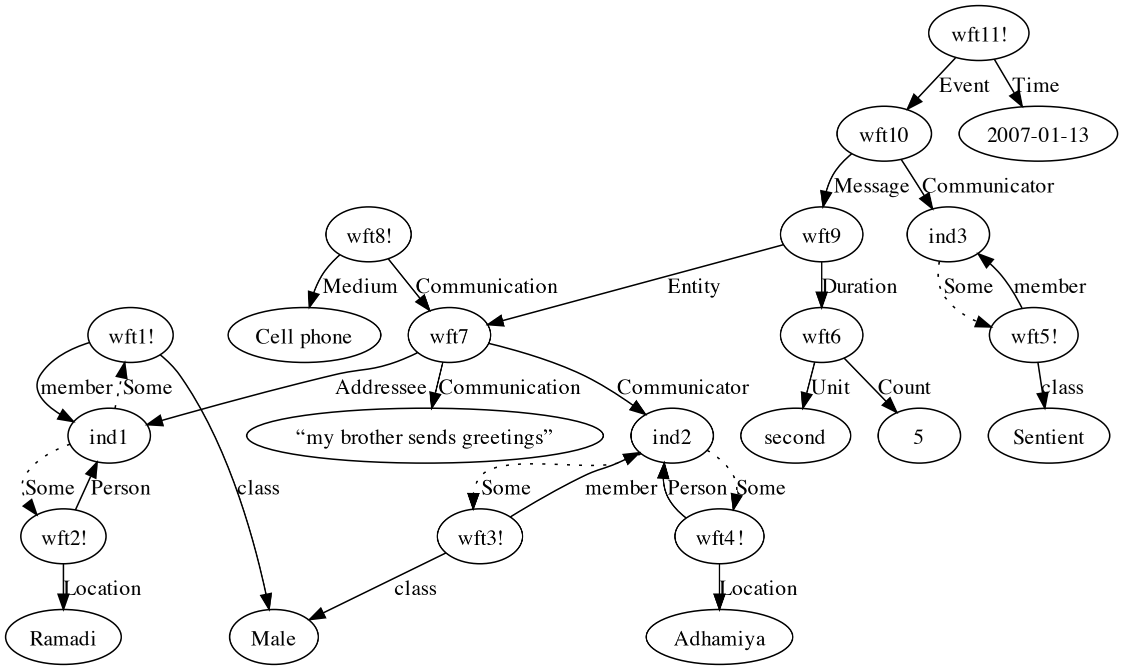

Figure 1. Example Propositional Graph

SNePS is a logic-, frame-, and graph-based knowledge representation and reasoning system developed at the University at Buffalo [7], [8]. This paper uses FrameNet frames as the semantic caseframes augmented by standard SNePS frames. In SNePS, FrameNet frames are SNePS caseframes, FrameNet core frame elements are slots of the caseframe, FrameNet non-core frame elements are propositional assertions made about the caseframe, and slot fillers have SNePS semantic types. Each term in the SNePS logic has a semantic type. Like FrameNet frames, SNePS caseframes are compositional: the slot filler of a caseframe can be another caseframe.

III. Propositional Graphs

A propositional graph is a directed graph that represents the semantics, or meaning, of a natural language input or set of inputs. A SNePS knowledge base can be viewed as a propositional graph of nodes and labeled directed arcs.

For example, Figure 1 is a visualization of the propositional graph in SNePS representing the contents of the message:

> 01/13/07 — Cell phone call from unidentified male in Adhamiya to unidentified male in Ramadi lasted just five seconds with the words "my brother sends greetings" spoken by originator of call.

This message is from the Soft Target Exploitation and Fusion (STEF) research project [9], which resulted in a message repository that contains natural language inputs from a manufactured counterinsurgency scenario. A STEF message is a timestamped, short English paragraph. STEF messages are assumed to come from a counterinsurgency operation and contain information about activities of interest.

A node in the graph is a well-formed term of SNePS. Atomic nodes contain only a unique identifier; "Adhamiya," for example.[1] A molecular node is a node with outgoing edges.

Aside from Some arcs, a propositional graph is a directed acyclic graph (DAG). Every molecular node is the root of a directed acyclic subgraph. This is important for maintaining compositionality.

Nodes have compositional semantics: the meaning of a molecular node is the meaning of the subgraph rooted at the node. wft9 is a *Duration relation* with *Duration* and *Entity* slots. wft6 is a *Measure duration* with *Count* and *Unit* slots. The meaning of wft6 is "five seconds," and the meaning of wft9 is that the contact denoted by wft7 had a duration of five seconds. wft8 is a *MediumOf* term relating a *Medium* and a *Communication*. The meaning of wft8 is that the contact wft7 was conducted over a cell phone.

An edge in the graph is directed and named. It is a slot of the caseframe of the originating node. A proposition is a molecular node. A "!" in the node label indicates a proposition that is asserted in the knowledge base (KB). For example, wft11 is an asserted proposition of caseframe *TimeOf*, with typed slots named *Time* and *Event*. The slot filler of the *Time* edge must be of semantic type *Time*, and the slot filler of the *Event* node must be of semantic type *Event*.

wft10 is a well-formed term of the *Communication* caseframe, with typed slots *Communicator* and *Message*. A *Communication* is of semantic type *Event*. The slot filler of a *Communicator* edge must be of semantic type *Sentient*.

ind nodes indicate indefinite individuals. These are terms about which we can make assertions and inferences without knowing the specific object referred to [11]. A dotted arc

[1]The issues involved in representing the names of objects and the distinction of the name from the object itself is beyond the scope of this paper. See Rapaport [10] for further discussion.

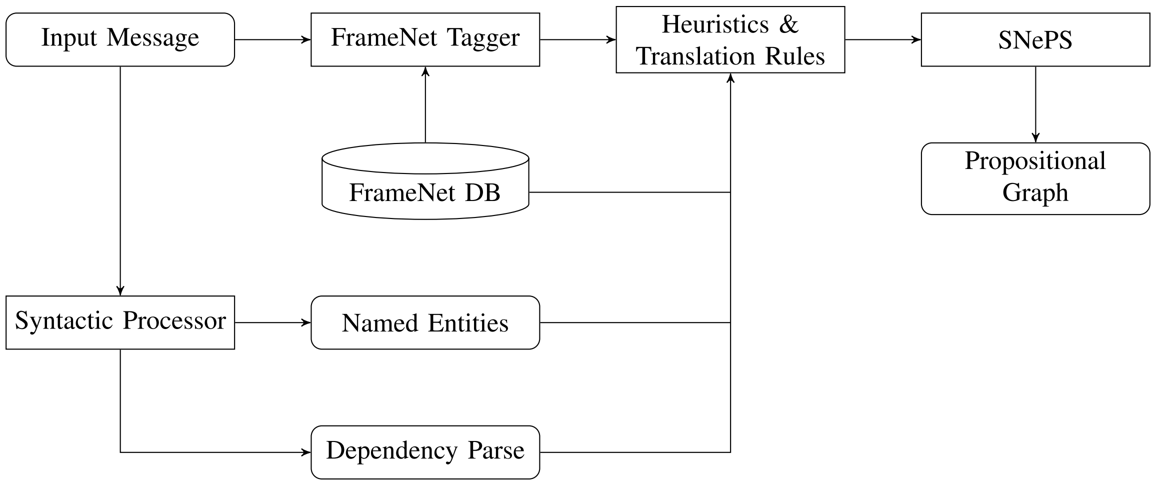

Figure 2. Propositionalizer — Detail

labeled "Some" is a restriction on the indefinite object. For example, we know that the indefinite object "ind3", which is the *Communicator* of the *Communication* term wft10, must be a *member* of *class Sentient*, because only a *Sentient* can fill the *Communicator* slot.

We also know that the indefinite object "ind2," which is the *Communicator* of a *Contacting* term (wft7), is a *member* of *class Male* and has an expected location of Adhamiya. Similar restrictions hold for the indefinite object "ind1." The relation between an indefinite object and its restriction is the only cycle in the graph.

The propositional graph structure easily allows us to represent binary and n-ary relations. *TimeOf* is a binary relation between a *Time* and an *Event*. wft7 is a *Contacting* term relating a *communicator*, an *Addressee*, and a *Communication*. wft1, wft3, and wft5 are *member-class* binary relations. wft2 and wft4 are locative binary relations between a *Person* and a *Location*.

Meta-information is a term about the term of interest. In our example, wft10 represents the *Communication* that took place. wft11 represents the proposition that the *Communication* took place at *Time* 2007-01-13. Pedigree information is meta-information that establishes the chain of custody of the information. In this case, the only pedigree information we have is that the *Communication* came from some *Sentient Communicator*, but more sophisticated pedigrees are possible.

There is information missing from this propositional graph. The graph represents only the information contained in the input message. For example, a human male is a subtype of *Sentient*, but that sort of background information is not shown in this graph. Nor does the graph contain contextual information about the current state estimate.

Background and contextual information is represented as a set of assertions in the knowledge base, which connects to message nodes through reasoning rules and the nodes' unique identifiers. SNePS uses this graph structure with a set of rules to implement reasoning.

IV. Propositionalizer

In order to produce a propositional graph from a natural language input, we need to convert from the syntactic information contained in the sentence, such as word dependencies and properly identified named entities, to the sentence's semantic payload. Figure 2 shows the design of the propositionalizer in Tractor.

Tractor uses the Stanford dependency parser [12]. Named entity recognition and dependency parsing are part of the syntactic processing in Tractor. In the example, Adhamiya and Ramadi are recognized as named entities of type location. Figure 3 shows a dependency parse for the example. The dependency parse shows typed syntactic dependencies, such as "nsubj" for nominal subject and "dobj" for direct object. The label "dep" is used for generic dependencies, which occur when the dependency parser recognizes a syntactic dependency but not its type.

In the dependency parsing view, sentences are built up from clauses and phrases and their relationships to their head words. A dependency parse is a DAG where node labels are heads of phrases and edge labels are typed dependencies. A node in the graph represents the clause or phrase rooted at the subgraph. For example, the node labeled "seconds" represents the phrase "just five seconds." The node labeled "just" is an adverbial modifier (advmod) of the phrase "five seconds." The node labeled "five" is a numerical modifier of the noun "seconds." The node labeled "lasted" is the head of the sentence and the root of the graph.

The dependency parse is a representation of the surface structure of the sentence. The meaning of a dependency parse is the syntactic relations between the words and phrases of that sentence.

In contrast, a propositional graph is a DAG (except "Some" arcs, discussed earlier) where nodes are well-formed terms in the SNePS logic. The meaning of a propositional graph is the meaning of the input text it represents. The similarity in structures between a dependency parse and a propositional

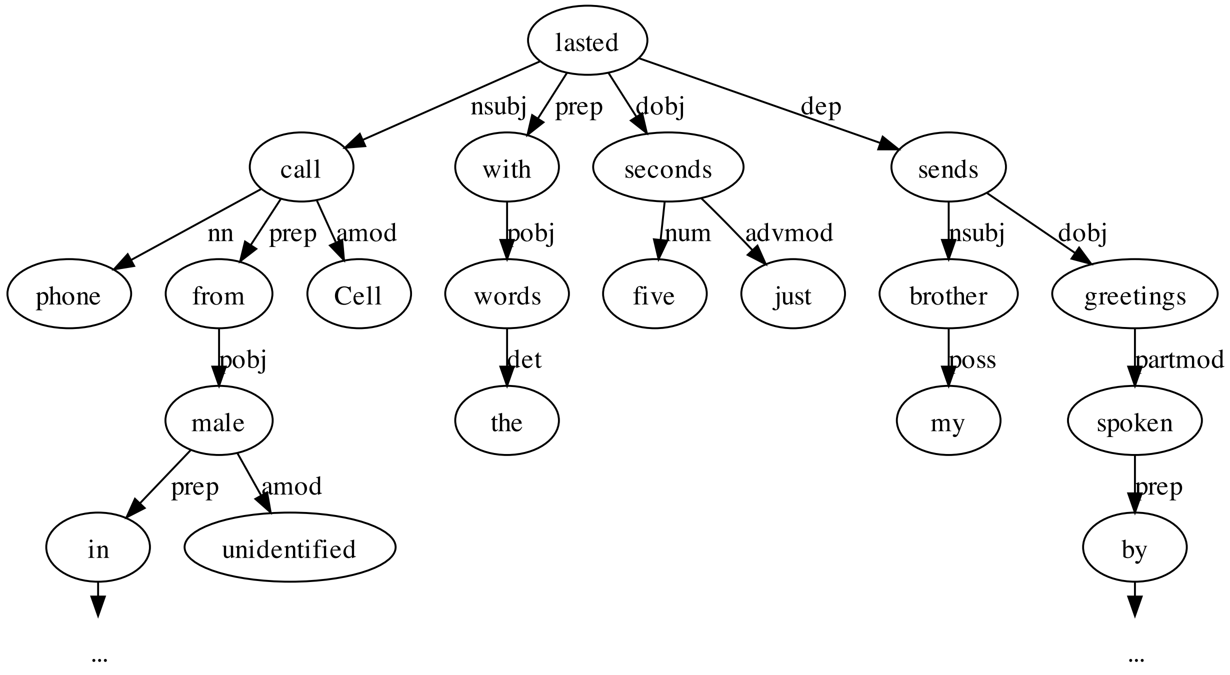

Figure 3. Example Dependency Parse (Partial)

graph allows a mapping from parse to propositional graph that is grounded in the theories of Fillmore and Beth Levin, who posit that syntactic similarities mirror underlying semantic similarities [4], [13].

The FrameNet database contains a many-to-many mapping of lexical entries to semantic frames. The FrameNet tagger takes a word stem from the input message and that word's part of speech tag, both obtained during syntactic processing, and tags the word with its associated FrameNet frame (or possibly set of frames). For example, the stem of "lasted" is "last." Last used as a verb is mapped to the *Duration relation* frame in FrameNet. The entity whose duration is being measured is a cell phone call. Call used as a noun is mapped to the *Contacting* frame.

Heuristics and translation rules gather the syntactic tag, dependency parse, and semantic tag information and produce a set of assertions in SNePS. These assertions fill the slots of SNePS caseframes. Multi-word named entities that occupy multiple nodes in the dependency graph merge into one entity node in the propositional graph.

The propositional graph *Measure duration* term wft6 with its *Count* and *Unit* slots filled in with their canonical forms is built from the dependency parse subtree rooted at the word "seconds" as follows. The word "seconds" maps to the FrameNet *Measure duration* frame. *Measure duration* requires a *Count* and a *Unit*. The surface structure of a *Measure duration* is a noun phrase whose head is a temporal unit, in this case the word "seconds," with a "num" dependent constituting the count. The *Unit* of the propositional graph is the canonical unit "second," derived from the word "second" from the dependency graph. The word "five," which is the "num" dependent in the phrase "just five seconds," maps to the canonical numeral 5, which is the slot filler for *Count*. The word "just" does not contribute to a *Measure duration* caseframe and is ignored.

The head of a clause is a verb. The verb "lasted," supplemented by its FrameNet *Duration relation* tag, maps to a term of the *Duration relation* caseframe in SNePS, shown as wft9 in the propositional graph. In an active voice sentence like the example, the *Entity* of the *Duration relation* is found as the noun subject (nsubj) of the clause, the *Duration* is the direct object (dobj). The *Duration* can be filled by a term of the *Measure duration* caseframe, in this case wft6.

The dependency parser parses quoted text, "my brother sends greetings" in the example. In this example the quoted text is a coded phrase and not meaningful without knowledge of the code. A challenge for developing rules for the propositionalizer is dealing with cases like this. The ideal case is to produce a propositional graph as shown in Figure 1, in which the coded phrase is simply a quoted string filling the *Communication* slot of the *Contacting* caseframe wft7.

The dependency parse also favors deep attachment of phrases, which can be problematic. For example, the "from" and "to" phrases should both be prepositional modifiers of "call," which would yield the *Communicator* and *Addressee* of the *Contacting* caseframe, respectively. Instead, the parser misparses the "to" phrase as a prepositional modifier of "Adhamiya," yielding the nonsensical phrase "Adhamiya to unidentified male in Ramadi." A significant research challenge remains in mapping mis-parsed sentences to the appropriate propositional graph.

Using sets of mappings like those detailed above, the

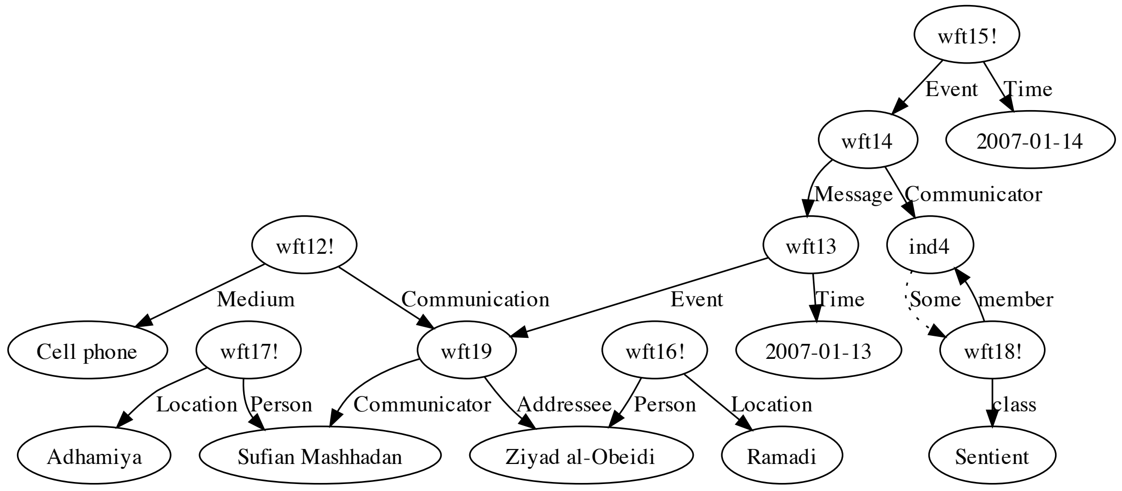

Figure 4. Partial Propositional Graph for Additional Information

propositional graph terms can be recursively built from the dependency parse subgraphs.

SNePS integrates the assertions into a global propositional graph that contains the assertions in context with other inputs and with background knowledge [14], [15]. The propositional graph is ready for forward inferencing with background knowledge and context.

V. Soft Information Fusion

By providing a structure for meaning, propositional graphs have great power and flexibility for merging new information as it becomes available. Figure 4 shows the propositional graph for an additional message from STEF:

> 01/14/07 — Originator of 1/13/07 cell phone call to Ramadi from Adhamiya has now been identified as Sufian Mashhadan. No other information about him. The recipient has been identified as Ziyad al-Obeidi, a suspected high-ranking member of al-Qaeda in Iraq.

This message contains information about the information in another message. Specifically, it identifies the previously unknown communicator and addressee of a cell phone contacting action. Previously it was known only that the indefinite individual ind2 was male and in Adhamiya. ind2 should now be identified in the graph as the individual named Sufian Mashhadan. Similarly, ind1 should be identified as Ziyad al-Obeidi. This poses problems for linguistic inference and graph matching.

wft7 of Figure 1 and wft19 of Figure 4 represent the same cell phone call. However, in Figure 1, the communication (wft10) takes place on 2007-01-13 (wft11!), but the time of the cell phone call is unspecified. An inference step is required to assert that in the case of an unspecified time of contacting, the contact referred to in the message takes place on the date of the message's communication.

Background and contextual information can be used to augment the propositional graph for further reasoning. For example, if it is known from a domain ontology or from previous messages that Sufian and Ziyad are males, those should be asserted. However, there is no such information contained in the message.

Figure 5 shows the two messages merged into the same graph. At this point, forward inferencing has yet to be done. Once forward inferencing identifies the time of the cell phone contact represented by wft7, the graph structures look similar and graph matching can be performed. Although the information required to identify ind1 with Ziyad and ind2 with Sufian is incomplete, the similarities in graph structure between ind1 and Ziyad, ind2 and Sufian, and wft7 with wft19 should allow identity assertions and node merges [16].

VI. Conclusions and Future Work

Propositional graphs are a knowledge representation formalism for natural language input that have strong advantages for incorporation into an information fusion framework, including the ability to represent n-ary relations as easily as binary relations, to store pedigree and meta-information in the same format as message data, to incorporate new information into background knowledge sources and prior inputs, to draw inferences, and to merging using graph matching techniques. The propositionalizer component in Tractor takes as input syntactically processed natural language text and produces propositional graphs representing the meaning contained in the input. FrameNet semantic frames and the SNePS knowledge representation and reasoning system are used by Tractor to produce the graphs.

The tools used by the propositionalizer are currently in development. Current work has created ground truth propositional graphs for comparison using FrameNet frames and the

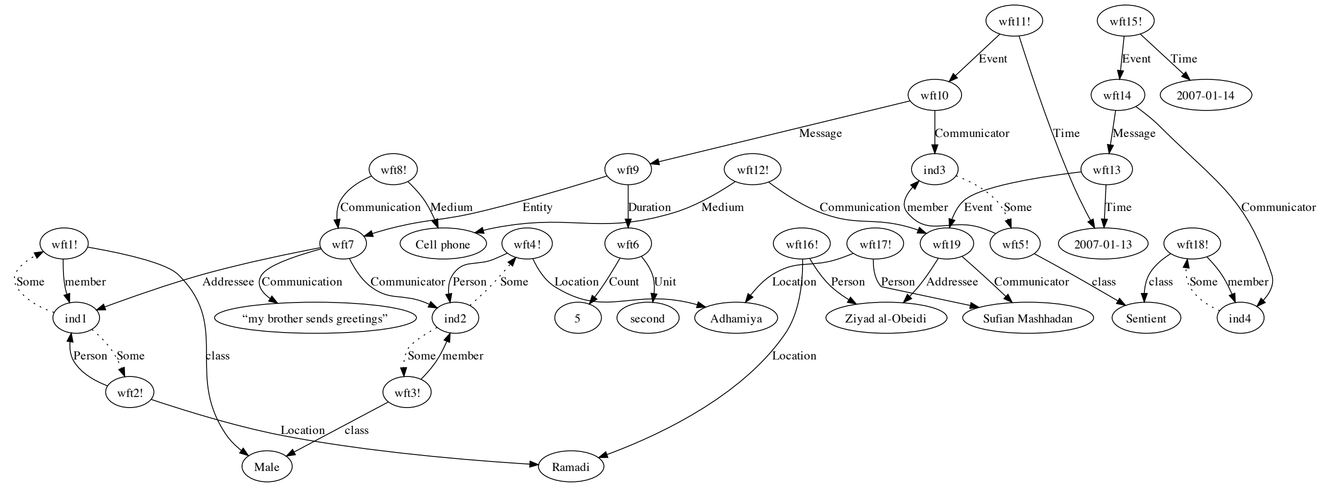

Figure 5. Complete Propositional Graph

SNePS system. The graphs used in this paper have been hand-built using the mappings detailed in section IV. Automating this process to produce propositional graphs such as these is the major implementation focus of future work on the Propositionalizer. Future work also includes creating measures of conformance to ground truth graphs and tools to measure that conformance.

ACKNOWLEDGEMENTS

We gratefully acknowledge that this research activity has been supported by a 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.

REFERENCES

[1] D. L. Hall, J. Llinas, M. McNeese, and T. Mullen, "A framework for dynamic hard/soft fusion," in *Proc. 11th Int. Conf. on Information Fusion*, Cologne, Germany, 2008.

[2] M. A. Pravia, R. K. Prasanth, P. O. Arambel, C. Sidner, and C.-Y. Chong, "Generation of a fundamental data set for hard/soft information fusion," in *Proceedings of the 11th International Conference on Information Fusion*, June 2008, pp. 1–8.

[3] M. Prentice, M. Kandefer, and S. C. Shapiro, "Tractor: An architecture for soft information fusion," in *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

[4] C. J. Fillmore, "The case for case," in *Universals in Linguistic Theory*, E. Bach and R. Harms, Eds. New York: Holt, Rinehart & Winston, 1968.

[5] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *Proceedings of the 17th international conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1998, pp. 86–90.

[6] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk, "FrameNet II: Extended theory and practice," *Unpublished Manuscript*, 2006. [Online]. Available: http://framenet.icsi.berkeley.edu/

[7] S. C. Shapiro, "An introduction to SNePS 3," in *Conceptual Structures: Logical, Linguistic, and Computational Issues. Lecture Notes in Artificial Intelligence 1867*, B. Ganter and G. W. Mineau, Eds. Berlin: Springer-Verlag, 2000, pp. 510–524.

[8] S. C. Shapiro and W. J. Rapaport, "The SNePS family," *Computers & Mathematics with Applications*, vol. 23, no. 2–5, pp. 243–275, January–March 1992, reprinted in F. Lehmann, Ed. *Semantic Networks in Artificial Intelligence*. Pergamon Press, Oxford, 1992, 243–275.

[9] K. Sambhoos, J. Llinas, and E. Little, "Graphical methods for real-time fusion and estimation with soft message data," in *Proceedings of the 11th International Conference on Information Fusion*, July 2008, pp. 1–8.

[10] W. J. Rapaport, "Logical foundations for belief representation," *Cognitive Science*, vol. 10, pp. 371–422, 1986.

[11] S. C. Shapiro, "A logic of arbitrary and indefinite objects," in *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, D. Dubois, C. Welty, and M. Williams, Eds. Menlo Park, CA: AAAI Press, 2004, pp. 565–575.

[12] M.-C. de Marneffe and C. D. Manning, "The Stanford typed dependencies representation," in *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*, 2008. [Online]. Available: http://nlp.stanford.edu/pubs/dependencies-coling08.pdf

[13] B. Levin, *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago: The University of Chicago Press, 1993.

[14] M. Kandefer and S. C. Shapiro, "A categorization of contextual constraints," in *Biologically Inspired Cognitive Architectures, Technical Report FS–08–04*, A. Samsonovich, Ed. AAAI Press, Menlo Park, CA, 2008, pp. 88–93.

[15] ——, "An F-measure for context-based information retrieval," in *Commonsense 2009: Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning*, G. Lakemeyer, L. Morgenstern, and M.-A. Williams, Eds. Toronto, CA: The Fields Institute, Toronto, CA, 2009, pp. 79–84.

[16] K. Sambhoos, R. Nagi, M. Sudit, and A. Stotz, "Enhancements to high level data fusion using graph matching and state space search," *Information Fusion*, vol. 11, no. 4, pp. 351 – 364, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B6W76-4XY4N25-1/2/3f1e5a92822dee799f290c60569bbda9