Introduction to WIMs and wimformat 1.0

Jesse English – jesse@jesseenglish.com

WIM [noun, singular]: "weakly inferred meaning"; a meaning representation of unstructured text derived from light semantic analysis that covers a limited, but important, set of relations: *We produced a set of WIMs from the text.*

Why use WIMs?

A WIM is a structured meaning representation, not unlike a TMR (text meaning representation), with a limited scope in expected coverage. The scope has been limited intentionally for performance reasons – one would use a WIM rather than a TMR when the scope of coverage is sufficient and the cost (in time or development) of a full TMR is too great.

Typically, the production of a full TMR would require a domain-comprehensive syntactic-semantic lexicon and accompanying ontology (as well as a wealth of other related knowledge bases). A compilation of microtheories of meaning analysis would be required to process the text using the knowledge – both of these resources are extremely expensive to produce, and accurate processing of the text rapidly becomes unscalable without introducing domain-dependent algorithmic shortcuts.

By relying on WIMs, rather than a full TMR, the most typically relevant semantic data can still be produced in linear time with off-the-shelf knowledge resources (e.g., WordNet).

The following sections describe the valid formats and usages of **wimformat 1.0**

What does a WIM look like?

A WIM looks very similar to a TMR, and can nearly be interchanged (with a few caveats). In general, a WIM can be described as a series of frames, each representing an instance of knowledge found in the input text with a complement of interconnecting relations and descriptive attributes.

- WIM frames do not use concept generalization – the frames are derived from unmapped text
- WIM frames are generated from each verb head in a text, and all interconnecting nouns to those verbs
- WIM frames may include any number of relations (which are a tuple: the relation name, and the connected WIM frame)
- WIM frames may include any number of attributes (which are a tuple: the attribute name, and a literal value)
- All properties (relations and attributes) are optional in any given WIM frame – it is valid to have an empty WIM frame
- All relations are non-unique in any given WIM frame – e.g., you may have multiple AGENT relations in a single WIM frame
- All attributes are unique in any given WIM frame – e.g., you may not have more than one PLURALITY relation in a single WIM frame

An illustrative example of the WIM produced by a simple sentence is below:

“The girl bought a sandwich.”

girl-1

- fromtext: “the girl”

bought-1

- fromtext: “bought”
- AGENT: girl-1
- THEME: sandwich-1

sandwich-1

- fromtext: “the sandwich”

What relations are defined for WIM frames?

The following is a complete list of the relations defined for WIM frames. Following each definition is an example sentence. The WIM frame's head is bolded, and the relation being illustrated is surrounded by brackets.

AGENT – who or what is taking or doing an action
“[The girl] **bought** a sandwich.”

THEME – who or what an action is targeted to or on
“The girl **bought** [a sandwich].”

LOCATION – where an action is taking place
“I **bought** a sandwich at [Lucky's Pub].”

INSTRUMENT – what is used to take an action
“The man **hit** the build with [the hammer].”

BENEFICIARY – who or what is being affected by an action
“I **told** [Jim] to buy a new car.”

SCOPE – the purpose, breadth, destination or additional meaning-driven modifier to the action
“She **took** her kids [to the beach].”
“She **took** her kids [on vacation].”
“She **bought** a book [so she could learn Spanish].”
“She **goes** to Montana [once a year].”

What attributes are defined for WIM frames?

The following is a comprehensive list of the relations defined for WIM frames. Each attribute has its valid set of values defined as well, along with an example sentence where the WIM frame's head is bolded and the text causing the attribute modification is surrounded by brackets.

PLURALITY – specifies if the frame represents a singular or plural value of the instance; valid values are “yes” (implying plurality) or “no” (implying singularity); if PLURALITY is not specifically

defined in a frame, it is inferred that the value is “no”
“I bought a few **book**[s].”

GENDER – specifies the gender of the WIM frame; valid values are “male” and “female”; if GENDER is not specifically defined in a frame, there is no inferred value
“[**She**] bought a book.”

NAME – any string that specifies a proper noun (note that the WIM frame will be something more generalized, such as “Mr. Jones” → man-1 if onomastic lookup is implemented); if NAME is not specifically defined in a frame, there is no inferred value
“[**Mr. Jones**] bought a book.”

RELQUANT – a 0 – 1 scale representing a relative quantity converted from a list of quantity modifiers, e.g., “lots” = 0.7; if RELQUANT is not specifically defined in a frame, it is inferred that the value is 1.0
“I bought [a ton] of **books**.”

ABSQUANT – any absolute number as found in the text that describes the quantity of a WIM frame element; if ABSQUANT is not specifically defined in a frame, it is inferred that the value is 1
“I bought [seven] **books**.”

TYPE – a value from an enumerated list that is context dependent, e.g., for question frames, valid values include “who”, “what”, “where”, “when”, “why”, “how”; if TYPE is not specifically defined in a frame, there is no inferred value
“[Where] did you go**?**”

SENSE – an implementation-dependent mapping to a disambiguated lexical sense (if the WIM frame's specific lexical sense is known, it can be recorded here); if the SENSE is not specifically defined in a frame, there is no inferred value

<u>How are questions handled?</u>

The format for questions involves creating a question frame that SCOPEs over the subject of the question. Additionally, the question frame will contain a TYPE. A simple example is show below:

“How do I build a desk?”

question-1

- fromtext: “?”
- TYPE: how
- SCOPE: build-1

build-1

- fromtext: “build”
- AGENT: I-1
- THEME: desk-1

I-1

- fromtext: “I”

desk-1

- fromtext: “a desk”