

# High-Fidelity Lexical Axiom Construction from Verb Glosses (Supplementary Document)

**Gene Kim**

University of Rochester  
Department of Computer Science  
gkim21@cs.rochester.edu

**Lenhart Schubert**

University of Rochester  
Department of Computer Science  
schubert@cs.rochester.edu

## 1 Suitability of EL-Based Lexical Axioms for Inference

Figure 1 illustrates a simple forward inference chain using an axiom from WordNet for the sentence “John stumbles, but does not fall”. Using the axiom for *stumble2.v*, and a hand written axiom schema asserting that statements conjoined with the connective “but” asserts the conjunction of the two statements as well. The semantics of abstract words, such as “but” need to be encoded by hand since dictionaries simply define abstract words in cycles. This second axiom is an axiom schema since it uses substitutional quantification over well-formed formulas,  $\forall_{wff}$ . Substitutional quantification is part of what allows EL to represent information about its own syntax and is used for meta-syntactic reasoning. Substitutional quantification and meta-reasoning in EL is explained in detail by Morbini & Schubert (2008). The inference process concludes that “John misses a step and nearly falls”. This is an example of an inference that representations using an intersective approach to predicate modification cannot make since “John nearly falls” and “John does not fall” would contradict each other.

Inference using only verb axioms is largely limited to paraphrasing verbs in terms of other words. Therefore, we limit ourselves to an inference example to demonstrate the inference capabilities of the axioms. Future work in axiomatizing nouns, adjectives, and adverbs will lead to much broader inference capabilities. This will allow us to demonstrate the inference in a more general setting. Nouns in particular would boost the inference process because many verbs are defined in terms of nominal forms of other verbs.

The example in Figure 1 only uses forward inference, but EL supports deductive, uncertain, and Natural Logic-like inference as well. Its suit-

ability for reasoning has been demonstrated extensively (Morbini and Schubert, 2009; Morbini and Schubert, 2011; Schubert and Hwang, 2000; Schubert et al., 2011; Schubert, 2013).

## 2 Addressing Circular Definitions

In order to use lexical axioms from WordNet for forward inference in practice, the circular definitions that exist in WordNet need to be addressed to avoid circular inferences. A couple of strategies already exist for this problem. One strategy is to select an alternative sense the word in gloss that leads to a cycle, as done by Allen et al. (2011). Another strategy is to create a set of core words that all glosses eventually converge to through lexical forward inference and avoid performing lexical inference on the core words. Mostafazadeh & Allen (2015) use Tarjan’s algorithm (Tarjan, 1972) to identify strongly connected components in the graph of words connected to words in their glosses to form their core words.

## Axioms

A1. *stumble2.v* : miss a step and fall or nearly fall

$$(\forall x, e ((x \text{ stumble2.v}) ** e) \rightarrow ((\exists z (z \text{ step2.n}) (x \text{ miss4.v } z)) \wedge ((x \text{ fall23.v}) \vee (x (\text{nearly.adv } \text{fall23.v})))) ** e)))$$

A2. If two statements are conjoined by “but”, then both statements are true (i.e. conjunction)

$$(\forall_{wff} x, y (\forall e ((x \text{ but.cc } y) ** e) \rightarrow ((x \wedge y) * e))))$$

## Inference

Sentence: “John stumbles, but does not fall”

I1. ((John stumble2.v) but.cc ¬(John fall23.v))	Initial parsed sentence
I2. ((John stumble2.v) ∧ ¬(John fall23.v))	I1 & A2
I3. (John stumble2.v), ¬(John fall23.v)	I2
I4. ((∃z (z step2.n) (John miss4.v z)) ∧ ((John fall23.v) ∨ (John (nearly.adv fall23.v))))	I3 & A1
I5. ((∃z (z step2.n) (John miss4.v z)) ∧ (John (nearly.adv fall23.v)))	I3 & I4

Figure 1: If John stumbles, but doesn’t fall, we can infer from the axioms extracted from WordNet verbs that he misses a step and nearly falls. This inference would not be possible with representations that use an intersective approach to predicate modification, such as OWL-DL.

## References

- James Allen, William de Beaumont, Nate Blaylock, George Ferguson, Jansen Orfan, and Mary Swift. 2011. Acquiring commonsense knowledge for a cognitive agent. In *Proceedings of the AAAI Fall Symposium Series: Advances in Cognitive Systems (ACS 2011)*, Arlington, VA, USA.
- Fabrizio Morbini and Lenhart Schubert. 2008. Metareasoning as an integral part of commonsense and autocognitive reasoning. In *AAAI-08 Workshop on Metareasoning*.
- Fabrizio Morbini and Lenhart K. Schubert. 2009. Evaluation of EPILOG: a reasoner for Episodic Logic. In *Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning*.
- Fabrizio Morbini and Lenhart Schubert. 2011. Metareasoning as an integral part of commonsense and autocognitive reasoning. In Michael T. Cox and Anita Raja, editors, *Metareasoning: Thinking about thinking*. MIT Press, January.
- Nasrin Mostafazadeh and James F. Allen. 2015. Learning semantically rich event inference rules using definition of verbs. In *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing Proceedings, Part I*, volume 9041 of *Lecture Notes in Computer Science*, pages 402–416, Cairo, Egypt, April. Springer.
- Lenhart K. Schubert and Chung Hee Hwang. 2000. Episodic Logic meets Little Red Riding Hood: A comprehensive natural representation for language understanding. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and Knowledge Representation*, pages 111–174. MIT Press, Cambridge, MA, USA.
- Lenhart K Schubert, Jonathan Gordon, Karl Stratos, and Adina Rubinoff. 2011. Towards adequate knowledge and natural inference). In *AAAI Fall Symposium: Advances in Cognitive Systems*.
- Lenhart Schubert. 2013. NLog-like inference and commonsense reasoning. *LiLT (Linguistic Issues in Language Technology)*, 9.
- Robert Tarjan. 1972. Depth first search and linear graph algorithms. *SIAM Journal on Computing*.