Detour

A detour to
FrameNet using WordNet

WordNet

- "English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept." (http://wordnet.princeton.edu/)
- Synonyms sets (synsets) are organized hierarchically (especially hypernyms, hyponyms, sometimes antonyms)

buy#v#1

- Description: “obtain by purchase; acquire by means of a financial transaction”
- Synonyms: buy, purchase
- Hypernyms: get#v#1, acquire#v#1
- Antonyms: sell#v#1

FrameNet

- Situational Knowledge
  - Includes Roles (Frame Elements)
  - Each Frame is annotated with Lexical Units, which “evoke” the frame

Frame “Attack”

- Description: An Assailant physically attacks a Victim (which is usually but not always sentient), causing or intending to cause the Victim physical damage. A Weapon used by the Assailant may also be mentioned, in addition to the usual Place, Time, Purpose, Reason, etc.
- Inherits: Intentionally_affect

Frame “Attack” (2)

- Frame Elements: Assailant, Victim, Circumstances, Containing_event, Depictive, Event_description, Explanation, ...
- Lexical Units: ambush.v, assault.v, attack.v, bomb.v, ...

Motivation

- Manually annotating frames is expensive and slow, therefore: FrameNet has a low coverage (FrameNet 1.2 contains only 750 frames)

Idea

- If a word is not contained in FrameNet (as Lexical Unit) - maybe one of its synonyms or hypernyms is?

Step 1

- Collect all synonyms and hypernyms one the input synset

$$SW = \{w | w \in synonyms(input) \vee w \in hypernyms(input)\}$$

- Add all the antonyms of each search word

$$SW = SW \cup \{w' | \exists w \in SW \wedge antonyms(w', w)\}$$

Step 1 - Example

- input: buy#v#1
- SW = { get, buy, acquire, purchase, sell }

Step 2

- Collect every frame that is evoked by every word in SW

$$Evoking(F) = \{w | w \in SW \wedge w \in LU(F)\}$$

$$Spreading(w) = |\{F | w \in LU(F)\}|$$

Step 2 - Example

- Evoking:
  - Getting: get, acquire
  - Arriving: get
  - Becoming: get
  - Grasp: get
  - Commerce_sell: sell
  - Commerce_buy: buy, purchase

Spreading Factors:
get: 4
acquire: 1
sell: 1
buy: 1
purchase: 1

Step 3

- Now we have a bag of candidate frames, but we need to choose among them. Therefore, we have a weighting function

$$weight(F) = \frac{1}{|Evoking(F)|} \sum_{synset \in Evoking(F)} \frac{sim(synset, input)^2}{Spreading(synset)}$$

$$sim(s1, s2) = \frac{1}{dist(s1, s2) + 1}$$

Step 3 - Example

$$weight(\text{"Getting"}) = \frac{1}{2}(\frac{0.5^2}{4} + \frac{0.5^2}{1}) = 0.156$$

$$weight(\text{"Commerce\_sell"}) = \frac{1}{1}\frac{0.167^2}{1} = 0.028$$

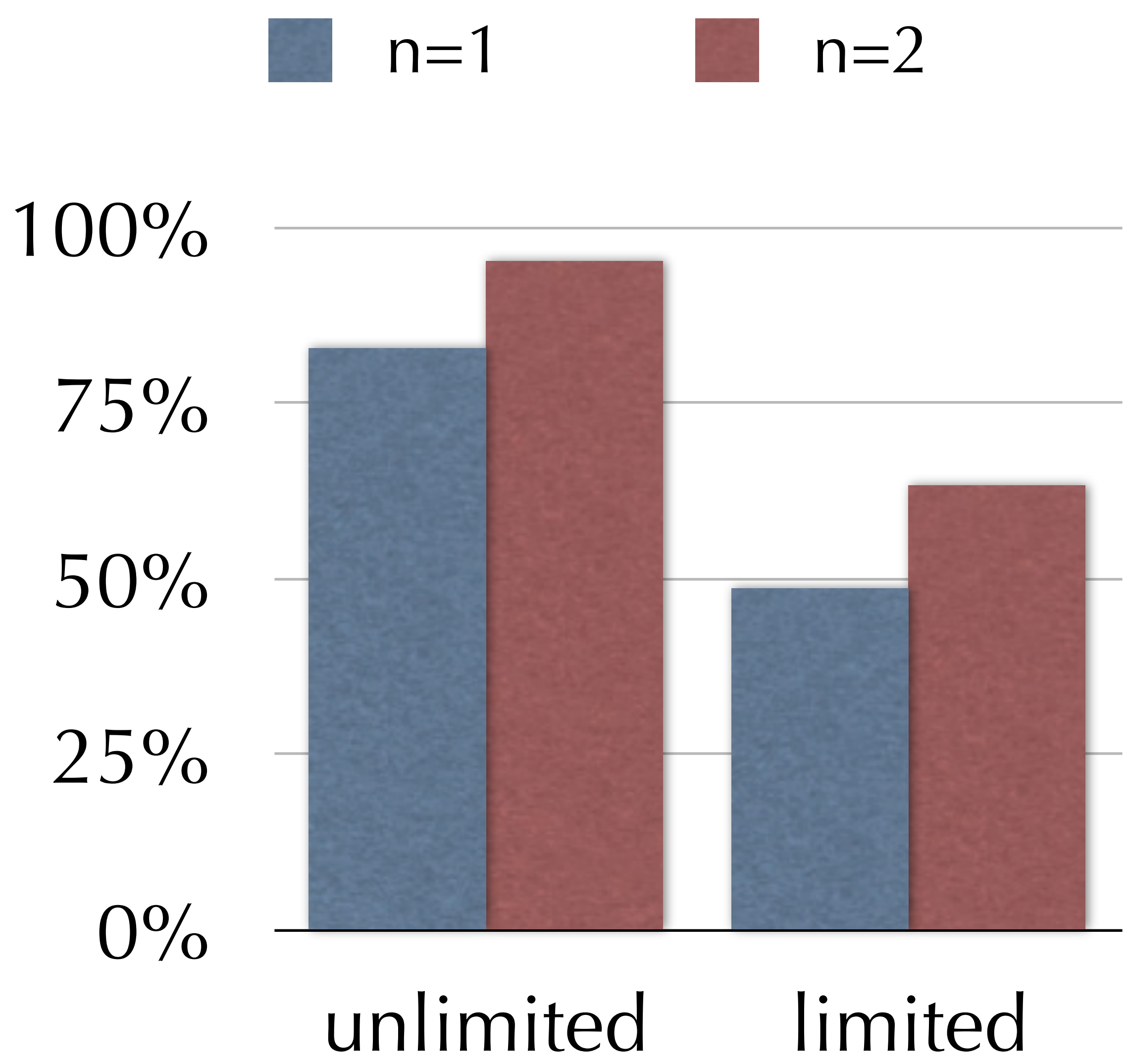
$$weight(\text{"Commerce\_buy"}) = \frac{1}{2}(\frac{1^2}{1} + \frac{1^2}{1}) = 1$$

Cheating

- Cheating: If the input synset is annotated in FrameNet, it is no challenge to find the appropriate Frame.
- Limited condition:
  Exclude the input synset
  "Behave as if the input synset would not be in FrameNet"

Evaluation

- "Gold Standard": Shi and Mihalcea, 2005
- Compared the first n results to the gold standard
- "In 50% of the cases, where we don't know anything, we can make a strong suggestion"



Technical Details

- Programming Language: Perl
- Released on CPAN as FrameNet::WordNet::Detour

Demo

Demo 1 - unlimited

```
Querying: buy#v#1 ...
Synsets considered:
buy#v#1(1)
  evokes(lu): [Commerce_buy]
get#v#1(0.5)
  evokes(lu): [Getting Arriving Becoming Grasp]
acquire#v#1(0.5)
  evokes(lu): [Getting]
sell#v#1(0.167)
  evokes(lu): [Commerce_sell]
purchase#v#1(1)
  evokes(lu): [Commerce_buy]

All Frames: Getting(0.156) Arriving(0.063) Becoming(0.063) Grasp(0.063) Commerce_sell(0.028) Commerce_buy(1)
Best result(s): Commerce_buy
Commerce_buy;
```

Demo 2 - limited

```
Querying: buy#v#1 ...
Synsets considered:
get#v#1(0.5)
  evokes(lu): [Getting Arriving Becoming Grasp]
acquire#v#1(0.5)
  evokes(lu): [Getting]
sell#v#1(0.167)
  evokes(lu): [Commerce_sell]
purchase#v#1(1)
  evokes(lu): [Commerce_buy]

All Frames: Getting(0.156) Arriving(0.063) Becoming(0.063) Grasp(0.063) Commerce_sell(0.028) Commerce_buy(1)
Best result(s): Commerce_buy
Commerce_buy;
```