

Mood Guessing

Steven Buraje Poggel
Tacos 2006

Outline

- Advanced word relations
- ConceptNet
- Testing the Mood Guessing functionality sytematically
- Possible Improvements of the Mood Guessing functionality

Advanced Word Relations

- VerbOcean (T. Chklovski, P. Pantel)
- Learner / Learner2 (T. Chklovski, Y. Gil)
- Open Mind Common Sense (P. Singh)
- ConceptNet (H. Liu, P. Singh)
- Summary and Comparison

ConceptNet

H. Lui and P. Singh

What is ConceptNet?

- Commonsense knowledge base
- Similar to Wordnet and Cyc
- Optimised for making practical context-based inferences over real world text
- Based on OMCS (Open Mind Common Sense) corpus, a corpus collected by volunteer contributors (Singh et al. 2002)

What is commonsense knowledge?

- It is defeasible world knowledge
- Often just a default assumption of what is likely
- Example: *getting fired*: people sometimes *get fired* because they are incompetent, a possible consequence is *not having money, people use money to buy food*

ConceptNet and WordNet

- WordNet is optimised for lexical categorisation and word-similarity
- WordNet is a database with words, mostly nouns, verbs and adjectives organised into discrete 'senses'
- Wordnet sense are linked by a small set of semantic relations

ConceptNet and WordNet

- ConceptNet is a simple semantic network with words at the nodes
- ConceptNet focuses on a richer set of semantic relations (EffectOf, DesireOf, CapableOf)
- ConceptNet is automatically built by extraction, not handcrafted

ConceptNet Development

- OMCS gathered 700 000 sentences of commonsense knowledge from over 14000 contributors
- OMCS alone sentences are not directly computable
- ConceptNet is the machine-computable form of the OMCS corpus

Building ConceptNet

- Automated process applies extraction rules to semi-structured sentences of OMCS
 - 50 extraction rules map from OMCS to binary relations
 - Each extracted concept node is a combination of the syntactic constructs: verbs, NPs, PPs, APs
- Relaxation procedures
 - Normalisation: Spellchecking + stripping (fin → inf, plur → sing) yields normalised 'lemma' form

Inference Example

```
[ (IsA 'apple' 'fruit');  
  (IsA 'banana' 'fruit');  
  (IsA 'peach' 'fruit') ]
```

AND

```
[ (PropertyOf 'apple' 'sweet');  
  (PropertyOf 'banana' 'sweet');  
  (PropertyOf 'peach' 'sweet') ]
```

IMPLIES

```
(PropertyOf 'fruit' 'sweet')
```

What can we do with ConceptNet?

- Find contextual neighborhoods: Given a concept, what other concepts are most relevant
- Analogy making: two ConceptNet nodes are analogue when their incoming edges overlap
- Do topic gisting: generate a topic from a given text
- Projection of transitive relations: e.g. *locationOf(SB, Saarland)*, *locationOf(Saarland, Germany)*, *locationOf(Germany, Europe)*, *locationOf(Europe, Earth)* projects to *locationOf(SB, Earth)*, *locationOf(SB, Europe)*, *locationOf(SB, Germany)*, *locationOf(Germany, Earth)*, ...
- And sure enough: Mood guessing

Contextual neighbourhood

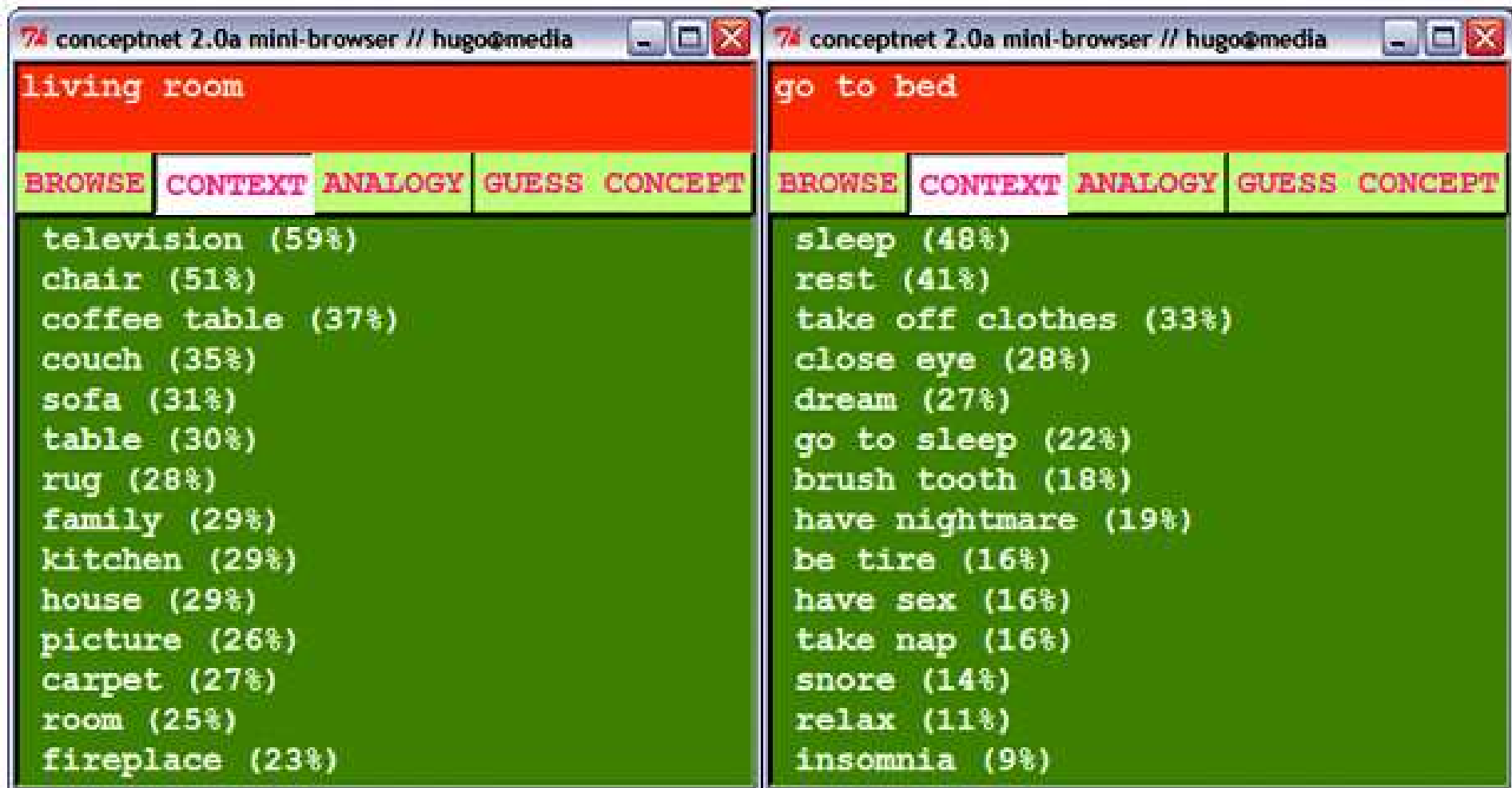


Fig 3 The results of two GetContext() queries are displayed in the ConceptNet knowledge browser.

```
74 conceptnet 2.0a mini-browser // hugo@media
war
BROWSE CONTEXT ANALOGY GUESS CONCEPT GUESS TOPIC ESS MO
[~ fire] (27.775489817)
==CapableOfReceivingAction==> stop
==PropertyOf==> bad
==CapableOf==> hurt person
==PropertyOf==> dangerous
==CapableOf==> kill person
==CapableOf==> destroy property

[~ murder] (23.6768166677)
==PropertyOf==> evil
==DesirousEffectOf==> complain about state of world
==PropertyOf==> wrong
==PropertyOf==> bad

[~ pollution] (7.69392452542)
==PropertyOf==> evil
==PropertyOf==> bad
==CapableOf==> spread

[~ gun] (21.217438356)
==CapableOf==> kill
==PropertyOf==> bad
==CapableOf==> hurt person
==PropertyOf==> dangerous

[~ car] (19.4276742038)
==CapableOfReceivingAction==> stop
==CapableOf==> kill
==PropertyOf==> expensive
==CapableOf==> kill person
==CapableOfReceivingAction==> start
```

09.06.20
Begin of

Mood guessing in ConceptNet

- ConceptNet uses 6 of Ekman's 7 basic moods
- Topic guessing functionality is used internally, combined with keyword matching
- Output: percentage of the basic moods (see demo)

Testing the Mood guessing function

- LiveJournal.com → Raw text annotated with moods
- Mapping between LiveJournal moods and Ekman's ground moods
- Running the function
- Comparing

Live Journal Moods

- <http://www.livejournal.com/moodlist.bml>
- <http://www.livejournal.com/moodlist.bml?moodtheme=70&mode=tree>

Mapping between different “mood systems“ #1

- Hierarchy 1 (no deviance from ideal mapping)
- happy
 - * happy (#15)
 - * okay (#61)
- sad
 - * sad (#25)
- angry
 - * angry (#2)
- fearful
 - * scared (#46)
- disgusted
- surprised

- Dismissed: * working (#88), * indescribable (#48), * nerdy (#102), * awake (#87), * energetic (#11), * thoughtful (#30), * determined (#45), * confused (#6), * devious (#130), * enthralled (#13)

Test runs and test results

- Two test runs have been done
 - 1:1 mapping
 - 1:many mapping
- Precision / Accuracy / F-measure not applicable
- 1:1 mapping
 - 38,5% correct
- 1:many mapping
 - 36,8% correct
- Baseline 1/6, one of six possible results: 16,6%

Improvement

- Key word matching is not optimal
 - Insertion of more keywords
 - Better choice of keywords, according to psycholinguistic studies
- Mood guessing relies highly on topic guessing functionality – improve topic guessing
- Enlarge ConceptNet in general

Demo

- ConceptNet Demo:
<http://conceptnet.org>

LOGOS

