Using WordNet in NLP

Christiane Fellbaum

A few more words on WN construction

Bootstrapping wordnets

Parallel corpora

(think of restaurant menu in several languages)

Corpora that are translated into many languages and in the public domain: the Bible, Orwell's *1984,* parliamentary proceedings (Canada, Europe)

Automatic alignment on the paragraph, then sentence level (96% agreement with human judgment, Gale and Church 1993)

Approach based on length (no. of characters) works suprisingly well

Word alignment:

Can automatically align certain strings

--proper names (*Lady Gaga, New York City*)

--dates (*1988*)

Some seed translated words (*house-maison*)

Parsing, POS assignment produces good word alignment

Bootstrapping WNs via parallel corpora

So we have aligned (French) *espace* and (Engl.) *room*

How do we know which sense of *room* is the context-appropriate? Some WordNet options:

(1) *room (an area within a building enclosed by walls and floor and ceiling) "the rooms were very small but they had a nice view"*

(2) *room, way, elbow room (space for movement) "room to pass"; "make way for"; "hardly enough elbow room to turn around"*

Solution

Look for words in context and match to WordNet definitions, related words

French context (1): *mur, sol, plafond*,…

(*wall, floor, ceiling*,..) => likely English sense (1)

vs. French context (2):

assez (enough), pour (for/in order to) => likely English sense (2)

Bootstrapping WNs via parallel corpora

Extension to words with similar contexts

E.g., distribution of French *espace* and *place* are similar

This makes them synonym candidates in French wordnet

Can also apply lexical-semantic patterns to detect related words in corpora of other languages (Xs and other Ys, etc.)

Application of WordNet

Supplies semantic priming data for psychologists

One word “primes” another when the two are semantically related

(Swinney 1979)

E.g., people recognize the word “nurse” faster after having been exposed to a related word like “doctor”

(but “apple” does not prime “nurse”)

Applications of WN

Lexical chaining (Morris & Hirst 1991)

Automatically detect words that don't fit the contexts, i.e., neighboring words that are not connected in WordNet and "break the chain"

These could be misspellings, malapropisms

(*ingenious-ingenuous, etymology-entomology*)

history, words insects, bugs

*In the historical linguistics class, the professor explained the **entomology** of "punch"*

Can build corrective tools for editing (note that chains do better than spell checkers, which don't catch correct spellings)

Detection of code words!

Applications of WordNet

IBM's "Watson" used WordNet to beat humans at "Jeopardy" (a kind of question answering application)

Can make connections among words/concepts that humans often don't make

Prevalent use of WordNet:

Word sense disambiguation

Required for many applications, incl. information retrieval, text mining, summarization, machine translation, question-answering

Some require the selection of the context-appropriate concept/sense (synset) only; others of the context-appropriate word (synset member)

Polysemy and synonymy: problematic for NLP

Language *understanding* requires identifying context-appropriate sense and resolving polysemy

Language *generation* requires selection of context-appropriate word form and choose the appropriate synonym (also difficult for language learners…)

Classic, real MT example from a Russian math text translated into English:

"many complicated changeables"

(several complex variables)

We'll focus on understanding/polysemy resolving

Natural Language Processing

Stemming, parsing currently at >90% accuracy level

Word sense discrimination (lexical disambiguation) still a major hurdle for successful NLP ("lexical bottleneck")

Which sense is intended by the writer (relative to a dictionary)?

Best systems: ~60% precision, ~60% recall (but recall that human inter-annotator agreement isn't perfect, either!)

Recall: basic assumptions

- Natural language context is coherent
- Chomsky's semantically ill-formed example *Colorless green ideas sleep furiously*
- Words co-ocurring in context are semantically related to one another
- Given word w_1 with sense σ_1, determine which sense τ_1, τ_2, τ_3,…of word w_2 is the context-appropriate one
- WN allows one to determine and measure meaning similarity among co-occurring words

WN for WSD

Different approaches use different kinds of information in WN

Different algorithms, tweaks

Using definitions (glosses)

Lesk's (1986) question:

"How to tell a pine cone from an ice cream cone?"

(One) answer: the dictionary definitions of *pine* and *ice cream* have no words in common (no lexical overlap)

Intuition:

words with similar meanings must have shared words in their dictionary definitions

Using definitions

So: measure overlap of words in WordNet's definitions (glosses)

Two senses of *bank*, each similar to different words/concepts

{*bank*, "sloping **land** beside a **body of water**"}

{*shore*, "**land** along the edge of a **body of water**"}

{*bank*, "financial institution that accepts deposits and channels **money** into lending activities"}

{*check*, "written order directing a bank to pay **money**"}

Number of shared words allow one to assign a similarity score

Extended Lesk algorithm

If there's no overlap among the words in any of the definitions of the words, look for overlap in the glosses of words that are in a specific relation (hyponymy, meronymy, etc.) to the target word

Using definitions (glosses)

Lesk algorithms yields good results without explicitly disambiguated definitions

Recall gloss corpus: open-class words in WordNet's glosses are manually linked to the appropriate synsets

Using such data allows WSD not just by measuring overlap of words (strings) but of meanings

Gloss Annotation

{debate, "**discussion** in which **reasons** are **advanced** for and against some proposition or **proposal**"}

{discussion, give_and_take,…}

{discussion, treatment,..}

{advance, move_forward,...}

{advance, progress,..}

{ advance, bring_forward}

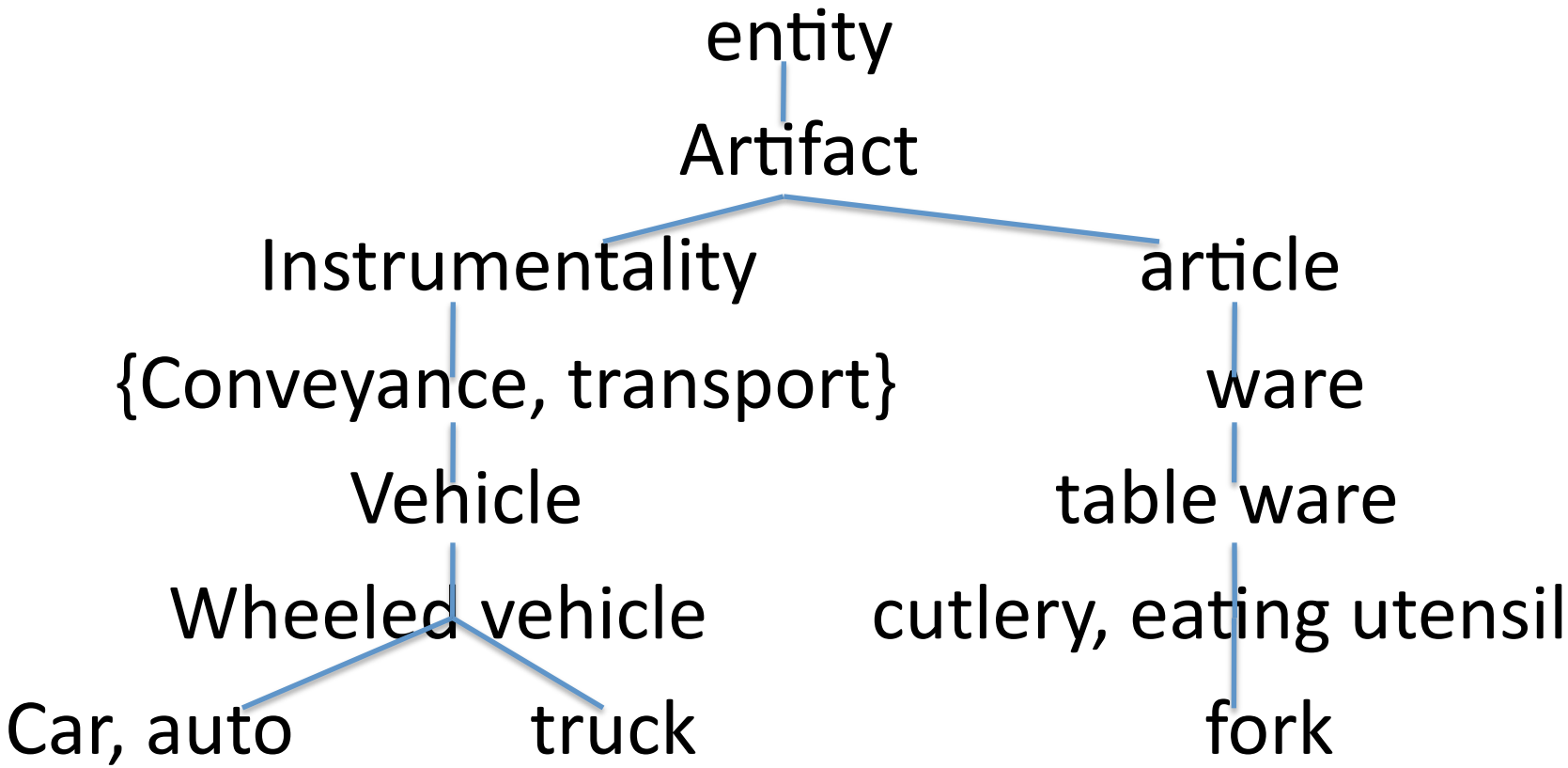
Other WordNet-based approaches

Structural: Intuition that words/meaning that are "close" to one another in the network (WordNet) are similar

If we encounter one in a context, others (with multiple meanings) can be disambiguated

Edge or node counting yields similarity score

Different kinds of paths may carry different weights


entity
Artifact
Instrumentality
article
{Conveyance, transport}
ware
Vehicle
table ware
Wheeled vehicle
cutlery, eating utensil
Car, auto
truck
fork

Similarity strength

(e.g., Hirst & St-Onge 1998)

Strong similarity:

synset members

synset members that are antonyms: *cheap-expensive)*

Medium strong or **weak** similarity: defined by different path lengths (<6)

changes in directions (change in relations)

combinations of length and direction changes lower similarity score/can be tweaked

Deriving meaning similarity metrics from WN's structure

Problem: edges are not of equal length in natural language (i.e., semantic distance is not uniform)

possession — *elephant*

| —— |

white elephant — *Indian elephant*

Similarly: both *home* and *hell* are subordinates of *location*, but are they really similar?

Some categories are huge (possession), others small (elephant)

One solution

Scale: take depth of the hierarchy into account

Corrections for differences in edge lengths: overview

- Scale by depth of hierarchies of the synsets whose similarity is measured (path from target to its root): the deeper the hierarchy, the more similar are words that are members of that hierarchy (Leacock and Chodorow 1998)
- Density of sub-hierarchy: intuition that words in denser hierarchies (more sisters) are very closely related to one another (Sussna 1993, Agirre and Rigau 1996)
- How many arcs of the same kind are leaving the superordinate? Fanout effect reflects dilution between super- and subordinate (Sussna 1993); scale by depth of the tree; siblings further down in the tree are more similar
- Types of link (IS-A, HAS-A, other): too many "changes of directions" reduce score (Hirst & St. Onge 1998)

- Wu and Palmer (1994)

Leacock and Chodorow (1998)

Similarity between two words

w1,w2 = -log (min length c1,c2/2D)

min length c1,c2 = length of shortest path between c1,c2

Length is measured in nodes traversed (not edges); synonyms have length 1

D = maximum depth of hierarchy (assuming unique beginner)

Leacock and Chodorow (1998)

Work through a couple of simple examples

Wu and Palmer (1994) Conceptual similarity

Sim c_1,c_2 = 2xN3/N1+N2+2xN3

N1=length of path c1 to lowest common subsumer (LCS) of c_1,c_2

N2= path length c_2 to c_3

N3= path length c_3 to root node

Greater distance of nodes to LCS => lower similarity

Other distance and similarity measures

Sussna (1993)

Intuition:

How many arcs of the same kind are leaving the superordinate?

“Fanout effect” reflects semantic “dilution” between super- and subordinate

Sussna (1993)

Each relation has a weight

Synonymy w=0, Antonymy w =2.5

Hypo-/hypernymy w ranges min=1 to max=2

w crucially depends on the number of arcs leaving the node (**n**): the greater **n**, the lower the weight

n differs depending on the direction of the link

Relation weight w for upward direction (usually) gets greater weight than downward

(but: some nodes are assigned multiple superordinates)

$$w(c_1 \Rightarrow_r c_2) = max - \frac{(max_r - min_r)}{\mathbf{n}_r(c_1)}$$

Sussna

This formula accounts for the category size: more arcs leaving a node “dilutes” weight between that node and its “target” nodes

Formula accounts for the asymmetry resulting from the difference in directions (up vs. down in the tree)

Sussna (1993)

Intuition:

siblings further down in the tree are more similar to one another than siblings "higher up"

Semantic distance between adjacent (sister) concepts $c_{1,}$ c_2 is scaled by depth of the tree (**d**)

D c_1 c_2= w (c_{1r} => c_2)+ w (c_{2r} => c_1)/2**d**

Sussna (1993)

Semantic distance between **any** two nodes:

Sum of the semantic distance (as per the previous formula) between pairs of adjacent nodes connecting c1, c2 along the shortest path

(it's easier than it looks—work out an example or two using any tree)

Many researchers combine knowledge-based approaches (using WordNet-based algorithms) with statistical measures

Resnik (1995)

Jiang and Conrath (1997)

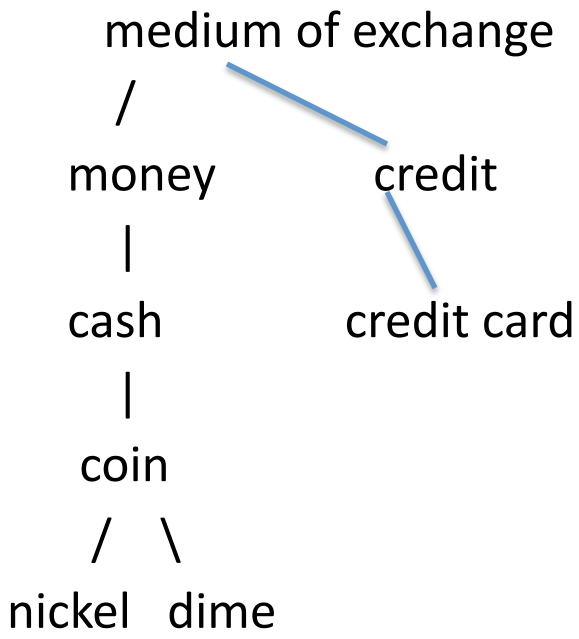
Information-based similarity measure (Resnik)

Intuition: similarity of two concepts is the degree to which they share information in common

Can be determined by the concepts' relative position wrt the lowest common superordinate (LCS)

(Cf. edge counting: similarity between two nodes decreases with distance from the LCS)

Information-based similarity measure (Resnik)



Resnik (claims)

Information content of *nickel* and *dime* is greater than that of *money* (the former are more specific and thus more informative than the latter)

Information content o*f nickel* and *dime* are similar/same (similarity is greater wrt position to LCS)

Given that shared information value is a measure of similarity, it follows that concepts *nickel* and *dime* are more similar than *nickel* and *money*

Information-based similarity measure

Information content of a concept c:

$$IC(c) = -\log(p(c))$$

p is the probability of encountering a token of concept *c* in a corpus (Brown)

p ranges from 0 to 1

Combining trees with information

Define “class”: all concepts below their LCS

Classes: *coin, medium of exchange*

Class notion entails that occurrence of any member of the class counts as occurrence of all class members (i.e., encountering **either** *nickel* **or** *dime* in the corpus counts as encountering *coin*)

(Problem: polysemy! There may be several classes of which the word *nickel* is a member)

Information and trees

p=1 for the LCS (least informative)

p<1 for class members (subordinates); *p* increases monotonically as you go up the hierarchy until you hit the LCS whose p=1

p is lowest for the lowest node (most informative, most specific concept)

Probability of a word is the sum of all counts of all class members divided by total number of words in the corpus

Information-based similarity measure (Resnik)

Similarity of a pair of concepts is determined by the least probable (most informative) class they belong to

$$sim_R(c_1\ c_2) = -\log_p(lcs(c_1\ c_2))$$

Thus, the higher the LCS of a pair of concepts, the lower their similarity (it's 0 for the root node)

Similarity measures

Good reference for comparing different similarity measures:

Ted Pedersen::WordNetSimilarity

http://wn-similarity.sourceforge.net

Adding Dense, Weighted Connections to WordNet

(Boyd-Graber, Fellbaum, Osherson, Schapire 2006)

Address specific shortcomings of WN:

--sparsity of links

--few cross-POS links (there are really four different networks, one for each major POS)

--links are not weighted

--bidirectionality of arcs not always felicitous (*dollar* is associated with *green*, but *green* less so with *dollar*)

Basic Idea

Connect **all** synsets (within/across POS) by means of **directed, weighted** arcs

(What's wrong with the idea of "all"?)

- Dense network can be exploited to find related/unrelated words and concepts
- Graded relatedness allows for finer distinctions
- Less training data needed for automatic WSD
- Algorithms relying on net structure will yield better results

From WordNet to WordNetPlus

- Cross-POS links (*traffic, congested, stop*)
- New relations: *Holland-tulip, sweater-wool, axe-tree, buy-shop, red-flame*,...
- Relations are not labeled
- Arcs are directed:*dollar-green/*green-dollar*
- Strength of relation is weighted

"Evocation"

- Based on synset, not word, pairs
- "How strongly does S_1 bring to mind S_2?"
- Exclude idiosyncratic associations (*grandmother-pudding*)
- Exclude formal similarity (*rake-fake*)
- Most synset pairs will not be related by evocation
- Cf. also Resnik's distinction between similarity and relatedness

Getting from WordNet to WordNetPlus

Semi-automatically identified 1K “core”synsets
(frequent, salient)

Experiment: Collecting Human Ratings

- Designed interface for ratings with sliding bar to indicate strength of association
- Strength of evocation ranged from 0-100
- Five anchor points (*no/remote/moderate/strong/ very strong association*)

Two experimenters rated evocations for two groups of 500 synsets each ("gold standards" for training and testing)

- Mean correlation was .78

Evocation Ratings: Training and Testing

24 Princeton students rated evocations for one group of 500 synsets, the training set

After each rating, the gold standard rating appeared as feedback

Students then rated the second group of 500 synsets without feedback (testing)

Calculated Pearson correlation betw. annotators' ratings and gold standard

median .72

lowest .64

avg. correlation between training and testing .70

Collecting Ratings

- Post-training/testing: collected 120K judgments for randomly chosen synsets (subset of 1K)

- At least three raters for each synset pair

Example Ratings

code-sip	0
listen-recording	60
pleasure-happy	100

Two thirds of ratings (67%) were 0 (as expected)

WordNetPlus Ratings and Other Similarity Measures

Rank order Spearman Coefficient for similarity measures (cf. WordNet::Similarity, Pedersen & Pathwardhan)

Leacock & Chodorow (similarity based on WordNet structure): 0.130

Lesk (overlap of strings in glosses): 0.008

Peters' Infomap (LSA vectors from BNC): 0.131

WordNetPlus Ratings and Other Similarity Measures

Lack of correlation shows that Evocation is an empirical measure of semantic similarity that is not captured by the other measures!

Partial explanation:

WordNet-based measures are within, not across, POS

Many WN-based similarity measures exclude verbs, adjectives

LSA is string, not meaning-based

Measures are based on symmetric relations, but evocation is not

Scaling Up

- Collection of 120,000 ratings took one year
- To connect all 1,000 synsets, 999,000 ratings are needed
- Too much to do manually!
- Current work: build an annotator “robot”
- Learn to rate evocations like a human

Features for Machine Learning

Different WN-based measures

Context vectors derived from a corpus

(relative entropy, frequency)

Learning Evocations

- Boosting (Schapire & Singer's BoosTex)
- Learns to automatically apply labels to examples in a dataset

Preliminary Results

- Algorithm predicted the right distribution of evocation (many 0's)
- For some data points with high (human) evocation ratings, prediction was zero evocation
- For many data points with zero (human) evocation, high evocation was predicted
- Best performance on nouns
- Worst on Adjectives

Work continues…(Nikolova et al. with AMT)