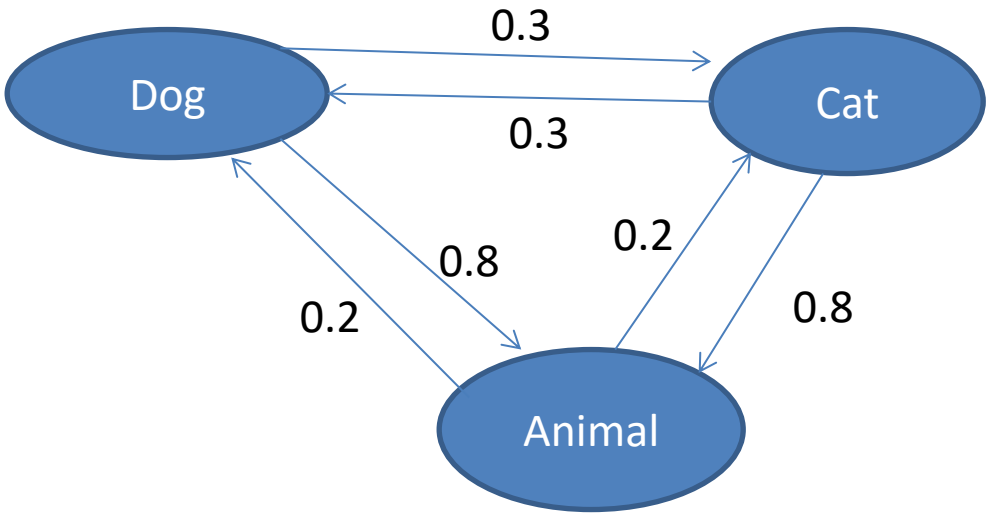
Creating a Probabilistic Graph using Markov Logic Network

Lubomir Stanchev

Example Probabilistic Graph



Applications

- If we type automobile in our favorite Internet search engine, for example Google or Bing, then all top results will contain the word automobile. Most search engines will not return web pages that contain the word car but do not contain the word automobile as one of the top results. The similarity graph will allow us to not only perform semantic search (i.e., search based on the meaning of the words), but it will also help us rank the result.
- We can use the semantic graph to cluster a set of documents based on the meaning of the words in them.
- The similarity graph can also be used as part of a query-answering system, such as the IBM Watson Computer that competed on the Jeopardy game show and the Siri system for the iPhone.

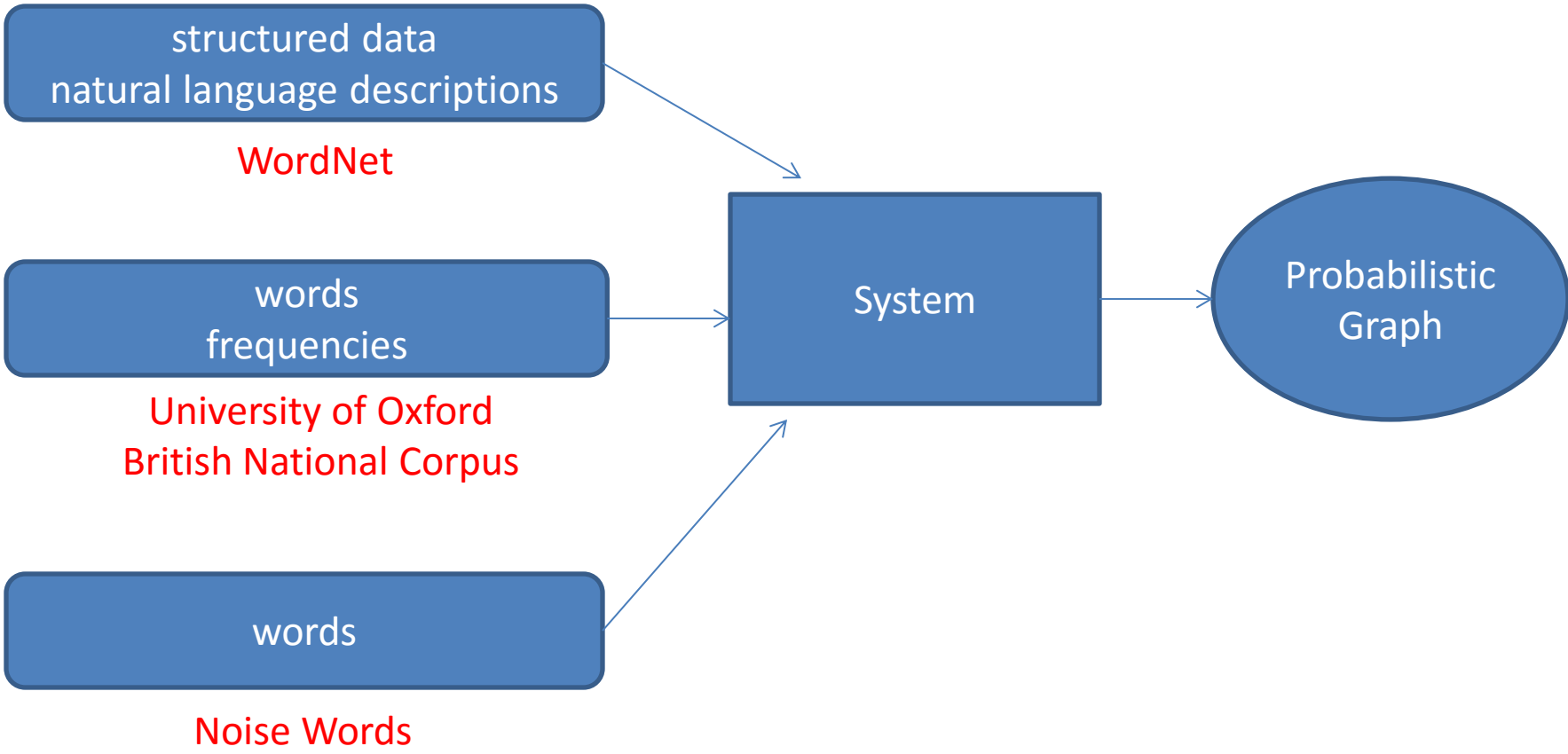
About WordNet

- WordNet gives us information about the words in the English language.
- In our study, we use WordNet 3.0, which contains approximately 150,000 different words.
- WordNet also contains phrases (or word forms), such as sports utility vehicle.
- The meaning of a word form is not precise. For example, spring can mean *the season after winter*, *a metal elastic device*, or *natural flow of ground water*, among others.
- WordNet uses the concept of a sense. For example, spring has the three senses.
- Every word form has one or more senses and every sense is represented by one or more word forms. A human can usually determine which of the many senses a word form represents by the context in which the word form is used.

About WordNet (cont'd)

- WordNet contains the definition and example use of each sense. It also contains information about the relationship between senses.
- The senses in WordNet are divided into four categories: nouns, verbs, adjectives, and adverbs.
- For example, WordNet stores information about the hyponym and meronym relationship for nouns. The hyponym relationship corresponds to the ``kind-of'' relationship (for example, dog is a hyponym of canine).
- The meronym relationship corresponds to the part-of relationship (for example, window is a meronym of building). Similar relationships are also defined for verbs, adjectives, and adverbs.

Our System


structured data
natural language descriptions
WordNet
words
frequencies
University of Oxford
British National Corpus
words
Noise Words
System
Probabilistic
Graph

Initial Similarity Graph

- Create a node for every word form.
- Create a node for every sense.

Processing the Senses

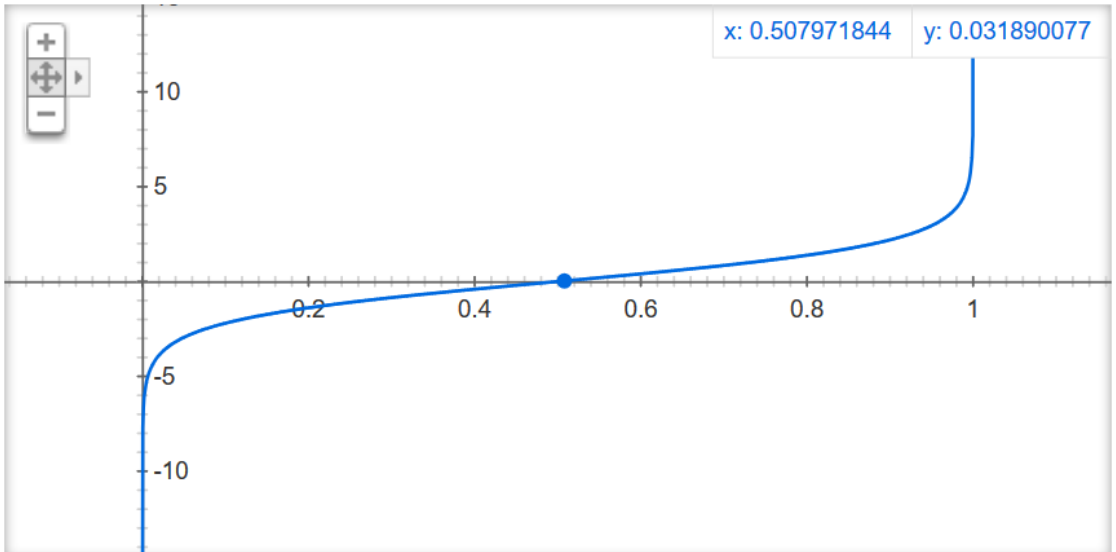
```
rel(chair) => rel(a seat for one person),  (35/38)
rel(chair) => rel(the position of a professor), (2/38)
rel(chair) => rel(the officer who presides at meetings),(1/38)
```

```
rel(a seat for one person) => rel(chair), 10
rel(the position of a professor) =>rel(chair), 10
rel(the officer who presides at meetings) => rel(chair), 10
```

Frequency of use of each sense is given in WordNet.

weight = ln(p/ (1-p))

$p = 1/(1+e^{w})$



Adding Definition Edges

```
rel(the position of a professor) => rel(position) (0.6)
rel(the position of a professor) => rel(professor) (0.4)
rel(position) = rel(the position of a professor) (0.19)
```

$$computeMinMax(minValue, maxValue, ratio) = minValue + (maxValue - minValue) * \frac{-1}{log_2(ratio)}$$

- Position is first word, so we give it greater importance.
- Forward edge: computeMinMax(0,0.6,ratio).
- If position appears in only three word form definitions,

then we compute backward edge as computeMinMax(0,0.3,1/3).

Processing Hyponyms

```
rel(a seat for one person) =>
  rel(chair with a support on each side for arms) (0.9*657/1208)
rel(a seat for one person) =>
           rel(a movable chair on large wheels) (0.9*551/1208)

rel(chair with a support on each side for arms) =>
                              rel(a seat for one person)  (0.3)
```

In the British National Corpus, the frequency of armchair is 657 and the frequency of wheelchair is 551.

Validating the Algorithm

- Miller and Charles study: 28 pairs of words. Study performed in 1991. Asked humans to write the similarity for pairs of words and recorded the results.
- WordSimilarity-353 study: 353 pairs of words. Study performed in 2002. Again, asked humans to write the similarity for each of the 353 pairs.
- We will use these benchmarks to validate our system.
- Need a way to measure the similarity between two words.

The Markov Logic Network Model

$$A_n \xrightarrow{w_{n-1}} A_{n-1} \quad \cdots \quad A_1 \xrightarrow{w_0} A_0$$

P(rel(A_n)|rel(A_0))= $$\frac{P(rel(A_0) \wedge rel(A_n))}{P(rel(A_n))} = \frac{f11(n-1)}{f10(n-1) + f11(n-1)}$$

$$f00(0) = e^{w_0}$$
$$f01(0) = e^{w_0}$$
$$f10(0) = 1$$
$$f11(0) = e^{w_0}$$

$$f00(i) = f00(i-1) * e^{w_i} + f10(i-1) * e^{w_i}$$
$$f10(i) = f00(i-1) * 1 + f10(i-1) * e^{w_i}$$
$$f01(i) = f01(i-1) * e^{w_i} + f11(i-1) * e^{w_i}$$
$$f11(i) = f01(i-1) * 1 + f11(i-1) * e^{w_i}$$

Measuring Semantic Similarity Between Words

$$|wf_1, wf_2|_{lin} = min(\alpha, \frac{P(wf_1|wf_2) + P(wf_2|wf_1)}{2}) * \frac{1}{\alpha}$$

$$|wf_1, wf_2|_{log} = norm(\frac{-1}{log_2(min(\alpha, \frac{P(wf_1|wf_2)+P(wf_2|wf_1)}{2}))})$$

The probabilities are normalized (i.e., the sum of the weights of all edges that leave a node add up to one). $P(wf_1|wf_2)$ is computed by adding up the conditional probabilities on disjoint paths between the two nodes.

word 1	*word 2*	*M&C*	*Linear*	*Logarithmic*
car	automobile	3.92	0.99	0.99
gem	jewel	3.84	1.0	1.0
journey	voyage	3.84	1.00	1.00
boy	lad	3.76	0.72	0.80
coast	shore	3.7	0.95	0.96
asylum	madhouse	3.61	1.0	1.00
magician	wizard	3.5	1.0	1.00
midday	noon	3.42	0.95	0.96
furnace	stove	3.11	0.98	0.98
food	fruit	3.08	0.03	0.26
bird	cock	3.05	0.63	0.73
bird	crane	2.97	0.76	0.82
tool	implement	2.95	0.78	0.83
brother	monk	2.82	0.41	0.59
crane	implement	1.68	0.00004	0.11
lad	brother	1.66	0.53	0.67
journey	car	1.16	0.10	0.36
monk	oracle	1.1	0.00004	0.18
food	rooster	0.89	0.42	0.60
coast	hill	0.87	0.00004	0.11
forest	graveyard	0.84	0.00004	0.11
monk	slave	0.55	0.00004	0.11
coast	forest	0.42	0.00004	0.11
lad	wizard	0.42	0.00004	0.11
chord	smile	0.13	0.00004	0.11
glass	magician	0.11	0.00004	0.11
noon	string	0.08	0.00004	0.11
rooster	voyage	0.08	0.00004	0.11

Experimental Results

algorithm	*correlation*
Hirst and St-Onge [9]	0.74
Leacock and Chodorow citeLC98	0.82
Resnik [23]	0.77
Jiang and Conrath [12]	0.85
Lin [18]	0.83
$\|\cdot\|_{lin}$ [34]	0.93
$\|\cdot\|_{log}$ [34]	0.93
$\|\cdot\|_{lin}$	0.89
$\|\cdot\|_{log}$	0.90

Miler and Charles

algorithm	*correlation*
Jarmasz[10]	0.27
Hirst and St-Onge[9]	0.34
Jiang and Conrath [12]	0.34
Strube and Ponzetto[40]	0.19-0.48
Leacock and Chodrow[17]	0.36
Lin[18]	0.36
Resnik[23]	0.37
Bollegala et al.[2]	0.50
$\|\cdot\|_{lin}$ [34]	0.54
$\|\cdot\|_{log}$ [34]	0.54
$\|\cdot\|_{lin}$	0.52
$\|\cdot\|_{log}$	0.53

WordSimilarity-353

Conclusion and Future Research

- We presented an algorithm for building a similarity graph from WordNet. We verified the data quality of the algorithm by showing that it can be used to compute the semantic similarity between word forms and we experimentally verified that the algorithm produces better quality results than existing algorithms on the Charles and Miller and WordSimilarity-353 word pairs benchmarks.
- We believe that we outperform existing algorithms because our algorithm processes not only structured data, but also natural language.