Word Representations

- How do we represent the meaning of words?
- How can we learn representations from data?

1 WordNet

- Early attempt by George Miller to create relationships between words.
- Lexical database: machine-readable dictionary and thesaurus.
- Grouped words as **Synsets**
- Synsets are connected to each other by semantic relations.
    - **Hypernyms**: Y is a hypernym of X if every X is a (kind of) Y.
        - E.g., canine is a hypernym of dog.
    - **Hyponyms**: Y is a hyponym of X if every Y is a (kind of) X.
        - E.g., dog is a hyponym of canine.

```
from nltk.corpus import wordnet as wn
panda = wn.synset ('panda.n.01')
hyper = lambda s: s.hypernyms()
list (panda.closure(hyper))
```

- Problems with WordNet
    - Requires lots of manual labor
    - Doesn't exist in other languages
    - Hard to adapt, incorporate new words, reflect new usage
    - Is subjective
    - Is missing nuance
        - Words within a synset are not equivalent in their usage (practiced, skillful, expert, etc.).
    - Hard to compute word similarity

2 Representations from context

- ***Distributional Similarity***:
    - "You shall know a word by the company it keeps" —John Rupert Firth
        - Must represent a word with statistical properties of its context
    - One of the most successful ideas in modern NLP
    - Different approaches based on this idea to represent words
        - Cluster words based on their context.
            - Brown clustering algorithm
            - Class-based language models
        - Use co-occurrence matrix representation.
            - Compress the matrix to get continuous word vector.
            - Use Latent Semantic Analysis and so on.
        - Train a model that predicts context words (word2vec).
            - Continuous bag-of-words (CBOW) and Skip-gram.

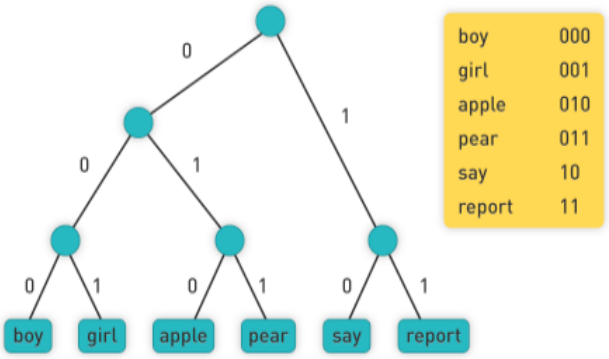
2.1 Word Clustering

2.1.1 Brown Clustering

- Developed at IBM in 1992. (Paper)
- Authors interested in class-based language models; addressed sparsity
    - For example, P ("to Shanghai") estimated from words similar to "Shanghai" if this is not present in training data.
- Used large corpus of text as input
    - Provided partition of words into classes as first output
        - Example of partitions created:

```
Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays
June March July April January December October November September August
people guys folks fellows CEOs chaps doubters commies unfortunates blokes
down backwards ashore sideways southward northward overboard aloft downwards adrift
water gas coal liquid acid sand carbon steam shale iron
```

- Provided hierarchical clustering of words as second output
  - Example hierarchy



- ***Brown clustering Intuition***
  - Similar words appear in similar contexts.
  - More precisely, similar words have similar distributions of words to their immediate left and right.
  - For example, “the” and “a” have similar contexts and are thus similar.

  - Similarly, Monday and Tuesday below

  -

End of Document