Distributed Distributional Similarity

Gareth Robinson
School of Information Technologies

ə-lab, Faculty of Engineering & Information Technologies

Distributional Similarity

The Distributional Hypothesis

"You shall know a word by the company it keeps." (Firth, 1957)

The Distributional Hypothesis proposes that we can infer the meaning of a word by looking at the context which the word is used in.

My bloobop is due in two weeks yet I am making a poster.

My supervisor will be disappointed with the quality of my honours bloobop.

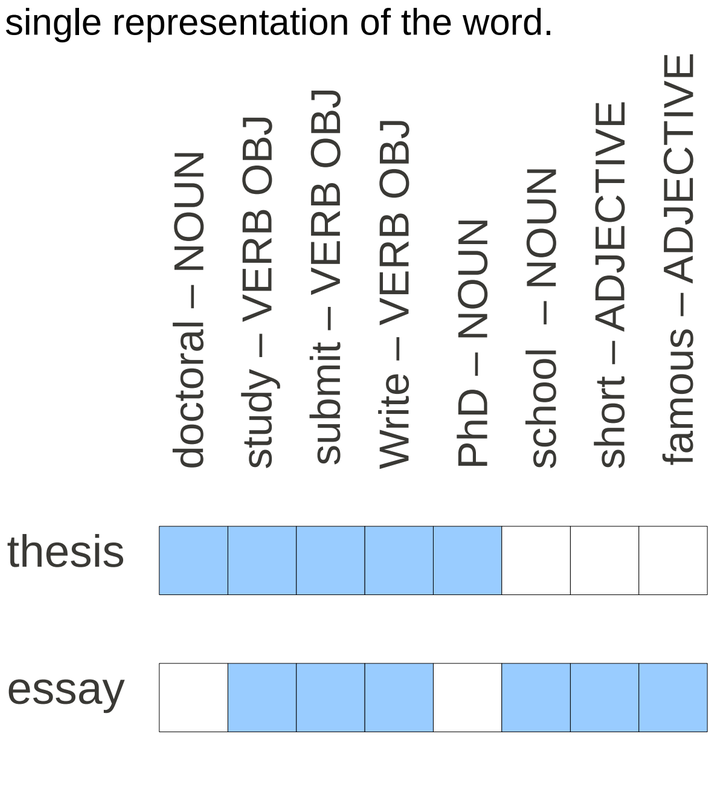
This PhD bloobop is quite extensive in its evaluation of early 20th century coat stands.

From the *context,* we can infer that the word *bloobop* has a meaning similar to *thesis.*

In distributional similarity, we aim to leverage this to find synonyms for words. This is useful for constructing thesauri, especially in new domains, spelling corection, and entity set expansion.

Context

From each occurrence of a term in the source text, we extract the context surrounding the term. This consists of the term, the context, and the part of speech pertaining to the relation between the term and the context. The contexts for each term are collated into a single representation of the word.



Context vectors for *thesis* and *essay*. Blue square indicates the term was seen in that context.

Similarity

The collated representations of terms are compared using the Jaccard Similarity Coefficient, which measures the similarity of two sets.

Most similar words for ***thesis****:*

essay finding topic conclusion perspective qualification curriculum proposition

Most similar words for ***resentment****:*

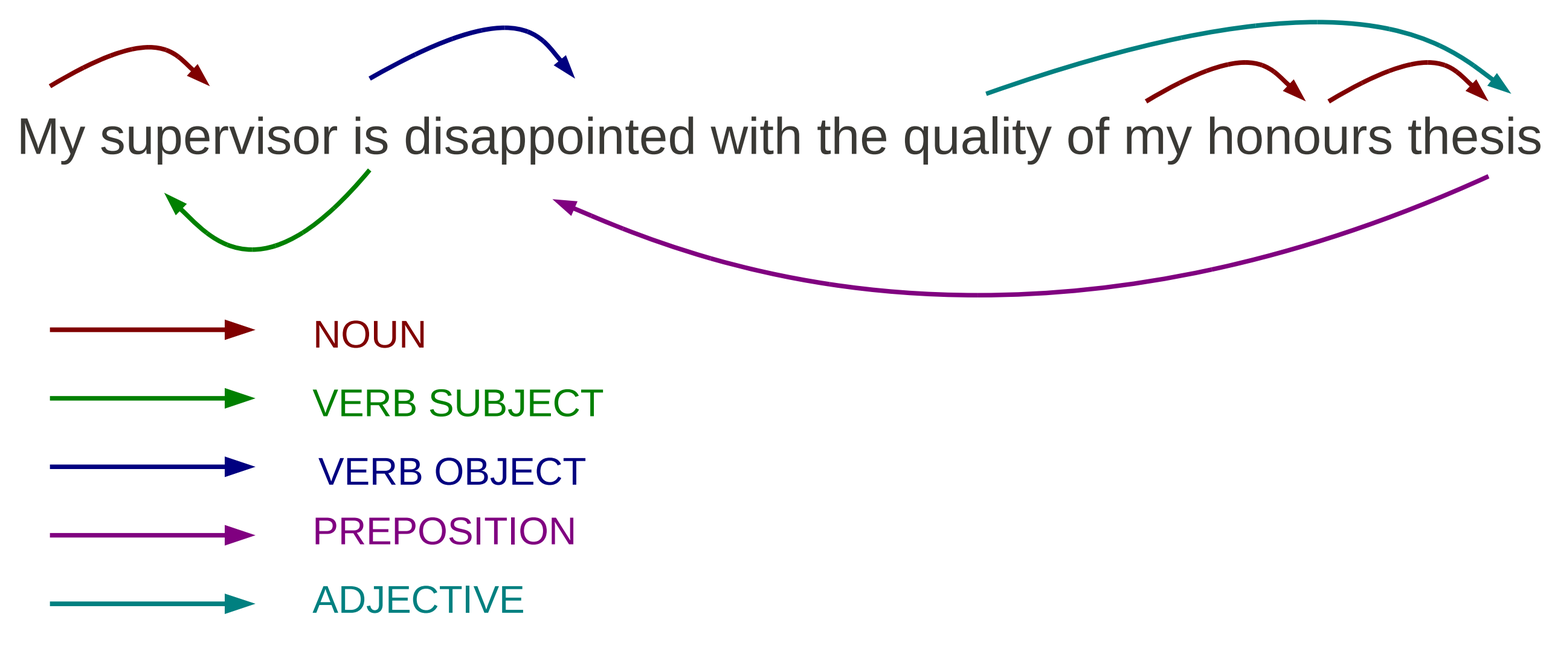
confusion hatred bitterness disappointment suspicion anger anxiety frustration hostility

Data Size

When using a statistical approach to language processing, the more data which we can process, the higher the quality of the result.

A naïve approach is to compare every term to every other term. This has a complexity of $O(n^2m)$, where *n* is the number of terms to be compared, and *m* is the average number of contexts in each term.

In this research we explore methods to improve the time complexity of the search, and implement these methods in a distributed fashion, to increase the amount of data which can be processed.



Distributing the search

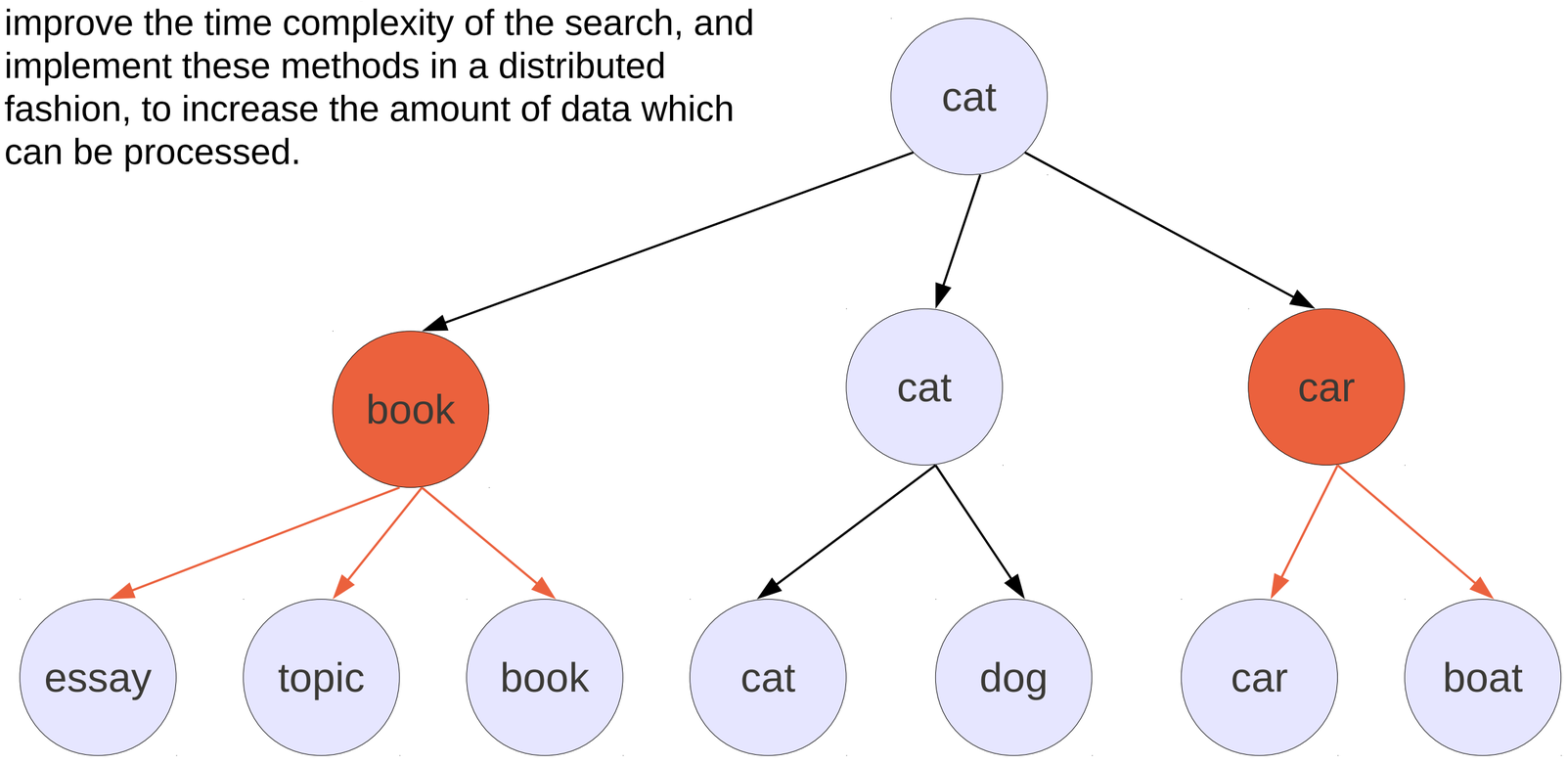
Rank Cover Tree

The contribution of this research is to introduce the Rank Cover Tree (Houle, 2011) to the field of distributional similarity. The Rank Cover Tree produces approximate results to a K nearest neighbour search. As a tree structure, this process can also be distributed.

The Rank Cover Tree is constructed from the terms to be compared. Each term is assigned a maximum level in the tree, and appears in that level and every level below it. Each term has as its parent the closest term in the level above.

To search through the tree, a beam search is used. Starting from the root, the closest at most *n* nodes are selected. For each level down the tree, the children of the best nodes are inspected, with the remainder discarded. This is continued until the bottom level of the tree is reached.

Using this approach we are able to compute similarity terms in a distributed fashion *20 times faster*, at only a *3%* loss of accuracy compared to the optimal solution.



Rank Cover Tree, searching for words similar to *thesis*, at level = 2. Current terms in the beam and children to be considered at next level highlighted orange

Acknowledgements

This work was supported by Australian Research Council Discovery grants DP0665973 and DP1097291, and the Capital Markets Cooperative Research Centre.

References

1. J. R Firth. A synopsis of linguistic theory 1930-1955. pages 1–32, 1957.
2. Michael E. Houle. Rank cover trees for nearest neighbor search. In Proceedings of The symposium on Discrete algorithms, 2011. (in prep.).