Computational Foundations of Cognitive Science

Lecture 8: Vector-space Models of Semantic Processing

Frank Keller

School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

February 23, 2010

Motivation

Vector representations can be used to model *word meaning*.

Based on these representations, we can determine if two words have similar meanings. The representations are built using co-occurrence counts in corpora.

Applications in cognitive science:

- model of human *language processing* (e.g., semantic priming, synonym detection);
- model of human *language acquisition* (e.g., lexicon development).

Engineering applications include information retrieval.

Distributional Hypothesis

Linguists have long conjectured that the *context* in which a word occurs determines its meaning:

- you shall know a word by the company it keeps (Firth);
- the meaning of a word is defined by the way it is used (Wittgenstein).

This leads to the *distributional hypothesis* about word meaning:

- the context surrounding a given word provides information about its meaning;
- words are similar if they share similar linguistic contexts;
- semantic similarity and be defined as distributional similarity.

Constructing Vector Spaces

Words occur in context:

car engine hood tires truck trunk

car emissions hood make model trunk

Chomsky corpus noun parsing tagging wonderful

Contexts can be obtained from corpora (large collections of text).

Note that we have already removed stop words (frequent words such as *the*, *of*, *although*).

Constructing Vector Spaces

Select target words:

car engine hood tires truck trunk

car emissions hood make model trunk

Chomsky corpus noun parsing tagging wonderful

Constructing Vector Spaces

Define the context (here: symmetric, -5, $+5$):

car engine hood tires truck trunk

car emissions hood make model trunk

Chomsky corpus noun parsing tagging wonderful

Constructing Vector Spaces

Define the context (here: symmetric, -5, $+5$):

car ***engine hood tires truck trunk***

car emissions hood make model trunk

Chomsky corpus noun parsing tagging wonderful

Constructing Vector Spaces

Define the context (here: symmetric, −5, +5):

car ***engine hood tires truck trunk***

car emissions hood ***make model trunk***

Chomsky corpus noun parsing tagging wonderful

Constructing Vector Spaces

Define the context (here: symmetric, -5, $+5$):

car ***engine hood tires truck trunk***

car emissions hood ***make model trunk***

Chomsky ***corpus noun parsing tagging wonderful***

Constructing Vector Spaces

Create co-occurrence matrix:

$$
\begin{array}{c l}
\begin{matrix} \text{car} & \text{hood} & \text{Chomsky} \end{matrix} & \\
\left[\begin{matrix}
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
0 & 0 & 1
\end{matrix}\right]
&
\begin{matrix}
\text{car} \\
\text{Chomsky} \\
\text{corpus} \\
\text{emissions} \\
\text{engine} \\
\text{hood} \\
\text{make} \\
\text{model} \\
\text{noun} \\
\text{parsing} \\
\text{tagging} \\
\text{tires} \\
\text{truck} \\
\text{trunk} \\
\text{wonderful}
\end{matrix}
\end{array}
$$

Constructing Vector Spaces

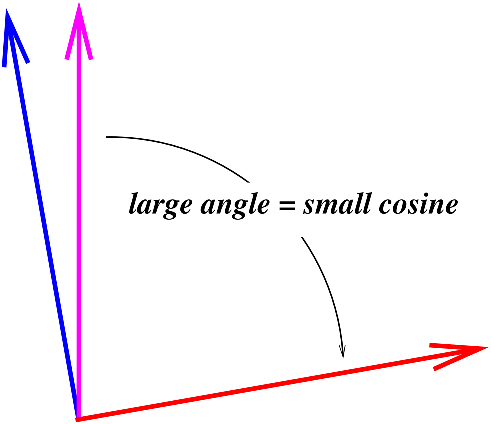
Informal algorithm for constructing vector spaces:

- pick the words you are interested in: *target words;*
- define number of words surrounding target word: *context window;*
- count number of times the target word co-occurs with context words: *co-occurrence matrix.*

The context can also be defined in terms of documents, paragraphs, or sentences (rather than words around target word).

Constructing Vector Spaces

Compare the word vectors using the cosine:



Measures of Distributional Similarity

The *cosine* of the angle between two vectors **x** and **y** is:

$$cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{||\mathbf{x}||\cdot||\mathbf{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

The *Euclidean distance* of two vectors **x** and **y** is:

$$||\mathbf{x}-\mathbf{y}|| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Many more similarity measures exist.

Dimensionality Reduction

Problem: the co-occurrence matrix can be very sparse (many zeros) and noisy (e.g., due to words with the same meaning).

auto engine bonnet tyres lorry boot

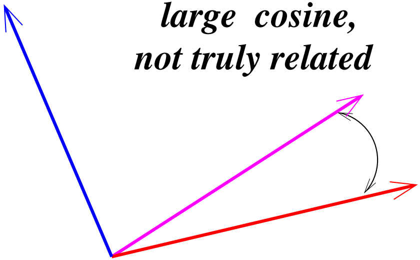
car emissions hood make model trunk

make hidden Markov model emissions normalize

Dimensionality Reduction

	auto	trunk	Markov
auto	0	0	0
bonnet	1	0	0
boot	1	0	0
car	0	1	0
emissions	0	1	1
engine	1	0	0
hidden	0	0	1
hood	0	1	0
lorry	1	0	0
make	0	1	1
Markov	0	0	1
model	0	1	1
normalize	0	0	1
tires	0	0	0
trunk	0	1	0
tyres	1	0	0

large cosine,
not truly related



Dimensionality Reduction

In order to address these problems, reduce the *dimensionality* of the co-occurrence matrix M:

- *project* the word vectors into a different subspace so that vector cosines more accurately represent semantic similarity;
- in a *lower dimensional space,* synonym vectors may not be orthogonal;
- *singular value decomposition* is a widely used projection methods;
- alternative: restrict matrix dimensions to most reliable words.

Mid-lecture Problem

Take the following text:

But Jones said the *octuplets* will soon realize that they have to wait for things to be done and they might not play up as much as children in smaller families.

Build a vector representation for *octuplets* based on a symmetric context of -5, $+5$ words.

Assume the dimensions of your vector: Jones, said, soon, realize.

Mid-lecture Problem

Now build the same vector for *octuplets* using the following text:

> But the *octuplets* said Jones will soon realize that they have to wait for things to be done and they might not play up as much as children in smaller families.

What's the different between the two vectors? Is this intuitively correct? How could it be fixed?

Limitations of Vector-space Models

A lot of relevant linguistic information is ignored:

- all words within window are considered related to the target word; all words outside window unrelated;
- all context words are equally important;
- distributional similarity is sometimes hard to interpret (e.g., *friend* and *enemy* are distributionally similar);
- *syntactic dependencies* between words are ignored.

Semantic Priming

Semantic priming: the processing of a target word is facilitated by the previous processing of a semantically related word (the prime).

Facilitation means: faster and more accurate processing (e.g., in lexical decision experiments).

fruit primes apple
desk doesn't prime apple

A large body of experimental literature exists that tests semantic priming for a various *lexical relations:*

- synonymy: *value–worth*; antonymy: *friend–enemy*;
- superordination and subordination: *pain–sensation*;
- category coordination: *truck–train*;
- association: *leash–dog*.

Results

We can use vector-space models to simulate semantic priming:

- take word pairs from the psychological literature;
- compute vector representations for target words and related and unrelated prime words;
- distributional distance between related prime and target should be smaller than distance between unrelated prime and target.

Padó and Lapata (2007) report:

- a reliable priming effect for a standard vector-space model ($\eta^2 = 0.284$);
- a model that takes syntactic dependencies between words into account performs better ($\eta^2 = 0.332$);
- using a larger contexts yields better results;
- but: lexical relations cannot be distinguished.

Detecting Synonymy

Two words can have the same meaning, this is called *synonymy*. Humans are very good at detecting synonymous words.

Synonymy is also used in language testing, for example in the TOEFL test of English.

You will find the office at the main intersection.	
(a)	place
(b)	crossroads
(c)	roundabout
(d)	building

This is a standard task in the cognitive modeling literature, and vectors space models are frequently used to solve it.

Detecting Synonymy

Two words can have the same meaning, this is called *synonymy.* Humans are very good at detecting synonymous words.

Synonymy is also used in language testing, for example in the TOEFL test of English.

You will find the office at the main intersection.	
(a)	place
(b)	crossroads
(c)	roundabout
(d)	building

This is a standard task in the cognitive modeling literature, and vectors space models are frequently used to solve it.

Results

We can use vector-space models to detect synonyms:

- take words from TOEFL synonym test;
- compute vector representations for question word and for suggested answer words;
- predict that the answer word with the smallest distributional distance to the question word is the correct one.

Padó and Lapata (2007) report:

- a standard vector-space model gets 61% correct;
- a model that takes syntactic dependencies between words into account gets 73% correct;
- average non-native speaker performance is around 65%; guessing gets you 25%.

Summary

- the meaning of words can be represented using vectors;
- these vectors tabulate distributional information about words, i.e., which other words they co-occur with;
- the distance between word vectors represents their semantic relatedness;
- problems: sparse representations; lexical relations; syntactic dependencies between words;
- applications in cognitive modeling:
  - semantic priming: words that are distributionally similar prime each other;
  - synonym detection: words that are distributionally similar are likely to be synonyms.

References

Padó, Sebastian and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* 33(2):161–200.