A Systematic Comparison of English Noun Compounds
Vered Shwartz

There are several ways to represent noun compounds as vectors.

The common alternative is to learn a composition function that operates on the vectors of the constituent words, typically with some arithmetic operations.

During training, the function is trained to estimate the distributional vector of each compound.

Composition Functions

A (w1, w2) → \[ \mathbf{v} \]

What would the vector of synaulete represent be like?

It may now be similar to company spokesperson. Composition allows generalizing from the constituent to the compound. But many of the nearest neighbors simply share constituents with the target compound.

Could it be due to the training objective? What if we trained the composition to be similar to vectors of other things which are known to be similar to the target?

Composition Functions

A (w1, w2) → \[ \mathbf{v} \]

Distributional Representation

\[ w_{\text{cheese wheel}} \rightarrow [ \mathbf{v} _{\text{cheese wheel}} ] \]

You can simply trad them as single tokens and learn word embeddings. It's the best way to represent frequent noun compounds.

But what about the many rare ones?

Oh, they're bad. Their nearest neighbors are 80% junks, like synaulete representatives: [golinsis, tframes, adaptents...]

Good point. In the general literature of phrase representation, it is common to encode phrases using an LSTM, and train to minimize the distance between paraphrases, such as street bed and ground floor.

Where do you get the paraphrases from?

We experimented with two sources: joint corpus occurrences of the constituents (computer power-of-cpu computing system) and translations of the noun compound to a foreign language and back to English Computing power... "Calculating capacity..."

Semantic Relation Classification

* On the TROTS (2011) dataset.
* Composition functions perform best.
* More compositional power->better.

Compositional representations performed best on classifying the semantic relation between the constituents (e.g., the is source, why all purposes). Especially when the underlying word embeddings were trained using a small window - this must give them a more "functional" nature.

But their absolute performance is still low on a lexical split of the data - with only F1=0.38 for the coarse-grained relation inventory and F1=0.3 on the fine-grained. So they don't generalize enough.

Property Prediction

* Based on Maltz Feature Norms (Milne et al., 2005).
* Paraphrase-based performs best.

In another experiment, we tried to see if we can use the noun compound vector to predict whether it holds a certain property or not - for example, is a cheese wheel round or not? The paraphrase-based representations performed best.

Looking forward we will need to address all the shortcomings of the existing representations. Not just better representations under the given assumptions. We will also need to consider context, handle non-compositional compounds, compounds with more than two words...

Thanks for listening!
The code is available and you can contact me if you have any questions.