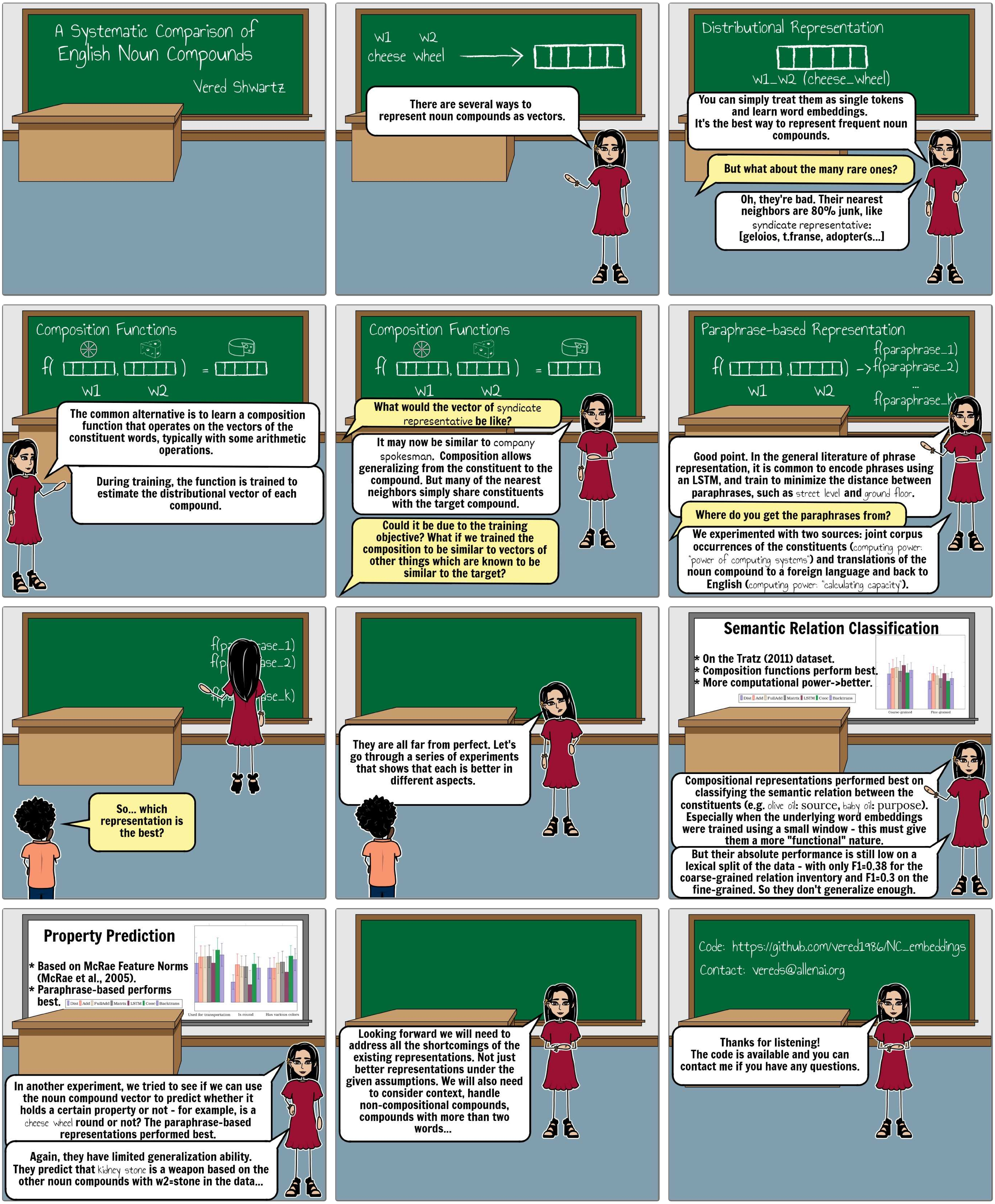
A Systematic Comparison of English Noun Compounds
Vered Shwartz
W1 W2
cheese wheel
There are several ways to represent noun compounds as vectors.
Distributional Representation
W1_W2 (cheese_wheel)
You can simply treat them as single tokens and learn word embeddings. It's the best way to represent frequent noun compounds.
But what about the many rare ones?
Oh, they're bad. Their nearest neighbors are 80% junk, like syndicate representative: [geloios, t.franse, adopter(s...]
Composition Functions
f(,) =
W1 W2
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
f(,) =
W1 W2
What would the vector of syndicate representative be like?
It may now be similar to company spokesman. 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?
Paraphrase-based Representation
f(,) -> f(paraphrase_1)
f(paraphrase_2)
...
f(paraphrase_k)
W1 W2
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 level and ground floor.
Where do you get the paraphrases from?
We experimented with two sources: joint corpus occurrences of the constituents (computing power: "power of computing systems") and translations of the noun compound to a foreign language and back to English (computing power: "calculating capacity").
f(pa ase_1)
f(p ase_2)
f(pa ase_k)
So... which representation is the best?
They are all far from perfect. Let's go through a series of experiments that shows that each is better in different aspects.
Semantic Relation Classification
* On the Tratz (2011) dataset.
* Composition functions perform best.
* More computational power->better.
Dist Add FullAdd Matrix LSTM Cooc Backtrans
Coarse-grained
Fine-grained
Compositional representations performed best on classifying the semantic relation between the constituents (e.g. olive oil: source, baby oil: purpose). 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 McRae Feature Norms (McRae et al., 2005).
* Paraphrase-based performs best.
Dist Add FullAdd Matrix LSTM Cooc Backtrans
Used for transportation
Is round
Has various colors
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.
Again, they have limited generalization ability. They predict that kidney stone is a weapon based on the other noun compounds with w2=stone in the data...
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...
Code: https://github.com/vered1986/NC_embeddings
Contact: vereds@allenai.org
Thanks for listening! The code is available and you can contact me if you have any questions.