Spot the Odd-Man-Out: Exploring the Associative Power of Lexical Resources

Gabriel Stanovsky[1,2], Mark Hopkins[3]

https://github.com/gabrielStanovsky/odd-man-out

[1]Allen Institute for Artificial Intelligence [2]University of Washington [3]Reed College

TL;DR

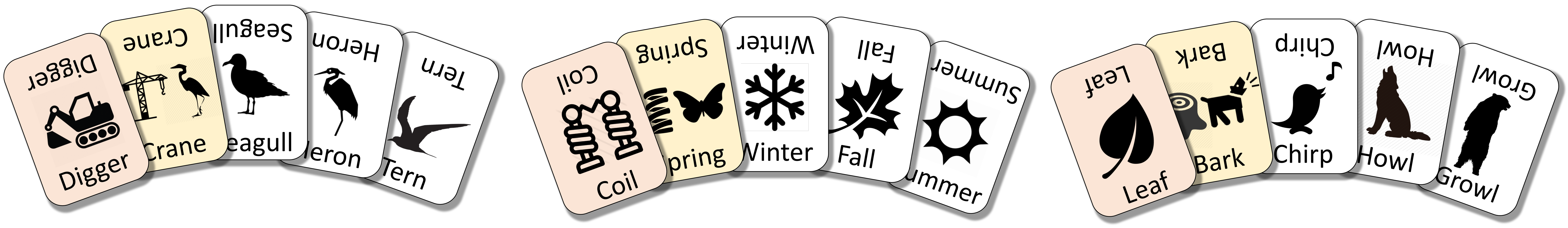
- Polysemous words (e.g., *spring, bat,* or *crane*) have distinct meanings in different contexts.
- Word embeddings traditionally average these different meanings into a single vector representation, which may harm performance in downstream tasks.
- We present the **Odd-Man-Out** puzzle, which targets word embedding's ability to correctly identify the different senses of polysemous words, by singling out a word that "doesn't belong" in a set of *n* words.
- **We show that this task is *intuitive* for humans, *cheap* to annotate, and *hard* for current word representations.**
- We collect and make available a large collection of Odd-Man-Out puzzles: **https://github.com/gabrielStanovsky/odd-man-out**

Task Definition

- Given a set of *n* words, a system is tasked with identifying the word which doesn't belong within the set (dubbed the *odd-man-out).*
- Our crowdsourcing methodology ensures that **within each puzzle hides a *distractor*** – a polysemous word which shares some category with both the odd-man-out and the other words in the puzzle.
- This requires the tested word embeddings to correctly disambiguate the distractor word to correctly identify its relation with the other words in the puzzle.

Related Work

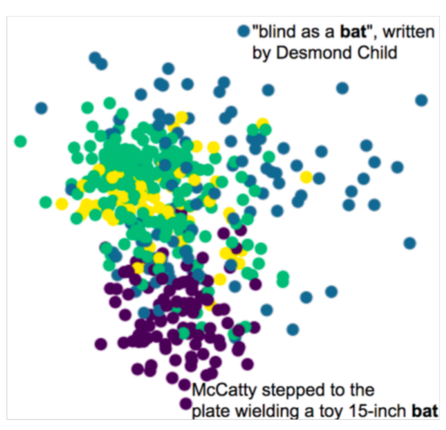
(Camacho-Collados and Navigli, 2016) create similar puzzles, focusing on outliers with varying degrees of similarities, while we target polysemous words, and collect data on a large scale.



An Odd-Man-Out is a puzzle composed of five words, **where one of the words doesn't fit with the others**.
In each puzzle, we plant a polysemous distractor which shares a common category with the odd-man-out.
We crowdsource 500K Odd-Man-Out puzzles semi-automatically with high inter-annotator agreement (84%).

ELMo Sense Embeddings

- ELMo word vectors (Peters et al., 2018) were recently shown to be useful in various tasks. However, they cannot be directly evaluated on our puzzles, since they rely on sentential context.
- We therefore first cluster the different representations of each word in a large unlabeled corpus.
- Then, we use the centroids of each cluster as representing different senses of each word.



Evaluation Framework

Intuitively, given an embedding function $E\colon \Sigma \rightarrow \{r \in R^d\}$, and a puzzle of n words, we look for the subset of $n-1$ words which minimizes some binary similarity score σ (e.g., cosine similarity).

Formally, we define a *cohesion score* over a set of words W as

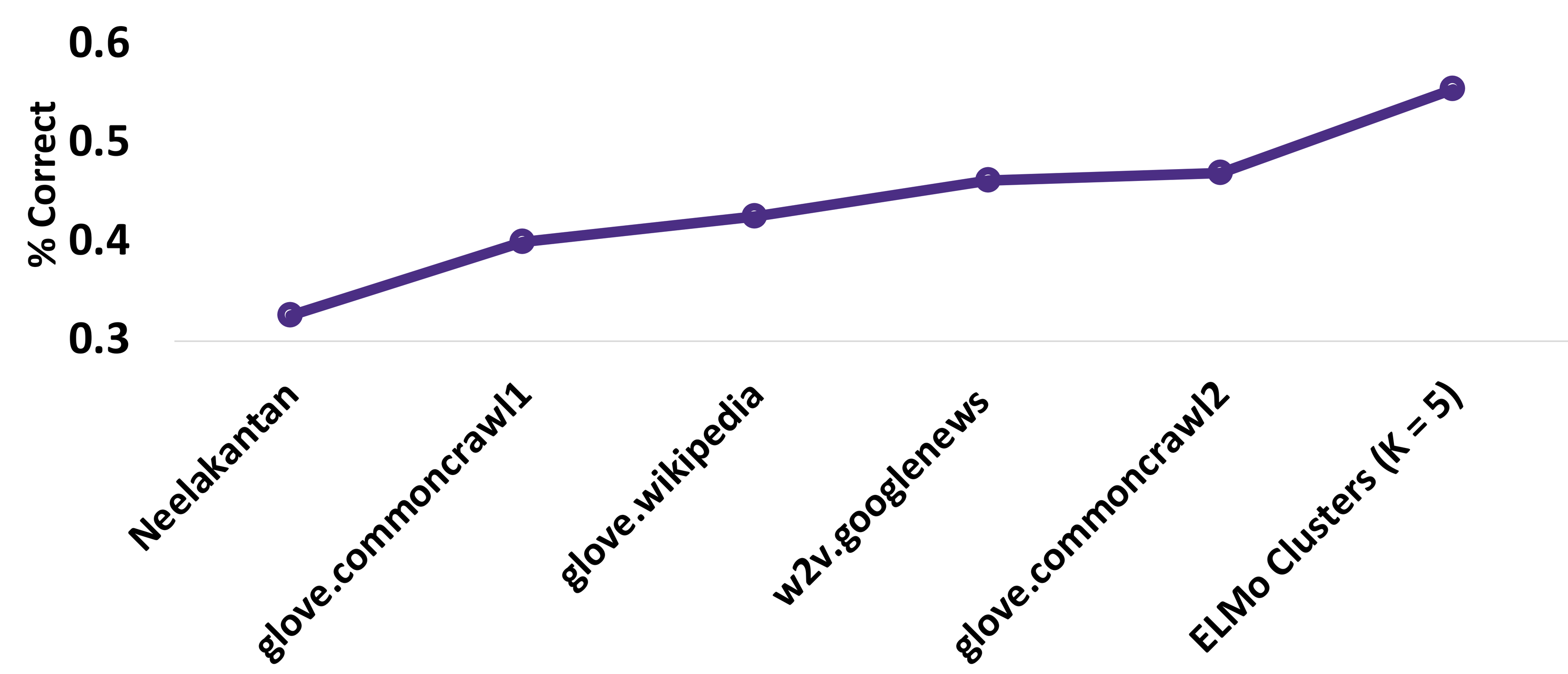
$$\kappa_{\sigma,E}(W) = \max_{v_i \in E(w_i)} \sum_{i \neq j} \sigma(v_i, v_j)$$

The solver returns $\widehat{w}$, such that –

$$\widehat{w} = \operatorname*{argmax}_{w_i \in W} \kappa(W \setminus \{w_i\})$$

Results

We tested various well-known word embeddings against our crowdsourced annotations:



Overall, while ELMo performed significantly better on all tested scenarios, we found that all of them are distracted by the polysemous words in the puzzles.

Conclusions

- Our evaluations suggest that current state-of-the-art word embeddings do not disambiguate well ($\leq$ 55% correct).
- Our crowdsourced corpus is made publicly available as a test-bed for future polysemous word disambiguation.