The task

- Many phrases have meanings not entirely predictable from the meanings of their parts (non-compositional)
- "bug" (usually compositional)
- "cold feet" (could be either)
- **Goal 1:** Using statistical methods, can we determine when phrases are compositional in a given context? (non-compositional phrase detection)
- **Goal 2:** For a given phrase, can we predict across a corpus whether it tends to be compositional or not? (non-compositional phrase induction)

We propose an unsupervised approach to both these tasks, using a probabilistic generative model

Building on a simpler task: word sense detection and induction

- To address goals 1 and 2, we extend an existing approach used in the simpler but analogous tasks of single word sense induction and detection
- Word sense detection is the task of classifying the sense of a word (e.g. distinguishing between the senses of bug in:
  - The bug crawled into the room.
  - The bug broke the code.

  **An unsupervised approach:** Latent Dirichlet Allocation (LDA)

  **Step 1:** choose a word and number of senses
  **Step 2:** create a corpus of sentences containing the word
  **Perform LDA, with the sentences as documents, and the word senses as topics**

Latent Dirichlet Allocation

\[ p: \text{the phrase in question. E.g. } \text{high time} \quad w_1: \text{first word of phrase} \quad w_2: \text{second word} \]
\[ D_s: \text{corpus of sentences in which word or phrase } x \text{ appears} \]
\[ K: \text{number of senses. Assume shared, fixed number for word 1, word 2, and phrase} \]

Compositional Latent Dirichlet Allocation

1. Run LDA on \( D_w \) and \( D_p \) to obtain \( \phi_{w_1} \) and \( \phi_{w_2} \)
2. Choose \( \kappa \sim \text{Beta}(\gamma, \gamma) \)
3. Choose \( \omega \sim \text{Beta}(\delta, \delta) \)
4. **for** \( i \) in range(\( K \)) **do**
5. Choose \( \phi_{x_1} \sim \text{Dirichlet}(\beta) \)
6. **end for**
7. **for** sentence \( d \) in corpus \( D_p \) **do**
8. Choose \( \lambda_d \sim \text{Bernoulli}(\kappa) \)
9. **if** \( \lambda \) **then**
10. Choose \( \theta_{d_1} \sim \text{Dirichlet}(\alpha) \)
11. Choose \( \theta_{d_2} \sim \text{Dirichlet}(\alpha) \)
12. **for** position \( j \) in \( d \) **do**
13. Choose \( w \sim \text{Bernoulli}(\omega) \)
14. Choose a sense index \( z_j \sim \text{Multinomial}(\theta_{d_1}) \)
15. Choose a word \( x_{d_2} \sim \text{Multinomial}(\phi_{x_2}) \)
16. **end for**
17. **else**
18. Choose \( \theta_{d_2} \sim \text{Dirichlet}(\alpha) \)
19. **for** position \( j \) in \( d \) **do**
20. Choose a sense index \( z_j \sim \text{Multinomial}(\theta_{d_2}) \)
21. Choose a word \( x_{d_2} \sim \text{Multinomial}(\phi_{x_2}) \)
22. **end for**
23. **end if**
24. **end for**

Corpus Collection

- Training and evaluating the model requires a corpus of common adjective-noun non-compositional phrase pairs, like black box and sour grapes
- This is obtained by constituency parsing the British National Corpus and the Corpus of Contemporary American English and using tree-regexes to match adjective-noun phrases
- This approach ensures that the expressions are of the correct syntactic type.

Implementation

- To implement our model, we use newly developed probabilistic programming language Gen, embedded in Julia.
- Probabilistic programming languages allow for probabilistic primitives.
- Gen in particular allows for easy specification of custom MCMC inference algorithms.

Further work

- Improved inference algorithm. **Possibilities:** collapsed Gibbs sampling, online Variational Bayes, special purpose algorithm
- Extending the approach to a non-parametric model.
  **Advantage:** avoids the need to prespecify a set of senses.

Audio

https://www.youtube.com/watch?v=kAbLaVxe0po