# Automated Code Extraction from Discussion Board Text Dataset

Sina Mahdipour Saravani, Sadaf Ghaffari, Yanye Luther, James Folkestad, and Marcia Moraes

sina@cs.utah.edu

marcia.moraes@colostate.edu

# Epistemic Network Analysis (ENA)

- A method for identification and quantification of connections between elements in coded data.
  - Applications in health care, educational games, and online discussion analysis.

- ENA for visualization of learning analytics for participatory Quantitative Ethnography.
  - Visualizing the connections between codes in an online discussion board.
  - Using manually coded data.

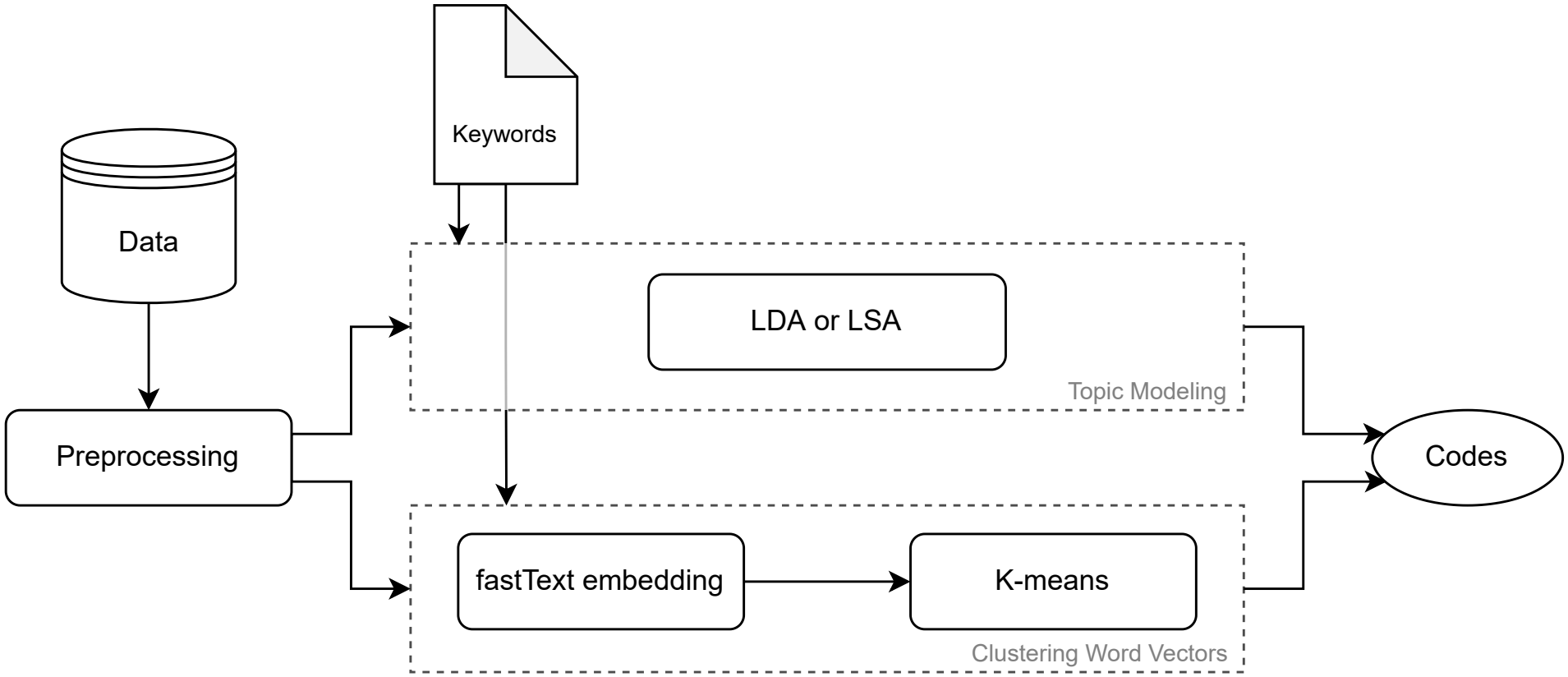
M. Moraes, J. Folkestad and K. McKenna, "Using Epistemic Network Analysis to Help Instructors Evaluate Asynchronous Online Discussions," in Second International Conference on Quantitative Ethnography: Conference Proceedings Supplement, 2021.

# ENA for Visualization in QE

- Based on feedbacks from reviewers of a teaching innovation grant committee:
  → Visualization is desirable.
  → Providing the codes is not desirable, since coding process is time consuming.
    - Even using tools such as nCoder.
  → Its easy to provide keywords that the codes should contain.

M. Moraes, J. Folkestad and K. McKenna, "Using Epistemic Network Analysis to Help Instructors Evaluate Asynchronous Online Discussions," in Second International Conference on Quantitative Ethnography: Conference Proceedings Supplement, 2021.

# Model Overview



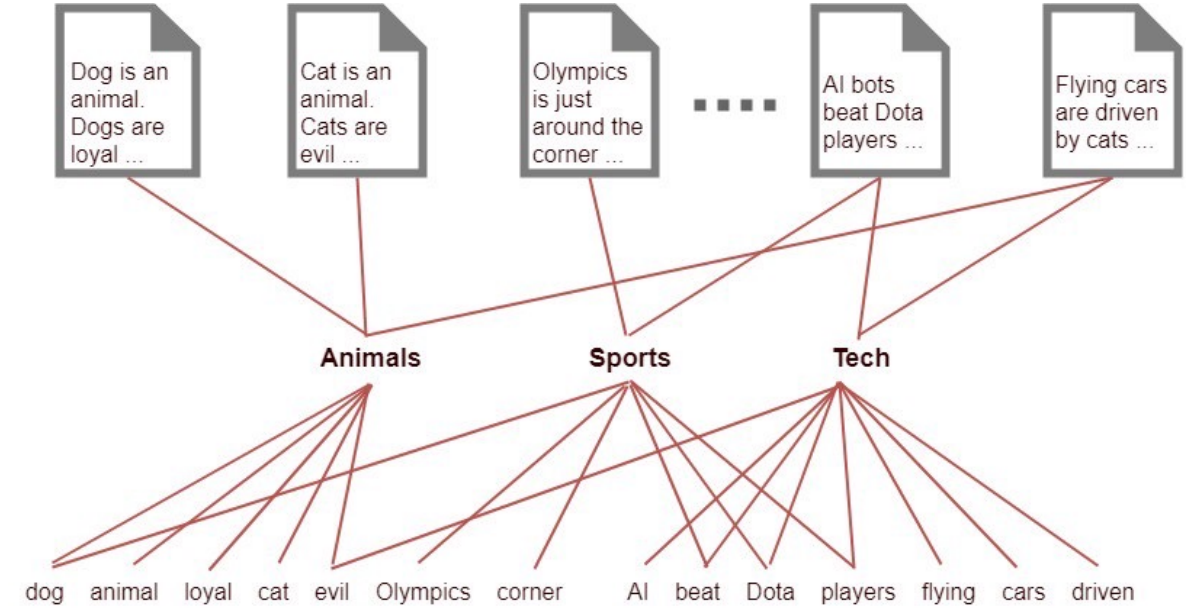
# Preprocessing

- Tokenization
- Lowercasing
- Stop word removal
- Applying minimum word length
- Irrelevant text removal
- Named-entity removal
- In-document frequency filtering
- Bigram and trigram addition

# Topic Modeling

- Latent Dirichlet Allocation:

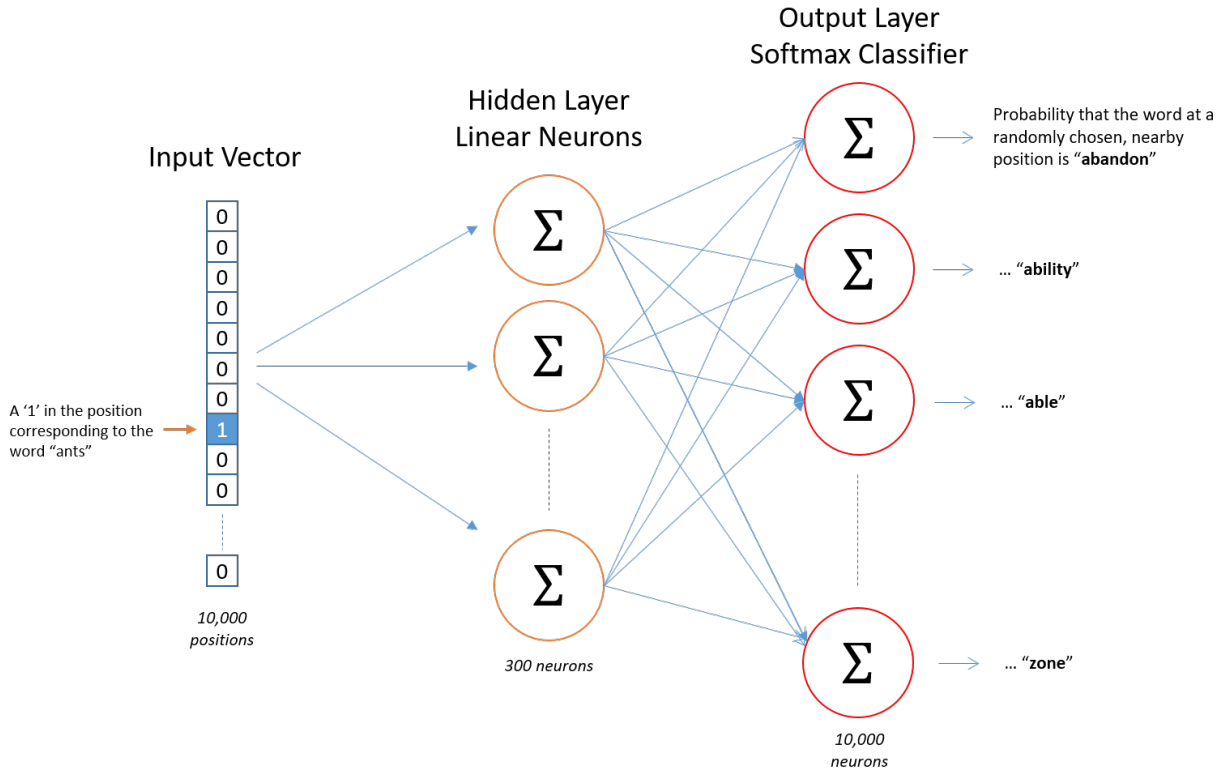
  → “Each document can be described by a distribution of topics and each topic can be described by a distribution of words”.



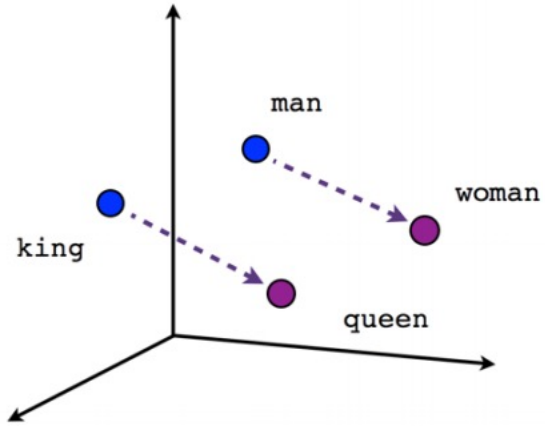

https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158

# Word Vectors (Embeddings)

- Skip-gram word2vec




$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

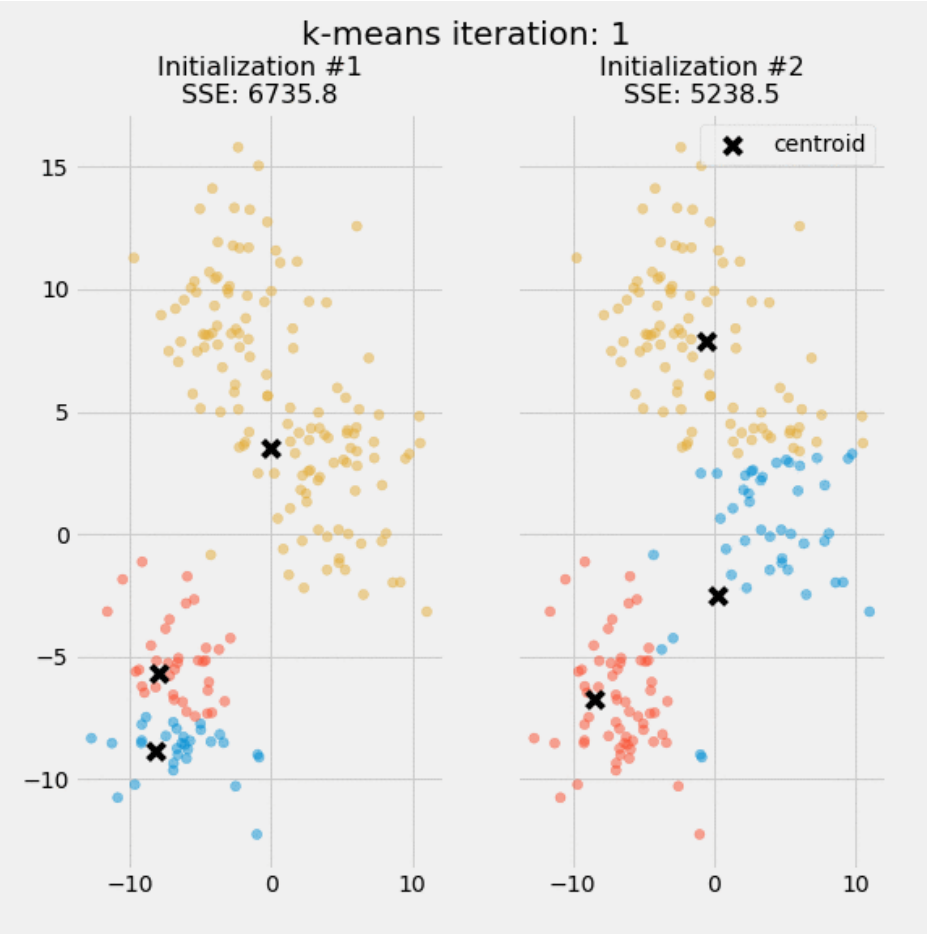



https://israelg99.github.io/2017-03-23-Word2Vec-Explained/

# K-means

**Algorithm 1** $k$-means algorithm

1: Specify the number $k$ of clusters to assign.
2: Randomly initialize $k$ centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.




https://realpython.com/k-means-clustering-python/

# Incorporating keywords

In Topic Modeling:

$$p(k_t, t) = \frac{keywords\_total\_probability}{n_{k_t}}$$

$$p(w, t) = \frac{(1 - keywords\_total\_probability)}{(n_w - n_{k_t})}$$

$$p(w, t) = \frac{1}{n_w}$$

In Clustering Word Vectors:

- Use the average of provided keywords for each code as the cluster centroid.

# Coherence Evaluation

- Gensim's coherence model from:

  *Michael Roeder, Andreas Both and Alexander Hinneburg: "Exploring the space of topic coherence measures"*

| No. Clusters | Coherence Score |
|---|---|
| 2 | 0.2851 |
| 3 | 0.2915 |
| 4 | 0.2944 |
| 5 | 0.5017 |
| 6 | 0.3776 |
| 7 | 0.4071 |
| 8 | 0.4067 |
| 9 | 0.3427 |
| 10 | 0.3773 |

# Results

LDA extracted codes

| Topic 0 | Topic 1 | Topic 2 | Topic 3 | Topic 4 |
|---|---|---|---|---|
| lecture | desire | dyslexia | confidence | mass |
| solution | desire_difficulty | learn_style | feedback | mass_practice |
| classroom | plf | individual | calibration | interleaving_practice |
| surgeon | resonate | learn_differ | confidence_memory | space_retrieval |
| acquire | parachute | disable | accuracy | tend |
| instruct | fall | intelligent | peer | day |
| learn_learn | land | prefer | answer | long_term |
| impact | jump | support | event | week |
| demand | parachute_land | dyslexia | state | myth |
| lecture_classroom | land_fall | focus | calibration_learn | practice_space |

Effortful learning code

Get beyond learning style

Illusion of mastery

Retrieval practice, Spacing out practice, and Interleaving

# Limitations

- LSA:
  - Extracted code words of a single topic often contain words from multiple manual codes.
  - Words from manual codes appear as keywords in multiple topics.
- LDA:
  - *Elaboration* code is not retrieved; however, it had the lowest kappa agreement between the human coders.
- Clustering Word Vectors:
  - Word-document information is lost as dataset is treated as a big dictionary.
  - Pretraining objective of fastText puts syntactically close words in proximity in the vector space.

# Conclusion and Future Work

- Even with small datasets, the presented method extracts many of the codes and would be a useful asset to course instructors.
- LDA performs the best among tested methods.
- Exploiting Word Vectors has a high potential due to superiority of embeddings to bag of words modeling

- Future directions:
  - Replace fastText with contextualized word embeddings such as BERT.
    → Mitigate the problem with syntax affecting the extracted codes.
    → Make use of word-document information as context affects vectors.
  - LDA2Vec:

    C. E. Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," arXiv preprint arXiv:1605.02019, 2016.

# Questions?

Thanks for your attention!