Exploiting Latent Features of Text and Graphs

Justin Sybrandt
Clemson University

Committee:
Ilya Safro
Amy Apon
Sez Atamturktur
Brian Dean
Alexander Herzog

Accomplishments: First Author Papers

- Published:
  - Moliere: Automatic Biomedical Hypothesis Generation System (KDD'17)
  - Large-scale validation of hypothesis generation systems via candidate ranking (BigData'18)
  - Are abstracts enough for hypothesis generation? (BigData'18)
- In-Submission
  - First-and High-Order Bipartite Embeddings
  - Hypergraph Partitioning with Embeddings
  - AGATHA: Automatic Graph-mining And Transformer based Hypothesis generation Approach.
  - CBAG: Conditional Biomedical Abstract Generation

Accomplishments: Co-Authored Papers

- Published:
  - Inhibition of the DDX3 prevents HIV-1 Tat and cocaine-induced neurotoxicity by targeting microglia activation. (JNP Dec. 2019)
  - Using Drive-by Health Monitoring to Detect Bridge Damage Considering Environmental and Operational Effects. (JSV Mar. 2020)
- In-Submission
  - Unsupervised Hierarchical Graph Representation Learning by Mutual Information Maximization
- Tech Reports & Submission Pending
  - To Agile, or not to Agile: A Comparison of Software Development Methodologies
  - Using BERT to Quantify Survey Responses
  - Learning GPU Memory Access Patterns

Accomplishments: Industry

- Los Alamos National Lab (Intern, Summer 2017)
- Google Pittsburgh (Intern, Summer 2018)
  - Presented at Google PhD Intern Research Conference (Only 30 presentations accepted)
- Facebook NYC (Intern, Summer 2019)
  - Intern Executive Dinner (Only 13 interns selected)

Accepted a position with Google Brain.
Starting in August at Pittsburgh office.

Exploiting Latent Features of Text and Graphs

Motivation:

Automatic Hypothesis Generation

- Goal: Predict new research
- Data sources:
  - Scientific Papers
  - Ontologies
  - Interaction Networks
- Need to find underlying trends

Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Each corresponds to first-authored publications

Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - **Agatha: deep-learning hypothesis generation**
  - **Conditional biomedical abstract generation**

New Since Proposal

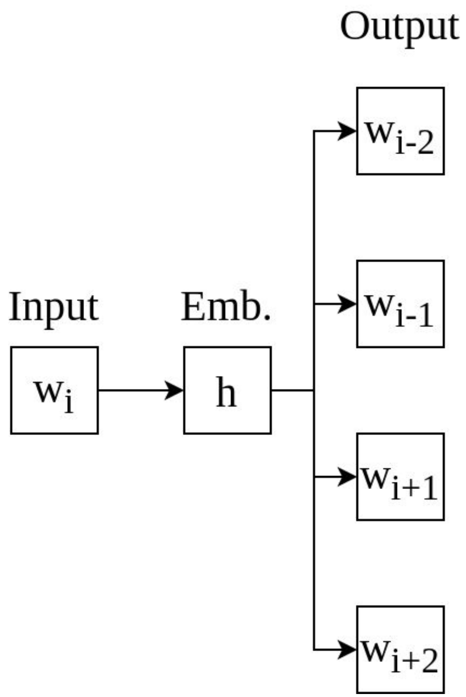
Background:

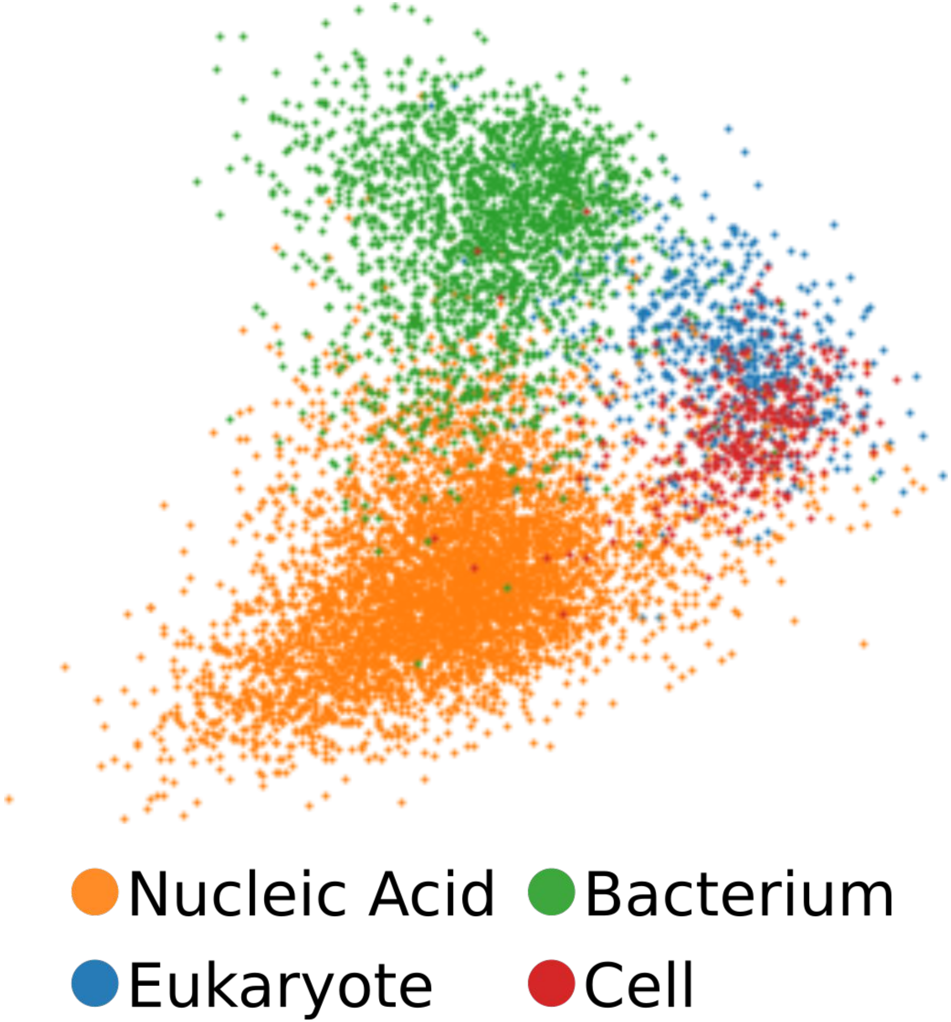
Embeddings

- Specialized statistical models:
  - Limited usability
  - Limited scope
  - Not data driven
- Embeddings:
  - Large scale
  - Wide scope
  - Data driven
  - Detects richer patterns
  - Applicable to ML

Word2Vec Text Embeddings

- Skip Gram Model
- Observe similarity:
  - Similar words share similar company
- Model Similarity:
  - Given one word, determine what others are likely to co-occur



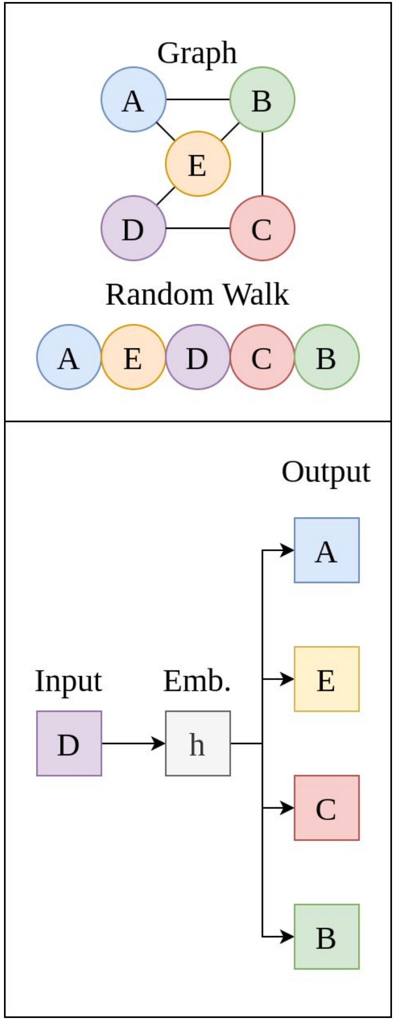

Nucleic Acid
Bacterium
Eukaryote
Cell

Contribution Summary

- **Graph Embedding**
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Graph Embedding

- DeepWalk Model
- Observe Similarity:
  - Similar nodes co-occur in random walks
- Model Similarity:
  - Given a node, determine others that are likely to co-occur



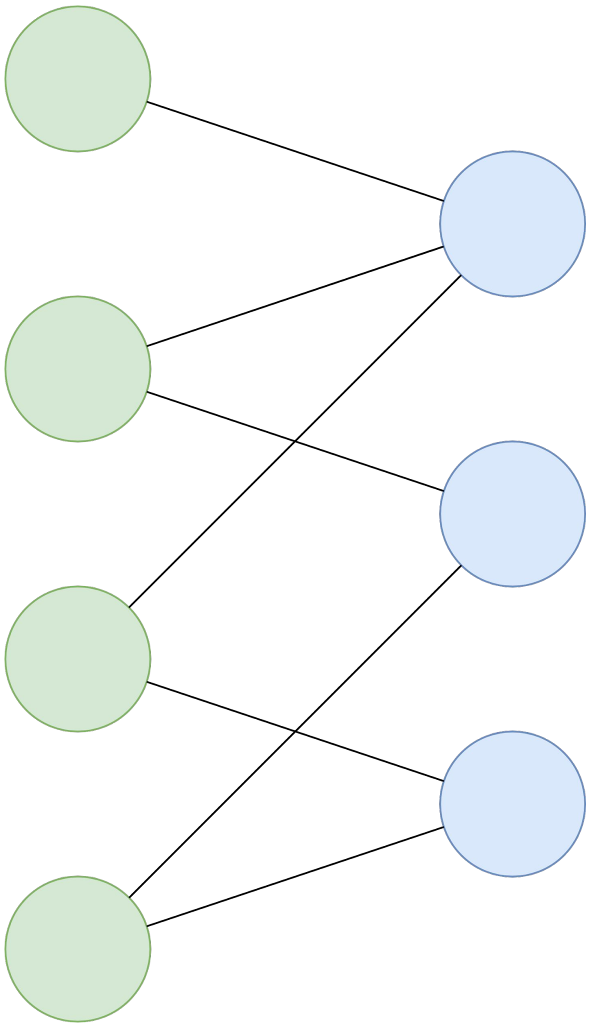
Contribution Summary

- Graph Embedding
  - **FOBE & HOBE bipartite embedding**
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

First- and High-Order Bipartite Embeddings
Sybrandt, Safro

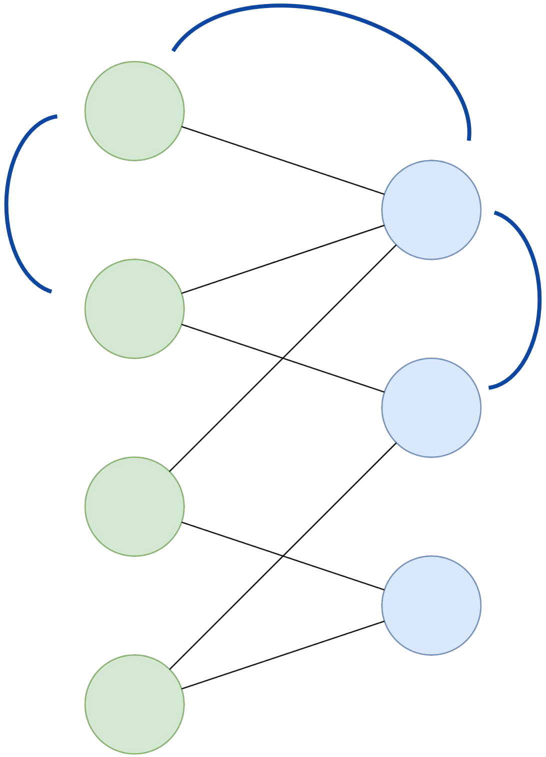
Bipartite Graphs

- $G = (V, E)$
  - $V = A \cup B$
  - $A \cap B = \emptyset$
- ε : embedding function
  - $\varepsilon : V \rightarrow R^n$
- $\Gamma(x)$: Neighborhood of x
- Typical embeddings fail to capture type-specific features



First-Order Bipartite Embedding (FOBE)

- Fast local samples
- No measurement of distant relationships



First-Order Bipartite Embedding (FOBE)

- Observations:

$$\mathbb{S}_A(\alpha_i, \alpha_j) = \begin{cases} 1 & \alpha_i, \alpha_j \in A \ \ \& \ \ \Gamma(\alpha_i) \cap \Gamma(\alpha_j) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{S}_B(\beta_i, \beta_j) = \begin{cases} 1 & \beta_i, \beta_j \in B \ \ \& \ \ \Gamma(\beta_i) \cap \Gamma(\beta_j) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{S}_V(\alpha_i, \beta_j) = \begin{cases} 1 & \alpha_i \beta_j \in E \\ 0 & \text{otherwise} \end{cases}$$

First-Order Bipartite Embedding (FOBE)

- Observations:
  - $\mathbb{S}_A, \mathbb{S}_B, \mathbb{S}_V$
- Estimations:

$$\widetilde{\mathbb{S}}_A(\alpha_i, \alpha_j) = \sigma\left(\epsilon(\alpha_i)^\intercal \epsilon(\alpha_j)\right)$$

$$\widetilde{\mathbb{S}}_B(\beta_i, \beta_j) = \sigma\left(\epsilon(\beta_i)^\intercal \epsilon(\beta_j)\right)$$

$$\widetilde{\mathbb{S}}_V(\alpha_i, \beta_j) = \mathop{\mathbb{E}}_{\alpha_k \in \Gamma(\beta_j)}\left[\widetilde{\mathbb{S}}_A(\alpha_i, \alpha_k)\right] \mathop{\mathbb{E}}_{\beta_k \in \Gamma(\alpha_i)}\left[\widetilde{\mathbb{S}}_B(\beta_j, \beta_k)\right]$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

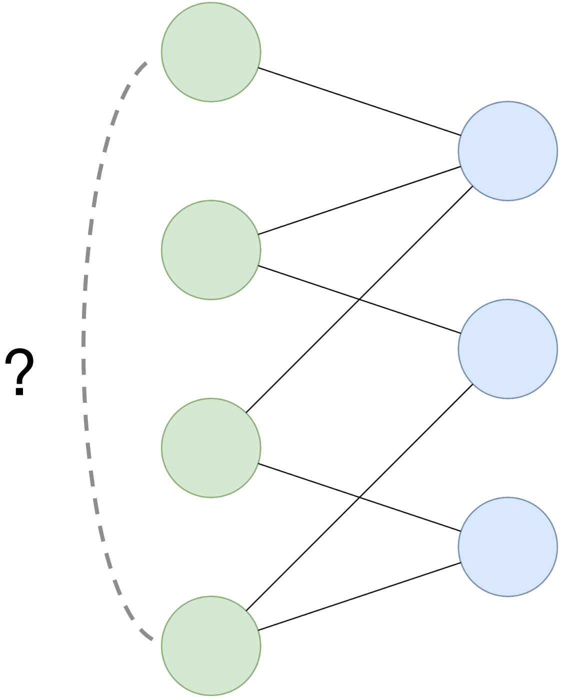
First-Order Bipartite Embedding (FOBE)

- Observations:
  - $\mathbb{S}_A, \mathbb{S}_B, \mathbb{S}_V$
- Estimations:
  - $\widetilde{\mathbb{S}}_A, \widetilde{\mathbb{S}}_B, \widetilde{\mathbb{S}}_V$
- Loss:
  - Divergence

$$\sum_{v_i, v_j \in V \times V} \left[\begin{array}{c} \widetilde{\mathbb{S}}_A(v_i, v_j) \log \left(\dfrac{\mathbb{S}_A(v_i, v_j)}{\widetilde{\mathbb{S}}_A(v_i, v_j)} \right) \\ + \widetilde{\mathbb{S}}_B(v_i, v_j) \log \left(\dfrac{\mathbb{S}_B(v_i, v_j)}{\widetilde{\mathbb{S}}_B(v_i, v_j)} \right) \\ + \widetilde{\mathbb{S}}_V(v_i, v_j) \log \left(\dfrac{\mathbb{S}_V(v_i, v_j)}{\widetilde{\mathbb{S}}_V(v_i, v_j)} \right) \end{array} \right]$$

Higher-Order Bipartite Embedding (HOBE)

- Emphasizes more-distant relationships
- Similarities approximated by algebraic distance

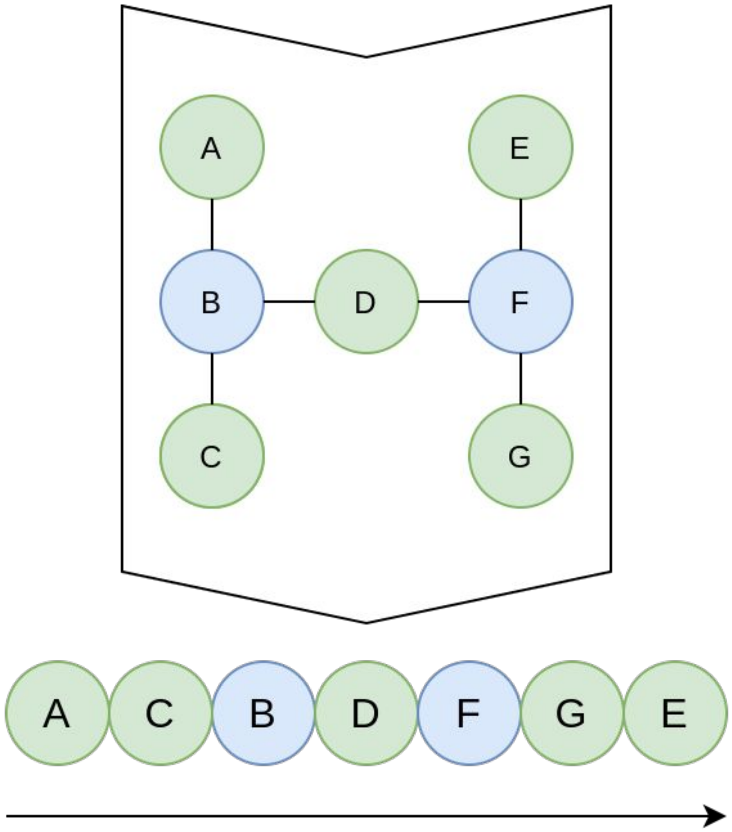


Algebraic Distance

- Assign nodes coordinates on unit interval
- Iterative process
- Fast to compute
- Run R trials
- Similarity measure:

$$d(v_i, v_j) = \sqrt{\sum_{r=1}^{R} \left(\mathbf{a}_r^{(K)}(v_i) - \mathbf{a}_r^{(K)}(v_j) \right)^2}$$

$$s(v_i, v_j) = \frac{\sqrt{R} - d(v_i, v_j)}{\sqrt{R}}$$



High-Order Bipartite Embedding (HOBE)

- Observations:

$$\mathbb{S}'_A(\alpha_i, \alpha_j) = \begin{cases} \max\limits_{\beta_k \in \Gamma(\alpha_i) \cap \Gamma(\alpha_j)} \min\left(s(\alpha_i, \beta_k), s(\alpha_j, \beta_k)\right) \\ \quad \text{if } \alpha_i, \alpha_j \in A \\ 0 \text{ otherwise} \end{cases}$$

$$\mathbb{S}'_B(\beta_i, \beta_j) = \begin{cases} \max\limits_{\alpha_k \in \Gamma(\beta_i) \cap \Gamma(\beta_j)} \min\left(s(\alpha_k, \beta_i), s(\alpha_k, \beta_j)\right) \\ \quad \text{if } \beta_i, \beta_j \in B \\ 0 \text{ otherwise} \end{cases}$$

High-Order Bipartite Embedding (HOBE)

- Observations:

$$\mathbb{S}_V^{'}(\alpha_i, \beta_j) = \max \begin{pmatrix} \max\limits_{\alpha_k \in \Gamma(\beta_j)} \mathbb{S}_A^{'}(\alpha_i, \alpha_k), \\ \max\limits_{\beta_k \in \Gamma(\alpha_i)} \mathbb{S}_B^{'}(\beta_j, \beta_k) \end{pmatrix}$$

High-Order Bipartite Embedding (HOBE)

- Observations:
  - $\mathbb{S}_A^{'}, \mathbb{S}_B^{'}, \mathbb{S}_V^{'}$
- Estimations:

$$\widetilde{\mathbb{S}}_A^{'}(\alpha_i, \alpha_j) = \max\left(0, \epsilon(\alpha_i)^\intercal \epsilon(\alpha_j)\right)$$

$$\widetilde{\mathbb{S}}_B^{'}(\beta_i, \beta_j) = \max\left(0, \epsilon(\beta_i)^\intercal \epsilon(\beta_j)\right)$$

$$\widetilde{\mathbb{S}}_V^{'}(\alpha_i, \beta_j) = \mathop{\mathbb{E}}_{\alpha_k \in \Gamma(\beta_j)}\left[\widetilde{\mathbb{S}}_A^{'}(\alpha_i, \alpha_k)\right] \mathop{\mathbb{E}}_{\beta_k \in \Gamma(\alpha_i)}\left[\widetilde{\mathbb{S}}_B^{'}(\beta_j, \beta_k)\right]$$

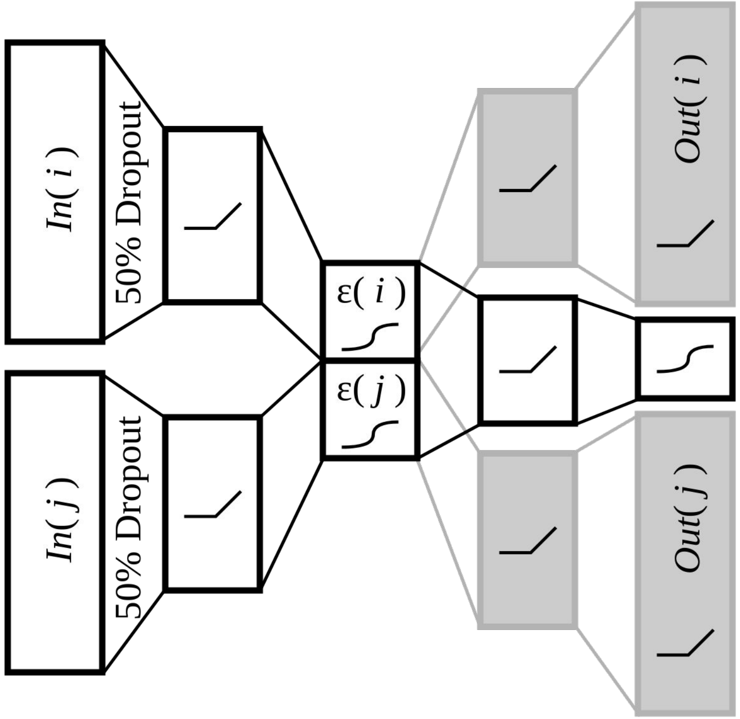
High-Order Bipartite Embedding (HOBE)

- Observations:
    - $\mathbb{S}_A^{'}, \mathbb{S}_B^{'}, \mathbb{S}_V^{'}$
- Estimations:
    - $\widetilde{\mathbb{S}}_A^{'}, \widetilde{\mathbb{S}}_B^{'}, \widetilde{\mathbb{S}}_V^{'}$
- Loss:
    - Mean Squared Error

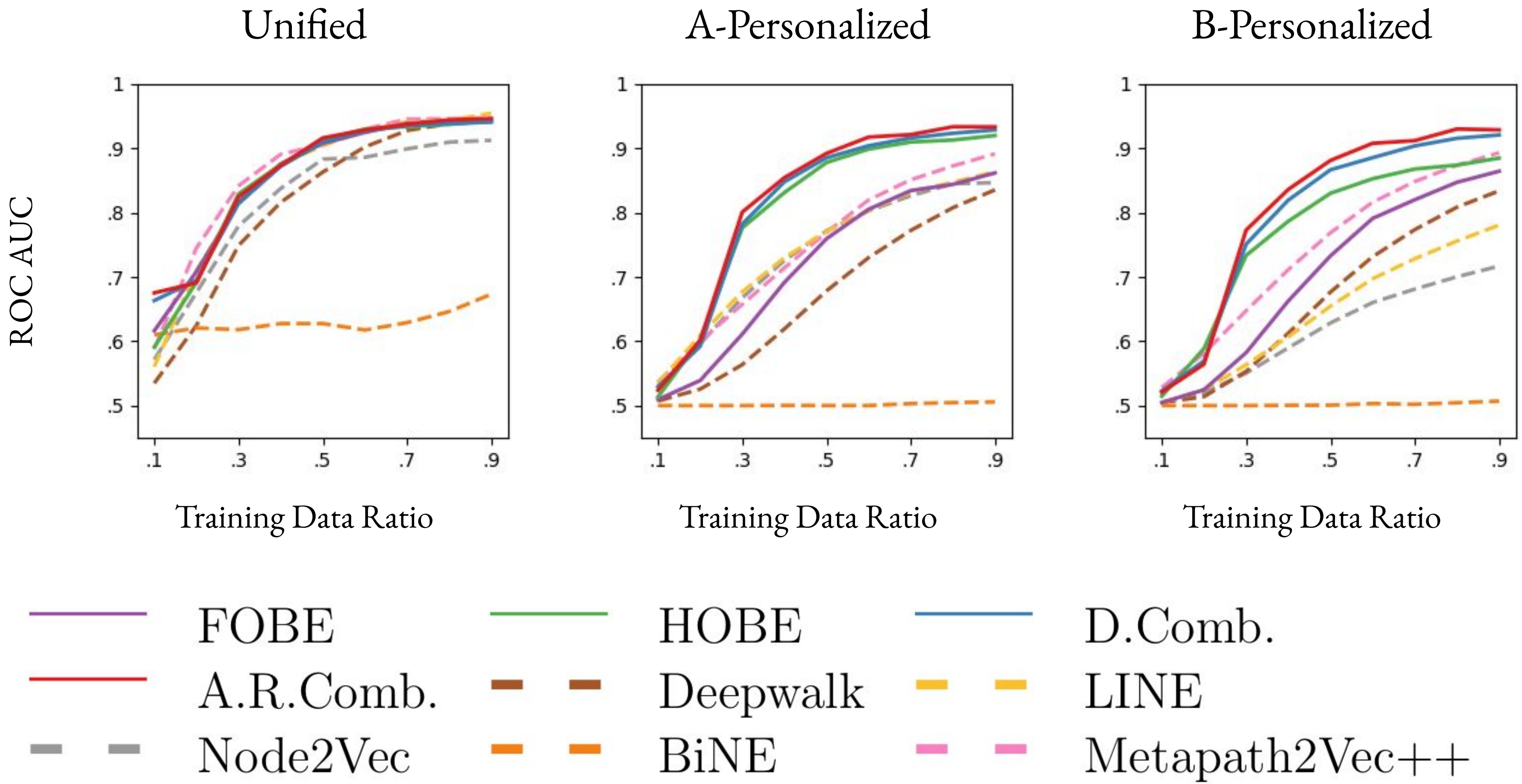
$$\sum_{v_i, v_j \in V \times V} \begin{bmatrix} (\mathbb{S}_A^{'}(v_i, v_j) - \widetilde{\mathbb{S}}_A^{'}(v_i, v_j))^2 \\ +(\mathbb{S}_B^{'}(v_i, v_j) - \widetilde{\mathbb{S}}_B^{'}(v_i, v_j))^2 \\ +(\mathbb{S}_V^{'}(v_i, v_j) - \widetilde{\mathbb{S}}_V^{'}(v_i, v_j))^2 \end{bmatrix}$$

Combination Embedding

- Merge pretrained embeddings
- Autoencoder combined with edge prediction
- Combines redundant signals
- Boosts distinct signals



Link Prediction Results : MadGrades Network



Recommendation Results

Metric@10:	F1	NDCG	MAP	MRR
DeepWalk	.0850	.2414	.1971	.3153
LINE	.0899	.1441	.0962	.1713
Node2Vec	.0854	.2389	.1944	.3111
MP2V++	.0865	.2514	.1906	.3197
BINE	**.1137**	.2619	.2047	.3336
FOBE	.1108	.3771	.2382	.4491
HOBE	.1003	**.4054**	**.3156**	**.6276**
D.Comb.	.0753	.2973	.2362	.5996
A.R.Comb.	.0667	.2359	.1730	.5080

DBLP

Metric@10:	F1	NDCG	MAP	MRR
DeepWalk	.0027	.0153	.0069	.1844
LINE	.0067	.0435	.0229	.2477
Node2Vec	.0279	.1261	.0645	.2047
MP2V++	.0024	.0153	.0088	.2677
BINE	.0227	.1551	.0982	.3539
FOBE	**.0729**	**.3085**	**.1997**	.3778
HOBE	.0195	.1352	.0789	.3400
D.Comb.	.0243	.1285	.0795	.3520
A.R.Comb.	.0388	.1927	.1249	**.3915**

LastFM

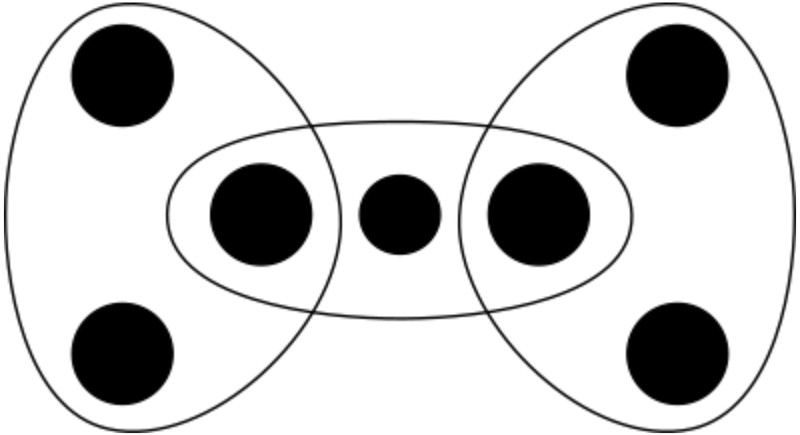
Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - **Embedding-based coarsening for hypergraph partitioning**
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

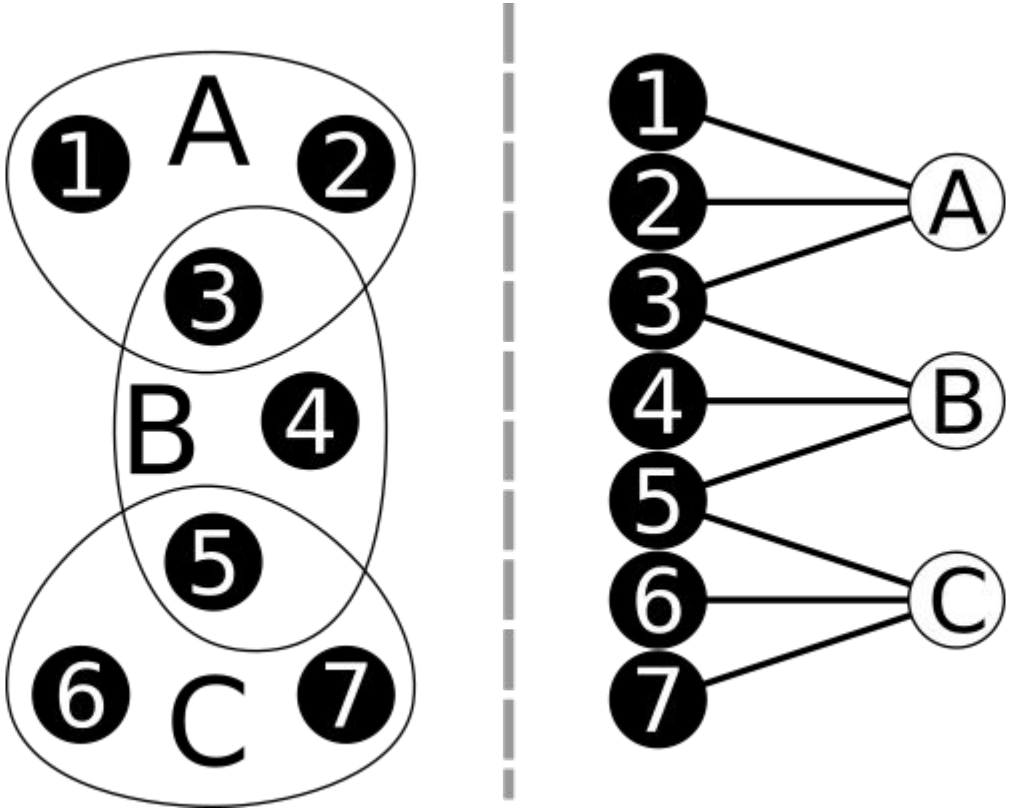
Partition Hypergraphs with Embeddings
Sybrandt, Shaydulin, Safro

Hypergraphs

- Generalization of graphs
- Hyperedges may contain any subset of nodes
- $H = (V, E)$

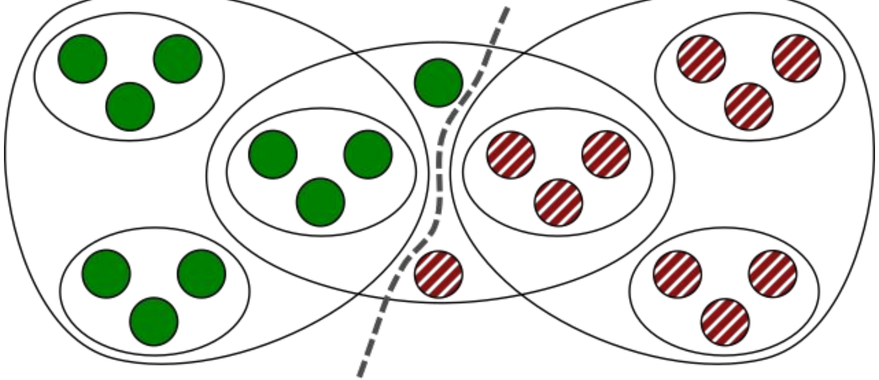


Hypergraphs are Bipartite Graphs



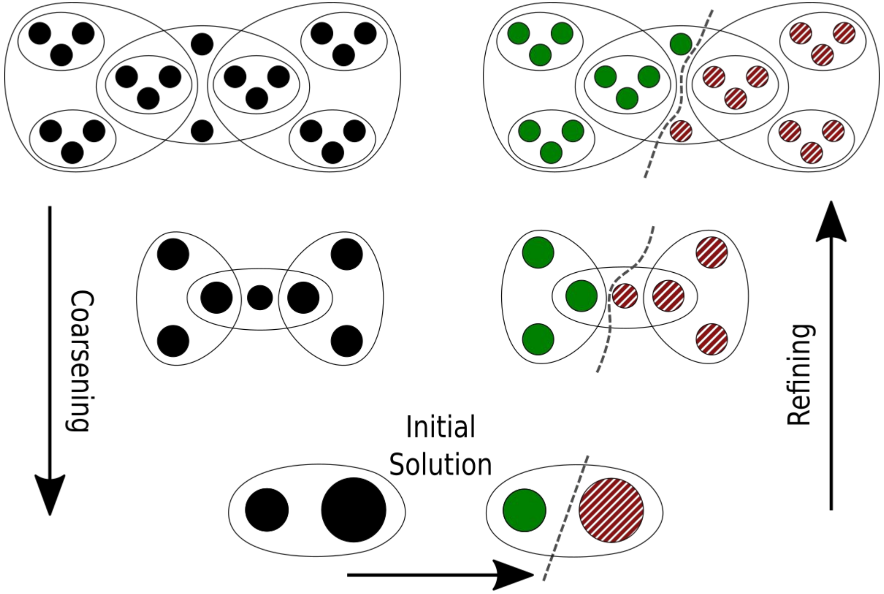
Hypergraph Partitioning

- Divide nodes into similarly-sized parts
- Minimize:
  - Cut hyperedges
  - Connectivity of cut hyperedges
- NP Hard
  - To solve
  - To approximate



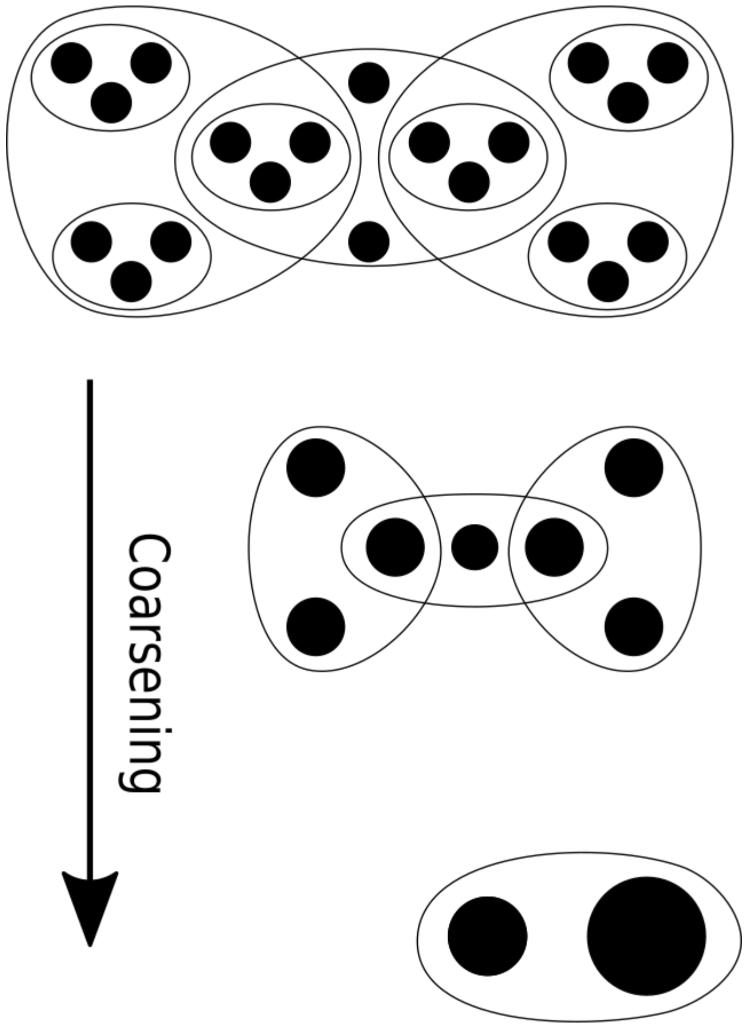
Multilevel Partitioning

- Best solution strategy
- Steps:
  - Coarsening
  - Initial Solution
  - Uncoarsening
    - Expansion
    - Interpolation
    - Refinement
- Paradigms:
  - n-Level
  - (log n)-Level



Coarsening

- Goals:
  - *Contract* similar nodes
  - Retain global structural features
  - Reduce hypergraph size
- Pattern:
  - Visit a node
  - Find a similar neighbor
  - Merge pairs
- Weights:
  - Nodes / edges start with $w = 1$
  - Contracted nodes / edges sum weights



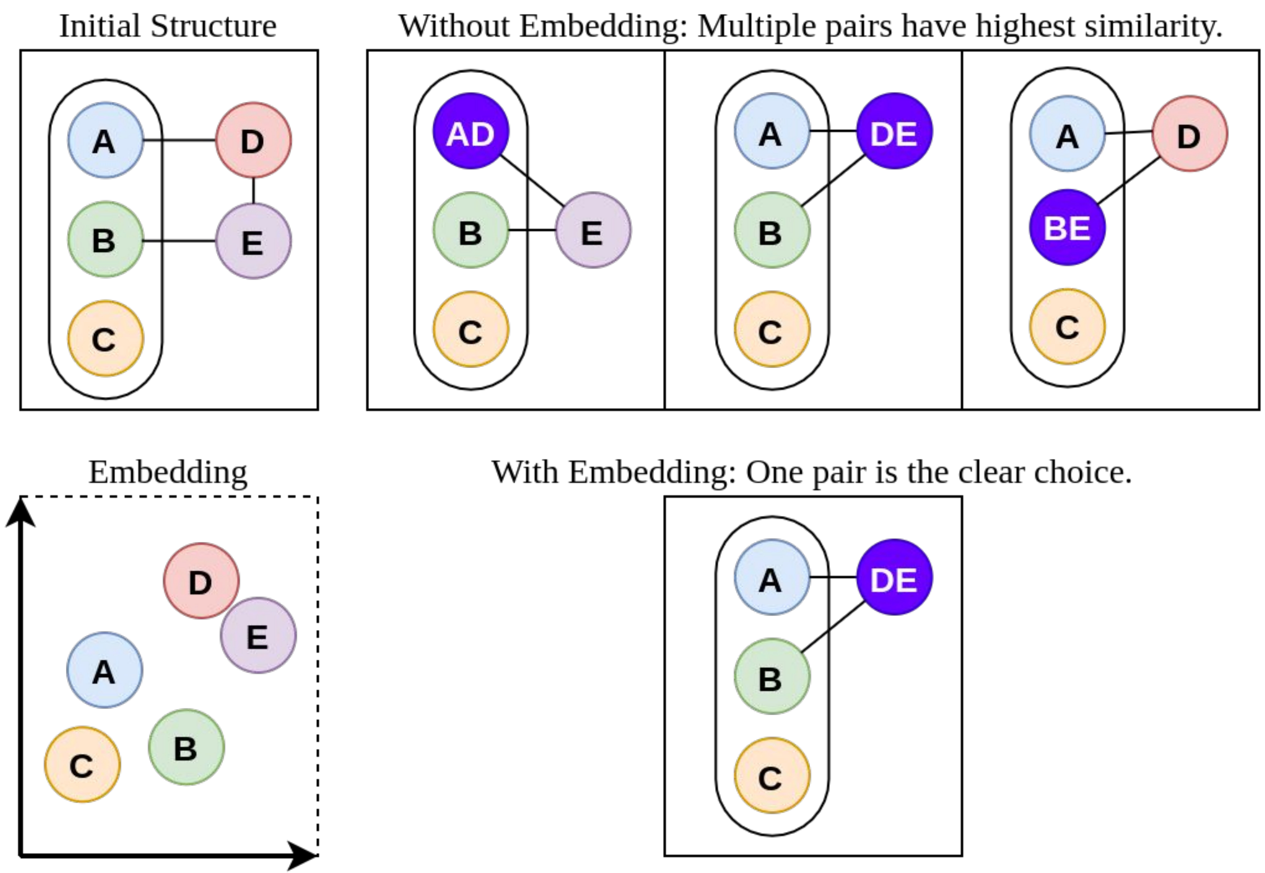
Heavy Edge Coarsening

- Visit nodes randomly
- Compare node pairs by shared edges

$$S_E(u, v) = \sum_{u,v \in e \in E} \frac{w_e}{|e| - 1}$$

- Only considers local information

Motivation for Embedding-Based Coarsening



Embedding-Based Coarsening

- Establishes a fixed visit order
  - Prioritizes node pairs that share embedding-based features

$$S_O(u) = \max_{v \in \Gamma(u), u \neq v} \frac{\epsilon(u)^\intercal \epsilon(v)}{w_u w_v}$$

Embedding-Based Coarsening

- Establishes a fixed visit order
  - Prioritizes node pairs that share embedding-based features

$$S_O(u) = \max_{v \in \Gamma(u), u \neq v} \frac{\epsilon(u)^\intercal \epsilon(v)}{w_u w_v}$$

- Scores partners with embeddings:

$$S_\epsilon(u, v) = \left(\frac{\epsilon(u)^\intercal \epsilon(v)}{w_u w_v} \right) \left(\sum_{e \in \Gamma(u) \cap \Gamma(v)} \frac{w_e}{|e| - 1} \right)$$

Embeddings for Newly Contracted Nodes

- Coarse nodes need embeddings
- Average embeddings of existing nodes:
- If *u* is a coarse node that contains input nodes *v*:

$$\epsilon(u) = \frac{1}{w_u} \sum_{i=0}^{w_u} \epsilon(v_i)$$

Effects on Runtime

- Embedding:
    - One-time cost
    - Varies by method
- Ordering:
    - Assign scores per-node at each level
    - Can reuse previous level's scores
    - Symmetric comparison between all neighborhood pairs [$O(n^2)$]

Considered Implementations

- Proposed Implementations:
  - KaHyPar: embedding-based coarsening
  - KaHyPar: embedding-based coarsening & flow-based refinement
  - Zoltan: embedding-based coarsening
- Baseline:
  - KaHyPar: community-based coarsening
  - KaHyPar: community-based coarsening & flow-based refinement
  - Zoltan
  - PaToH
  - hMetis

Partitioning Benchmark

- 96 Hypergraphs
  - Sparse matrix collection
- 7 Partition counts
  - k = 2, 4, 8, 16, 32, 64, 128
- 6 embeddings
  - FOBE & HOBE
  - Node2Vec
  - Metapath2Vec++
  - FOBE+HOBE Combination
  - All 4 Combination
- 2 Partitioning Objectives:
  - cut
  - connectivity (k-1)
- 20 trials per combination
- Metrics:
  - Macro-mean
  - Macro-min
  - Macro-max
  - Macro-std
- **Over 500,000 individual trials**

Results: Direct Improvement

- Compare embedding-based implementation to corresponding baseline

Average connectivity improvement.

# Parts(k):	2	4	8	16	32	64	128
KaHyPar	8%	13%	10%	6%	4%	3%	1%
KaHyPar(flow)	9%	11%	4%	2%	3%	2%	0%
Zoltan	48%	28%	15%	14%	9%	5%	3%

Average cut improvement.

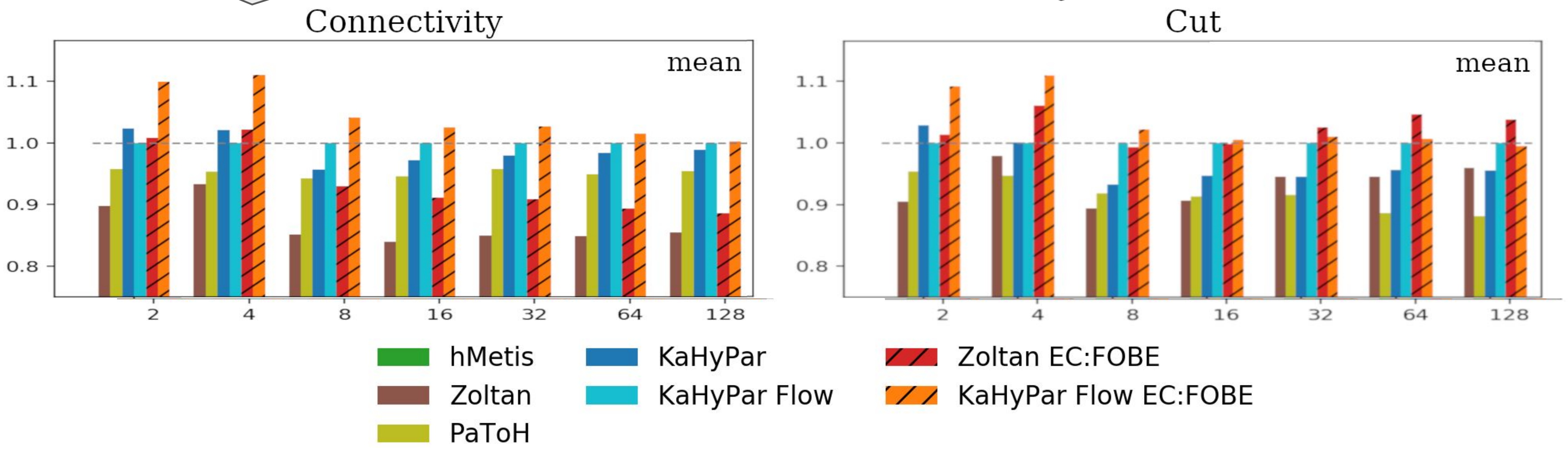
# Parts(k):	2	4	8	16	32	64	128
KaHyPar	8%	16%	9%	1%	3%	1%	0%
KaHyPar(flow)	10%	11%	3%	1%	1%	1%	-1%
Zoltan	51%	45%	51%	41%	31%	14%	8%

Greatest improvement for smaller # of partitions

Results: Average Improvement w.r.t. KaHyPar Flow

All methods compared to KaHyPar with flow-based refinement

Zoltan with embedding-based coarsening outperforms KaHyPar



Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- **Automatic Hypothesis Generation**
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Motivation: Drug Discovery

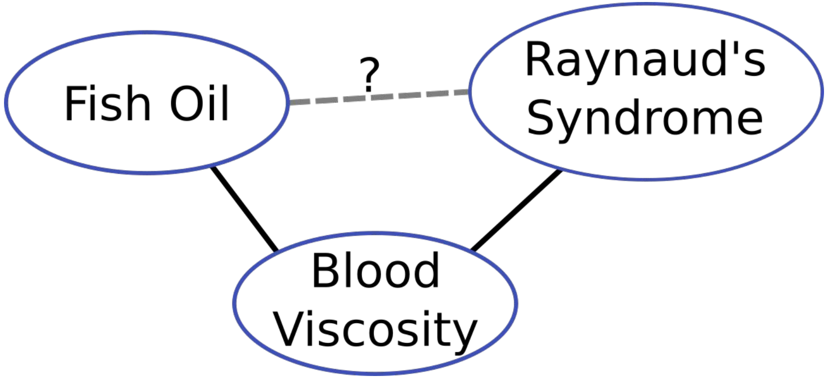
- Steps:
  - Select disease to treat
  - Identify ~1000 target substances
  - Determine ~10 candidates
  - Prioritize investment
  - Conduct ~1 human trial
  - Go to market
- Want to give information:
  - To decision makers
  - Earlier in the process
  - Cheaply

Wealth of Available Data

- National Library of Medicine provides public databases
- MEDLINE contains nearly 30 million biomedical abstracts
- Data available through PubMed
- New papers per-year is increasing!
  - Nearly 1 million last year

The ABC Model

- Hypothesis Generation:
  - Identify *implicitly* available knowledge
- Pattern:
  - Given two terms: A, C
  - Find words related to A
  - Find words related to C
  - Find overlap
- Key Limitations:
  - Only simple connections
  - Biased to incremental results



Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - **Moliere: hypothesis generation via topic modeling**
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Moliere: Automatic biomedical hypothesis generation system
Sybrandt, Shtutman, Safro

KDD'17

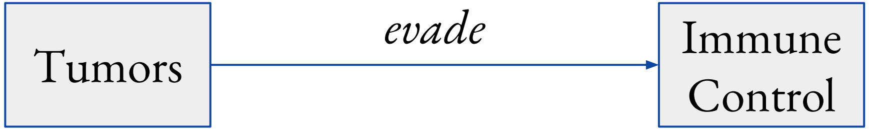
Moliere

- Preprocessing
  - Data collection
  - Semantic network
- Querying
  - Shortest paths
  - Topic models
- Analysis
  - Word distributions
  - Small-scale results

Data Collection

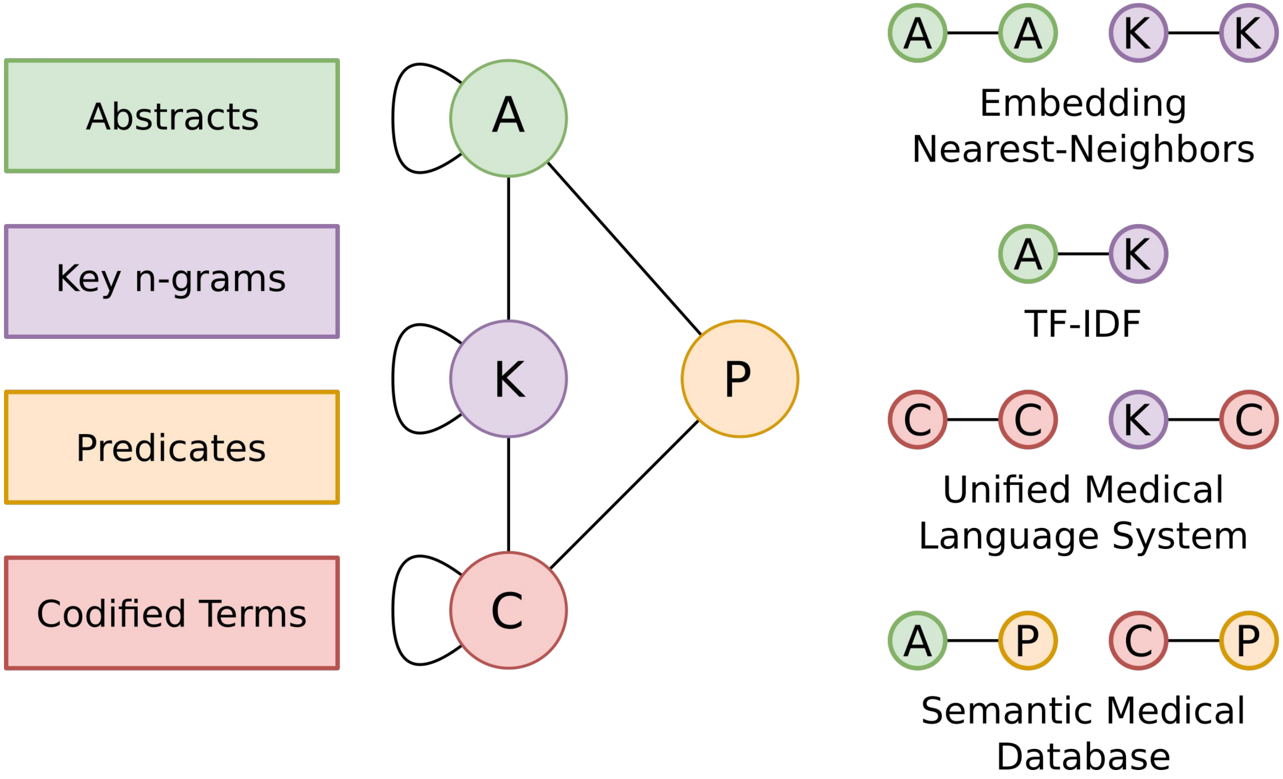
- Original Text
  - Titles
  - Abstracts
- Phrases (n-Grams)
  - Collected by ToPMine
- Predicates
  - Subject, verb, object statements
- Coded Terms
  - Unified Medical Language System (UMLS)

Tumours evade immune control by creating hostile microenvironments that perturb T cell metabolism and effector function.

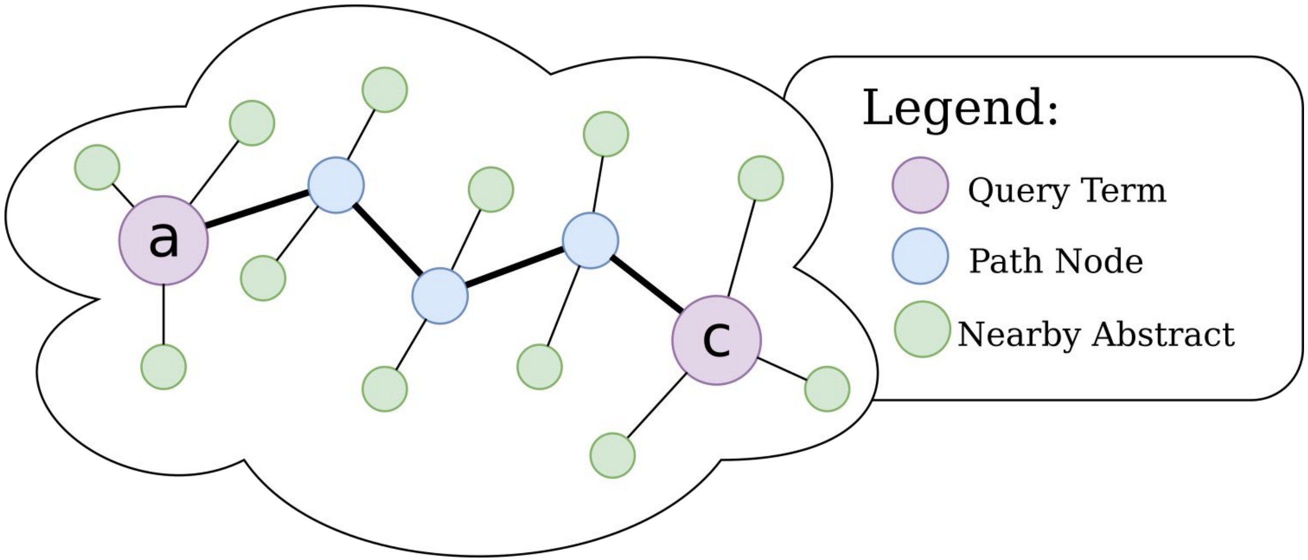


Neoplasm:
- Tumor
- Tumour
- Oncological Abnormality

Semantic Network

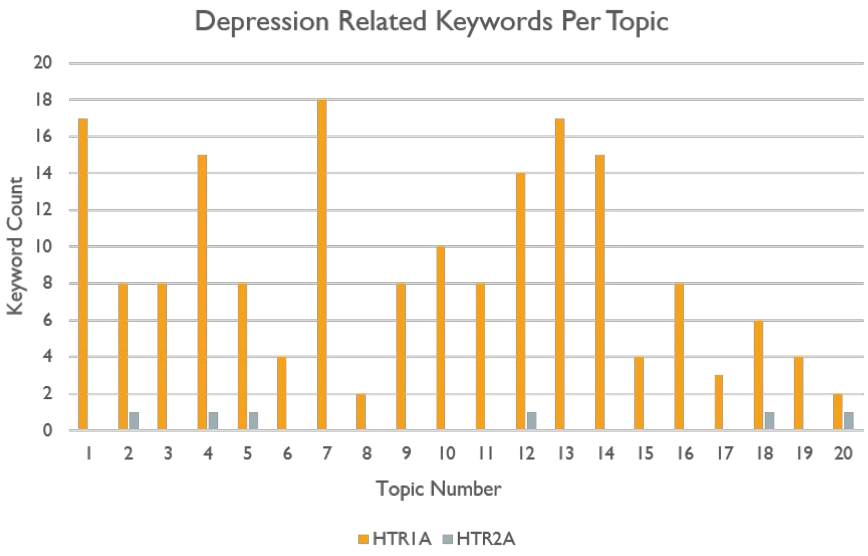


Query Shortest Paths



Topic Modeling

- Cluster words in select abstracts
- Explore trends
- Rely on expert analysis
- Example:
  - Venlafaxine - HTR1A
  - Venlafaxine - HTR2A



Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - **Validation of hypothesis generation via candidate ranking**
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Large-scale validation of hypothesis generation systems via candidate ranking
Sybrandt, Shtutman, Safro

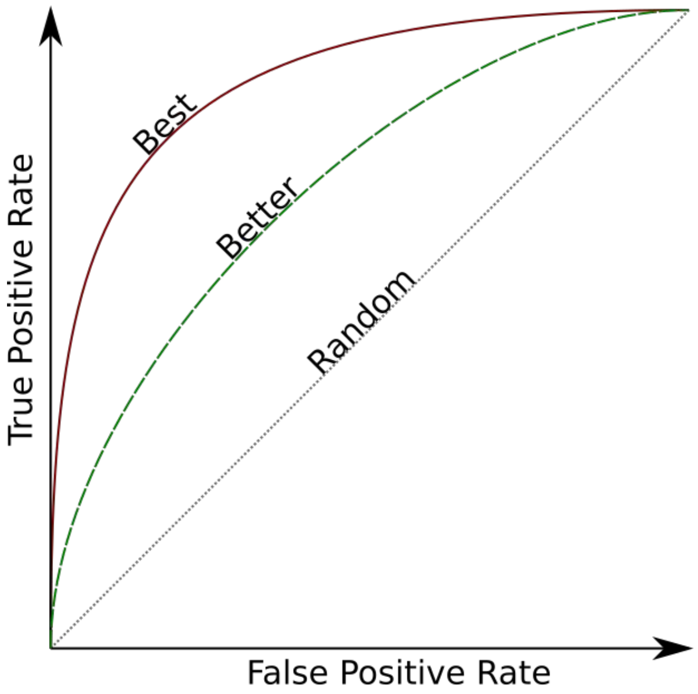
BigData'18

Existing Validation

- Methods:
  - Recreate 7 experiments from early 90's
  - Domain-specific statistics
  - Expert interpretation
  - Publish in medicine
- Complications:
  - Too narrow: Only specific domains
  - Too slow: Human in the loop
  - Too small: Datasets of <10 hypotheses

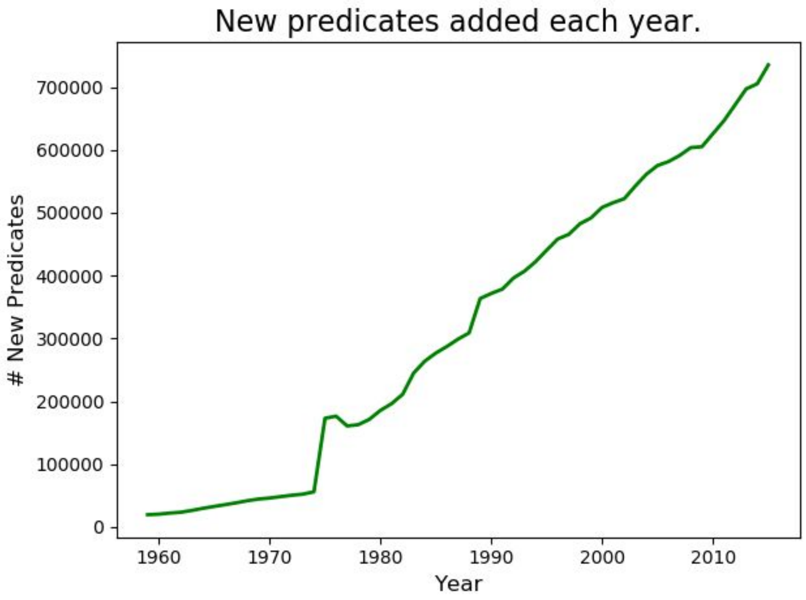
Validation via Ranking

- Drug discovery is ranking
- Requires only a ranking criteria
- Standard metrics
  - ROC
  - PR
  - Recommender Metrics
- Requires:
  - Positive and negative samples



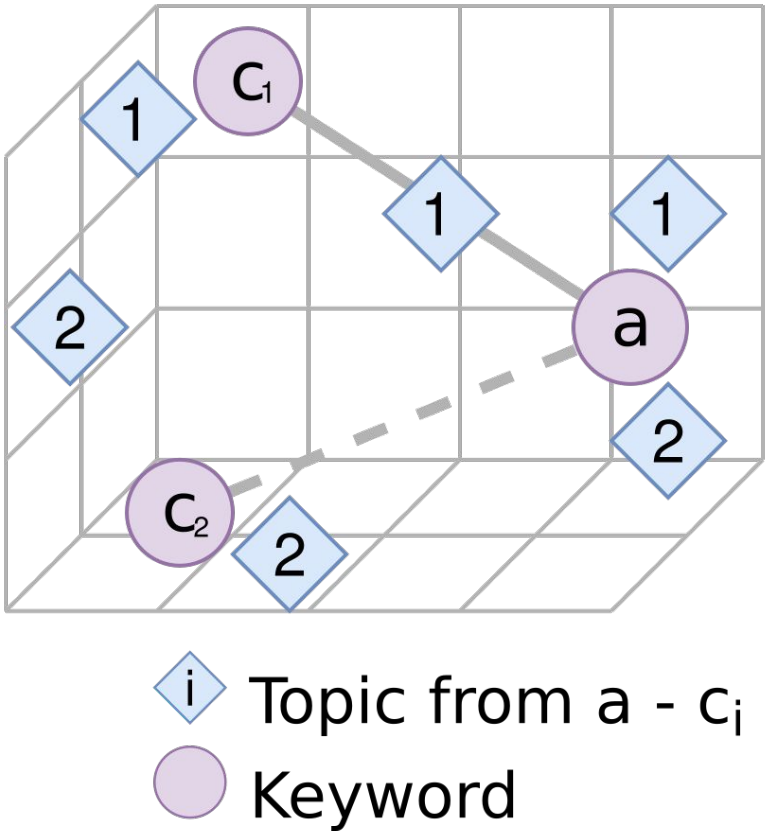
Validation Dataset

- Positive Samples: Predicates
  - New predicates added every year
  - Perform temporal holdout
- Negative Samples: Random
  - Generate pairs of terms
  - Select unpublished pairs
- Strengths:
  - Simple
  - Scalable
- Weaknesses:
  - Class balance
  - Distribution of neg. terms



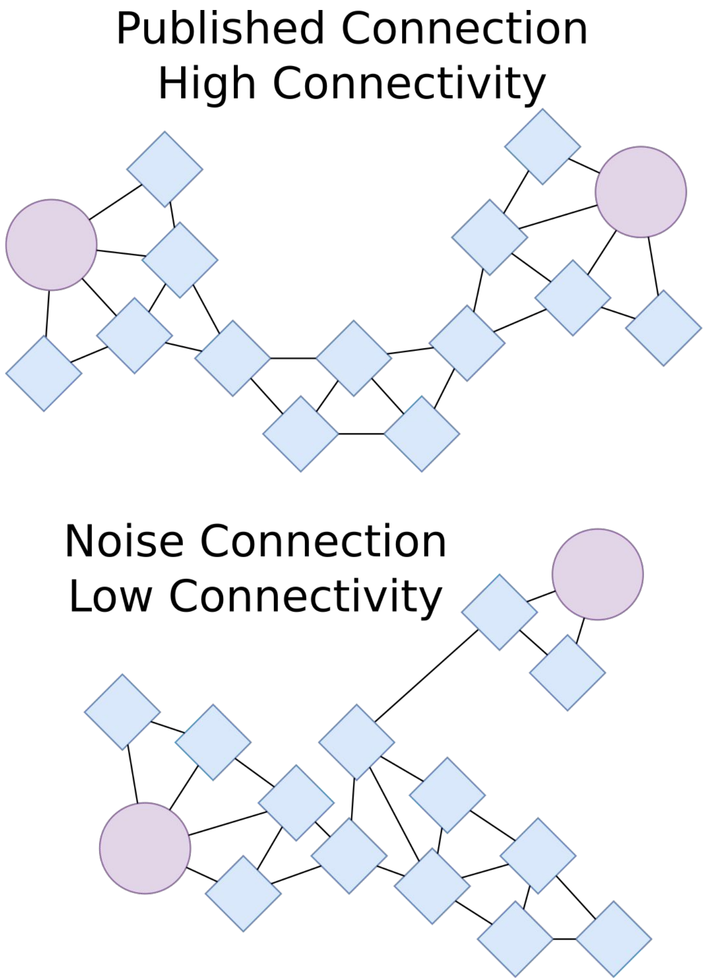
Embedding Measures

- Heuristic principles:
  - Similar keywords
  - Shared similar topic
  - Topic distance correlation
- Measured with:
  - Cosine Similarity
  - Euclidean Distance



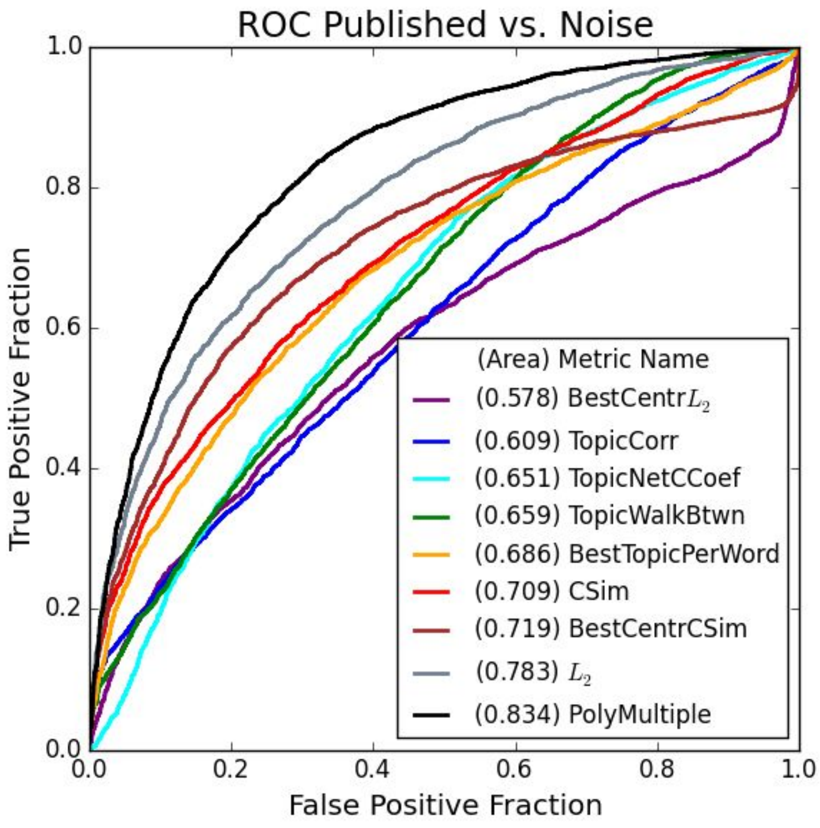
Topic Network Measures

- Heuristic principles:
  - Connectivity
  - Clustering
  - Shortest Path
- Measured:
  - Path length
  - Path betweenness
  - Centralities
  - Modularity



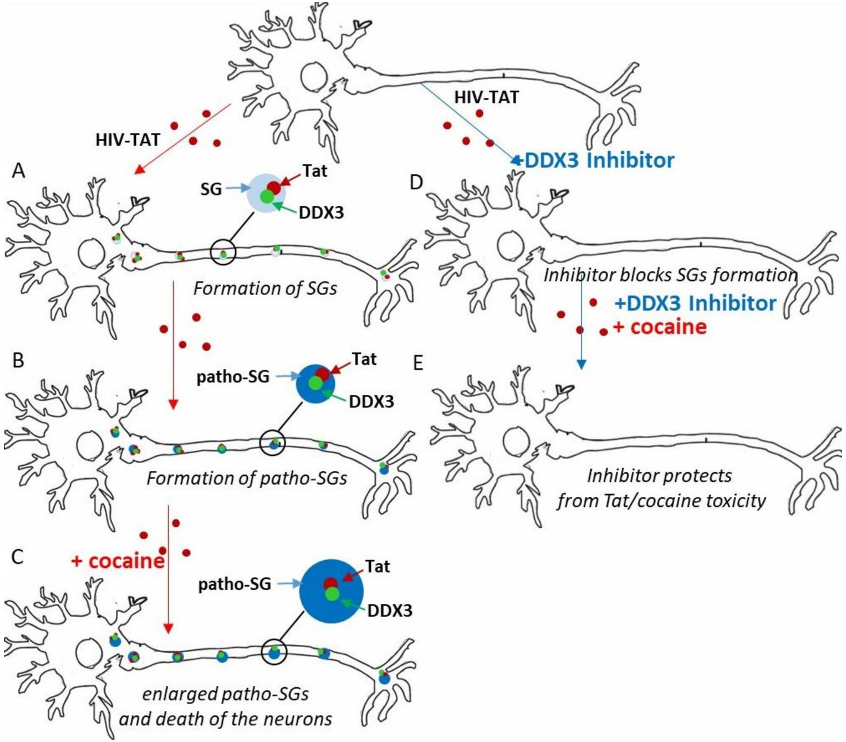
Validation Results

- Cut year: 2010
- 8,638 total queries
  - half positive, half negative
- Important measures:
  - Polynomial
  - Euclidean distance of keywords
  - Cosine similarity of shared topic



Real World Application

- Validation does not tell us how Moliere performs in reality
- Moliere ranked 40,000 genes
- DDX3 ranked highly
- Confirmed in laboratory



Inhibition of the DDX3 prevents HIV-1 Tat and cocaine-induced neurotoxicity by targeting microglia activation
Aksenovam, Sybrandt, Chu, Sikirzhytski, Ji, Odhiambo, Lucius, Turner, Broude, Pena, Lizzaraga, Zhu, Safro, Wyatt, Shtutman

JNP'19

Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - **Evaluation of corpora on generated hypotheses**
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Are abstracts enough for hypothesis generation?
Sybrandt, Carrabba, Herzog, Safro

BigData'18

Pros and Cons of Full Text

- Pros:
    - Contains greater detail
    - Contains auxiliary information
- Cons:
    - Challenging to parse PDFs
    - Noisy
    - Expensive
        - Computationally
        - Financially

PubMed Central (PMC)

- Publically available full text papers
- Limited in scope
- Only recent papers
- Plain text release

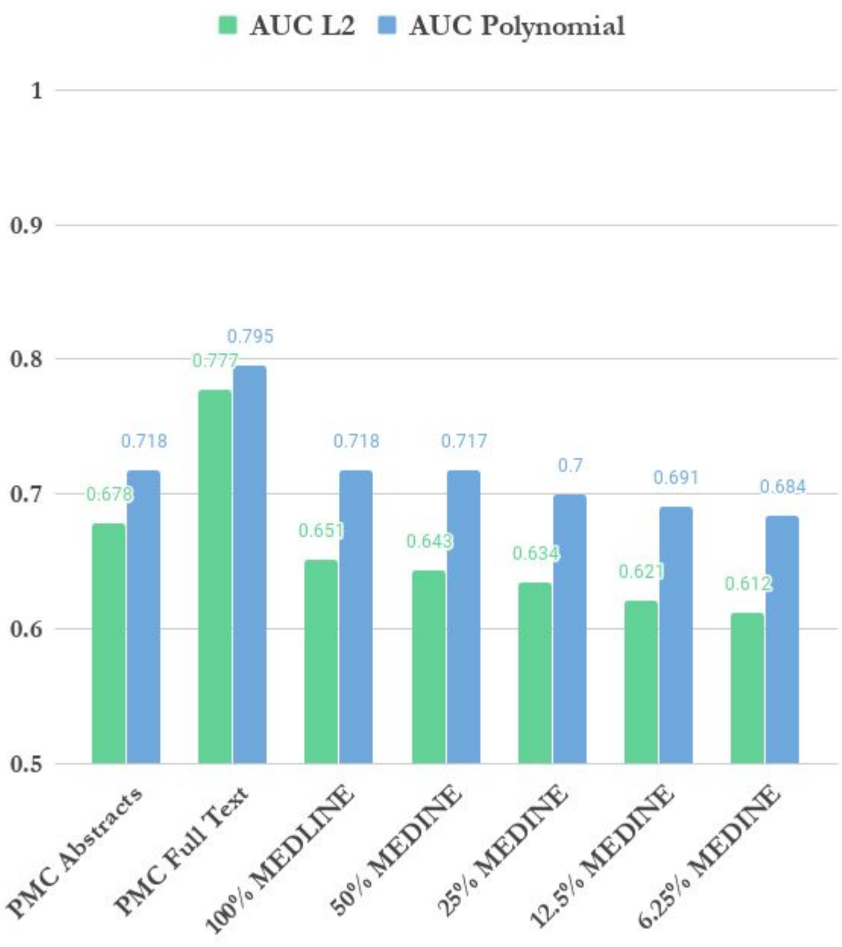
Experimental Setup

- Retrain multiple instances of Moliere
- Use different subsets of MEDLINE and PubMed Central
- Perform validation on the same set of predicates

Corpus	Total Words	Unique Words	Corpus Size	Median Words per Document
PMC Abstracts	109,987,863	673,389	1,086,704	102
PMC Full-Text	1,860,907,606	6,548,236	1,086,704	1594
MEDLINE	1,852,059,044	2,410,130	24,284,910	71
1/2 MEDLINE	923,679,660	1,505,672	12,142,455	71
1/4 MEDLINE	460,384,928	920,734	6,071,227	71
1/8 MEDLINE	229,452,214	565,270	3,035,613	71
1/16 MEDLINE	114,385,607	349,174	1,517,806	71

Validation Results

- Cut date: 2015
- Full text increase performance by 10%
- Euclidean distance of word embeddings most valuable for full text
- Full text has less benefit from topic models
- Full text takes **45x** longer to perform one query



Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - **Agatha: deep-learning hypothesis generation**
  - Conditional biomedical abstract generation

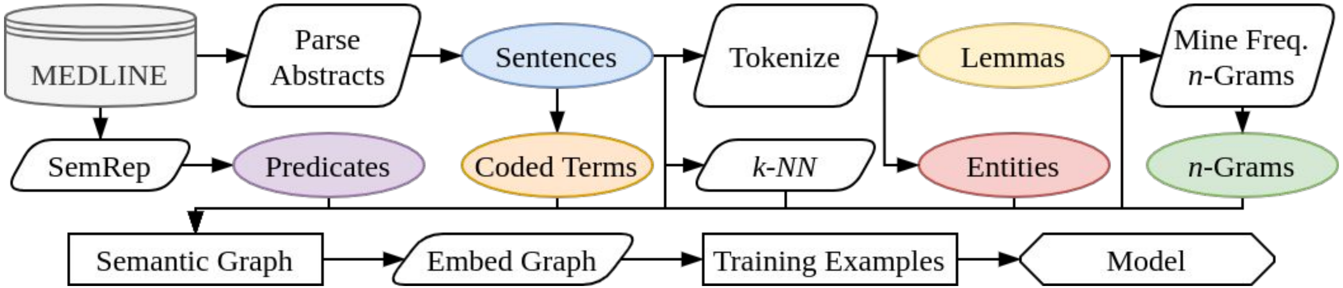
AGATHA: Automatic Graph-mining And Transformer based Hypothesis generation Approach
Sybrandt, Tyagin, Shtutman, Safro

Agatha

Motivation

- Slow analytics → Fast inference
- Heuristics → Data-driven measures
- Abstract focus → Sentence focus

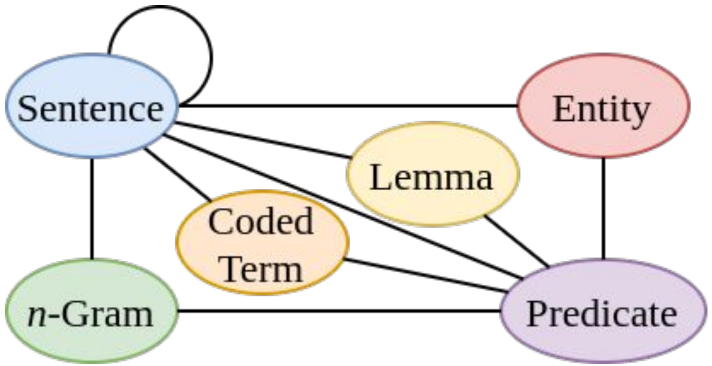
Agatha Pipeline



- Semantic Graph:
  - Split abstracts into sentences
  - Parse sentences into entities, phrases, and lemmas
  - Compute nearest-neighbors network of sentences
  - Cross reference predicate data
- Predicate Modeling
  - Embed graph
  - Learn ranking criteria

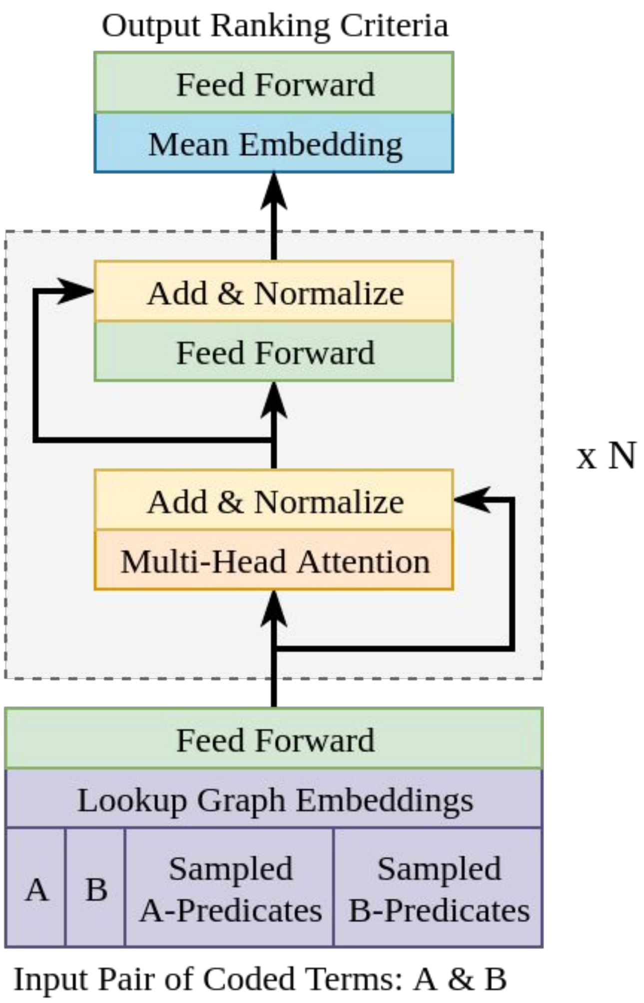
Semantic Graph

- Sentences:
  - Connected by nearest-neighbors
  - Edges to contained elements
- Predicates
  - Edges to info supplied by SemMedDB
- Size:
  - 2015 Release:
    - 188 M. Nodes
  - 2020 Release:
    - 270 M. Nodes



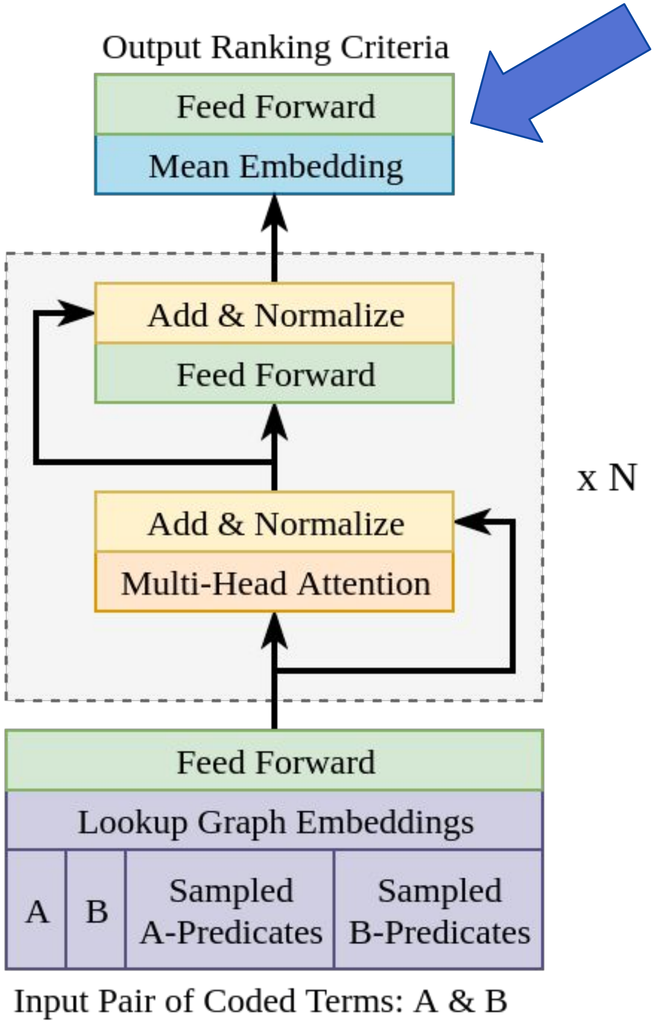
Agatha Deep Learning Model

- Goal: train a transformer encoder to accept two query terms and produce ranking criteria
- Objective: Margin Ranking Loss
- Model: Transformer Encoder
- Graph Embedding



Agatha Deep Learning Model

- Goal: train a transformer encoder to accept two query terms and produce ranking criteria
- **Objective: Margin Ranking Loss**
- Model: Transformer Encoder
- Graph Embedding



Predicate Modeling Objective

$$\mathcal{L}(\alpha, \beta) = \sum_{i=0}^{n} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nscr}_{\alpha\beta}^{(i)}\right) + \sum_{j=0}^{n'} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nswp}_{\alpha\beta}^{(j)}\right)$$

$$\text{where } L(p, n) = \max\left(0, m - \mathcal{H}(p) + \mathcal{H}(n)\right)$$

Predicate Modeling Objective

$$\mathcal{L}(\alpha, \beta) = \sum_{i=0}^{n} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nscr}_{\alpha\beta}^{(i)}\right) + \sum_{j=0}^{n'} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nswp}_{\alpha\beta}^{(j)}\right)$$

$$\text{where } L(p, n) = \max\left(0, m - \mathcal{H}(p) + \mathcal{H}(n)\right)$$

Loss associated with two connected terms

Predicate Modeling Objective

Positive Sample

Negative Sample (Scramble)

Negative Sample (Swap)

$$\mathcal{L}(\alpha, \beta) = \sum_{i=0}^{n} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nscr}_{\alpha\beta}^{(i)}\right) + \sum_{j=0}^{n'} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nswp}_{\alpha\beta}^{(j)}\right)$$

$$\text{where } L(p, n) = \max\left(0, m - \mathcal{H}(p) + \mathcal{H}(n)\right)$$

Loss associated with two connected terms

Predicate Modeling Objective

Positive Sample

Negative Sample (Scramble)

Negative Sample (Swap)

$$\mathcal{L}(\alpha, \beta) = \sum_{i=0}^{n} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nscr}_{\alpha\beta}^{(i)}\right) + \sum_{j=0}^{n'} L\left(\mathrm{PS}_{\alpha\beta}, \mathrm{Nswp}_{\alpha\beta}^{(j)}\right)$$

$$\text{where } L(p, n) = \max\left(0, m - \mathcal{H}(p) + \mathcal{H}(n)\right)$$

Loss associated with two connected terms

margin ranking loss

margin

Model output

Predicate Formulation

$$\mathrm{PS}_{\alpha\beta} = \left\{\alpha, \beta, \gamma_1^{(\alpha)}, \ldots, \gamma_s^{(\alpha)}, \gamma_1^{(\beta)}, \ldots, \gamma_s^{(\beta)}\right\}$$

where $\gamma_i^{(\alpha)} \sim \{\Gamma(\alpha) - \Gamma(\beta)\}$, and $\gamma_i^{(\beta)} \sim \{\Gamma(\beta) - \Gamma(\alpha)\}$

Predicate Formulation

Set containing two terms and other associated predicates

$$\mathrm{PS}_{\alpha\beta} = \left\{\alpha, \beta, \gamma_1^{(\alpha)}, \ldots, \gamma_s^{(\alpha)}, \gamma_1^{(\beta)}, \ldots, \gamma_s^{(\beta)}\right\}$$

where $\gamma_i^{(\alpha)} \sim \{\Gamma(\alpha) - \Gamma(\beta)\}$, and $\gamma_i^{(\beta)} \sim \{\Gamma(\beta) - \Gamma(\alpha)\}$

Set of predicates using a given term

Sampled predicate

Negative Samples

- Scramble (easy):

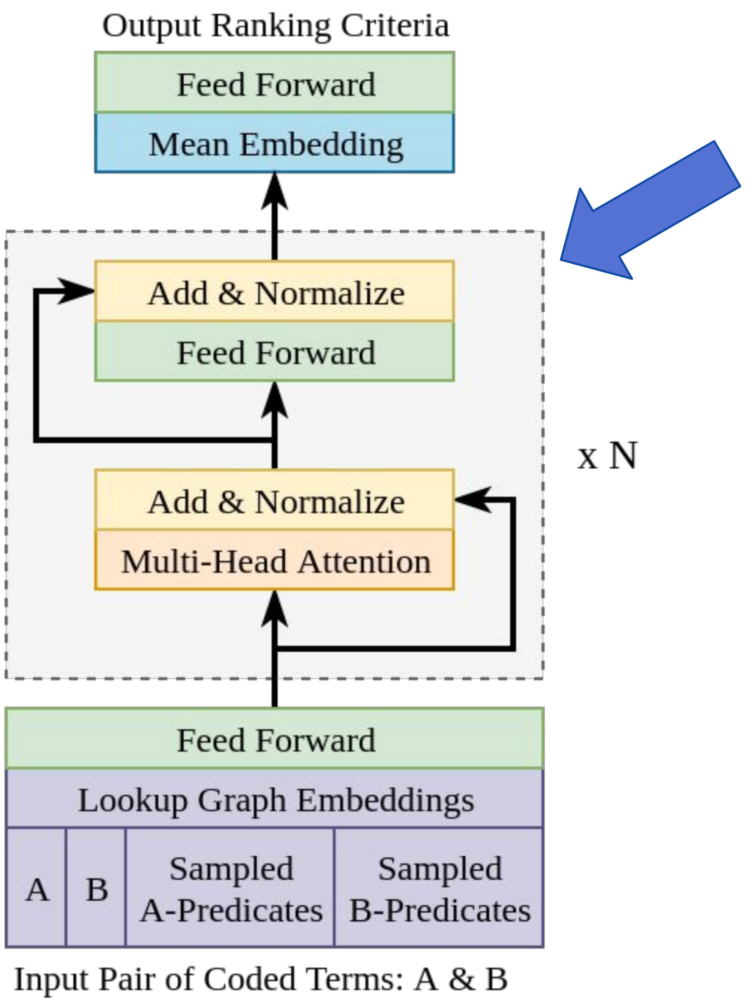
$$\text{NScr}_{\alpha\beta} = \{x, y, \gamma_1, \ldots, \gamma_{2s}\}$$
$$\text{where } x, y \sim T,$$
$$\text{and } \gamma_i \sim P,$$
$$\text{s.t. } \Gamma(x) \cap \Gamma(y) = \emptyset$$

- Swap (hard):

$$\text{NSwp}_{\alpha\beta} = \left\{x, y, \gamma_1^{(x)}, \ldots, \gamma_s^{(x)}, \gamma_1^{(y)}, \ldots, \gamma_s^{(y)}\right\}$$
$$\text{where } x, y \sim T,$$
$$\text{and } \gamma_i^{(x)} \sim \{\Gamma(x) - \Gamma(y)\},$$
$$\text{and } \gamma_i^{(y)} \sim \{\Gamma(y) - \Gamma(x)\},$$
$$\text{s.t. } \Gamma(x) \cap \Gamma(y) = \emptyset$$

Agatha Deep Learning Model

- Goal: train a transformer encoder to accept two query terms and produce ranking criteria
- Objective: Margin Ranking Loss
- **Model: Transformer Encoder**
- Graph Embedding



Model Formalism

- Prediction Model:

$$\mathcal{H}(X) = \text{sigmoid}(\mathcal{M}W)$$

$$\mathcal{M} = \frac{1}{|X|} \sum_{x_i \in X} E_N(FF(e(x_i)))$$

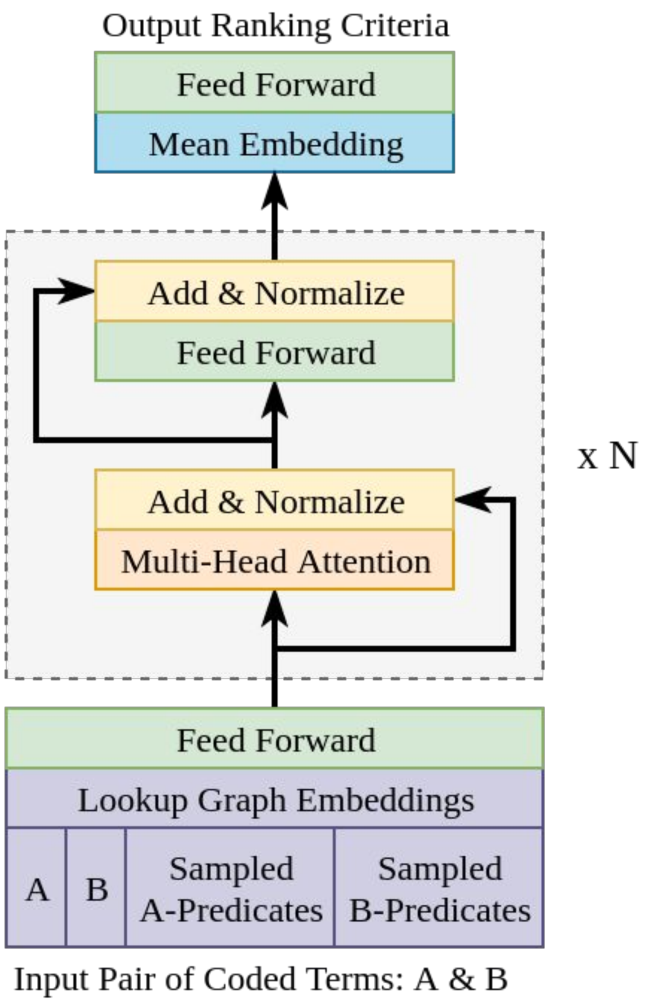
$$E_{i+1}(x) = \mathcal{E}(E_i(x)), \text{ and } E_0(x) = x$$

- Encoder Block:

$$\mathcal{E}(X) = \text{LayerNorm}(FF(\alpha) + \alpha)$$

$$\text{where } FF(Y) = \max(0, YW)\, W'$$

$$\text{and } \alpha = \text{LayerNorm}(\text{MultiHead}(X) + X)$$



Predicate Modeling

- Attention: learned weighted averages

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\intercal}}{\sqrt{d_k}}\right)V$$

Predicate Modeling

- Attention: learned weighted averages

Think: if key matches query

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

... then add in value

Predicate Modeling

- Attention: learned weighted averages

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V$$

- Multi Head Self Attention:

$$\text{MultiHead}(X) = [h_1; \ldots; h_k]W^{(4)}$$
$$\text{where } h_i = \text{Attention}\left(XW_i^{(1)}, XW_i^{(2)}, XW_i^{(3)}\right)$$

Predicate Modeling

- Attention: learned weighted averages

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\intercal}{\sqrt{d_k}}\right)V$$

- Multi Head Self Attention:

$$\text{MultiHead}(X) = [h_1; \ldots; h_k]W^{(4)}$$

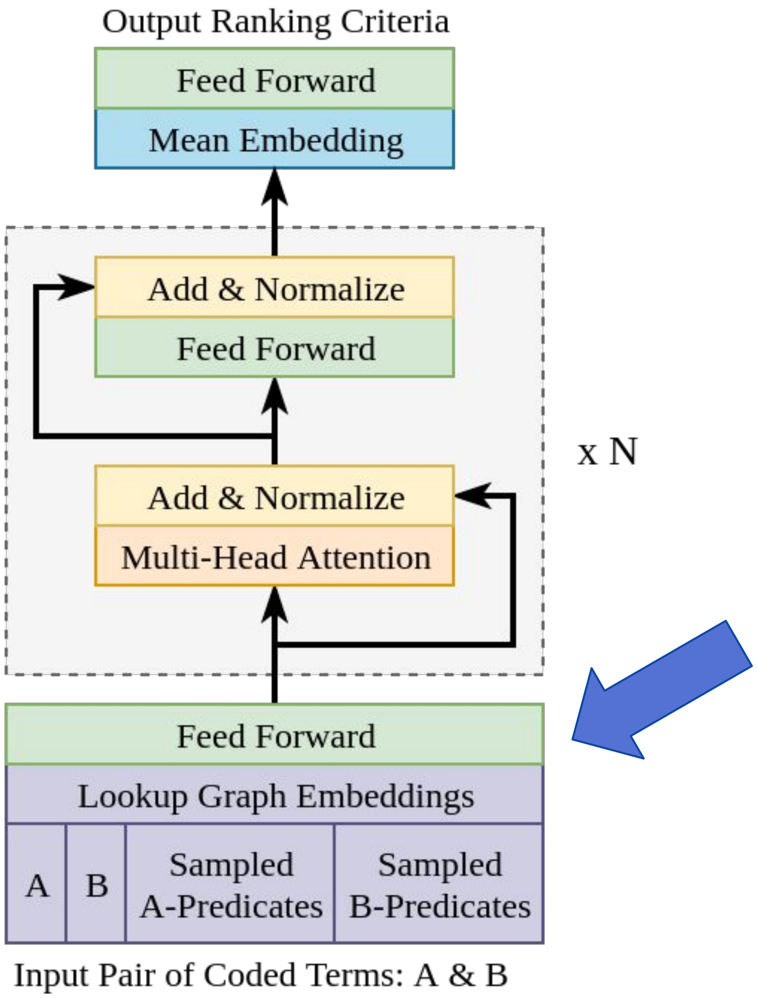
Compute multiple times and merge

$$\text{where } h_i = \text{Attention}\left(XW_i^{(1)}, XW_i^{(2)}, XW_i^{(3)}\right)$$

Derive Q, K, and V from X

Agatha Deep Learning Model

- Goal: train a transformer encoder to accept two query terms and produce ranking criteria
- Objective: Margin Ranking Loss
- Model: Transformer Encoder
- **Graph Embedding**



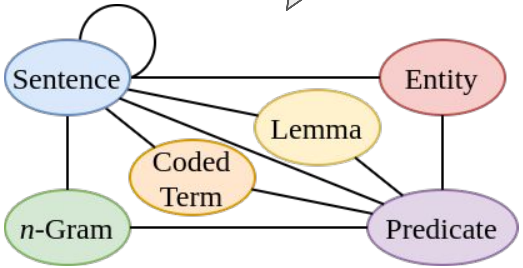
Graph Embedding

- Uses PyTorch-BigGraph (PTBG) distributed embedding
- Similarity measure:
  - biased dot product of nodes
  - includes typed translation

$$s(ij) = e(i)_1 + e(j)_1 + T_1^{(t_i t_j)} + \sum_{k=2}^{N} e(i)_k \left(e(j)_k + T_k^{(t_i t_j)} \right)$$

Graph Embedding

Each is a type



- Uses PyTorch-BigGraph (PTBG) distributed embedding
- Similarity measure:
  - biased dot product of nodes
  - includes typed translation

$$s(ij) = e(i)_1 + e(j)_1 + T_1^{(t_i t_j)} + \sum_{k=2}^{N} e(i)_k \left(e(j)_k + T_k^{(t_i t_j)} \right)$$

Estimated sim. btwn. i and j

First dim. is bias

Translated dot product

T translates between types

Graph Embedding Objective

- Minimizes Softmax Loss:
  - Positive probability close to 1
  - All negative probabilities close to 0

$$\text{GraphLoss}_{ij} = -s(ij) + \log \sum_{n=0}^{100} \exp\left(s\left(x_n^{(ij)} y_n^{(ij)}\right)\right)$$

Graph Embedding Objective

- Minimizes Softmax Loss:
  - Positive probability close to 1
  - All negative probabilities close to 0

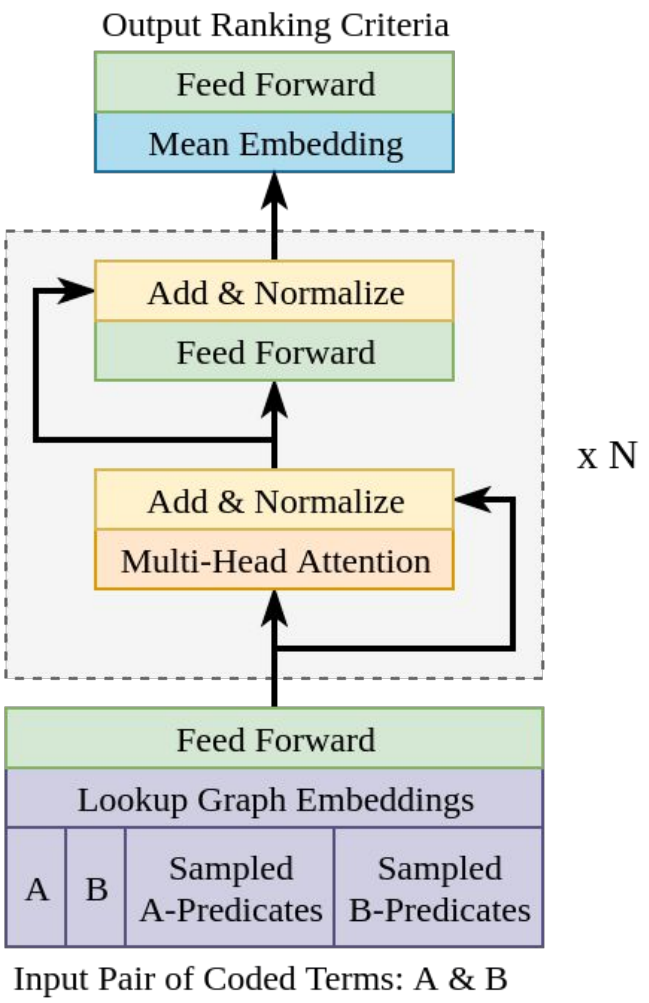
$$\text{GraphLoss}_{ij} = -s(ij) + \log \sum_{n=0}^{100} \exp\left(s\left(x_n^{(ij)} y_n^{(ij)}\right)\right)$$

positive similarity

ij score must be higher than 100 negative samples

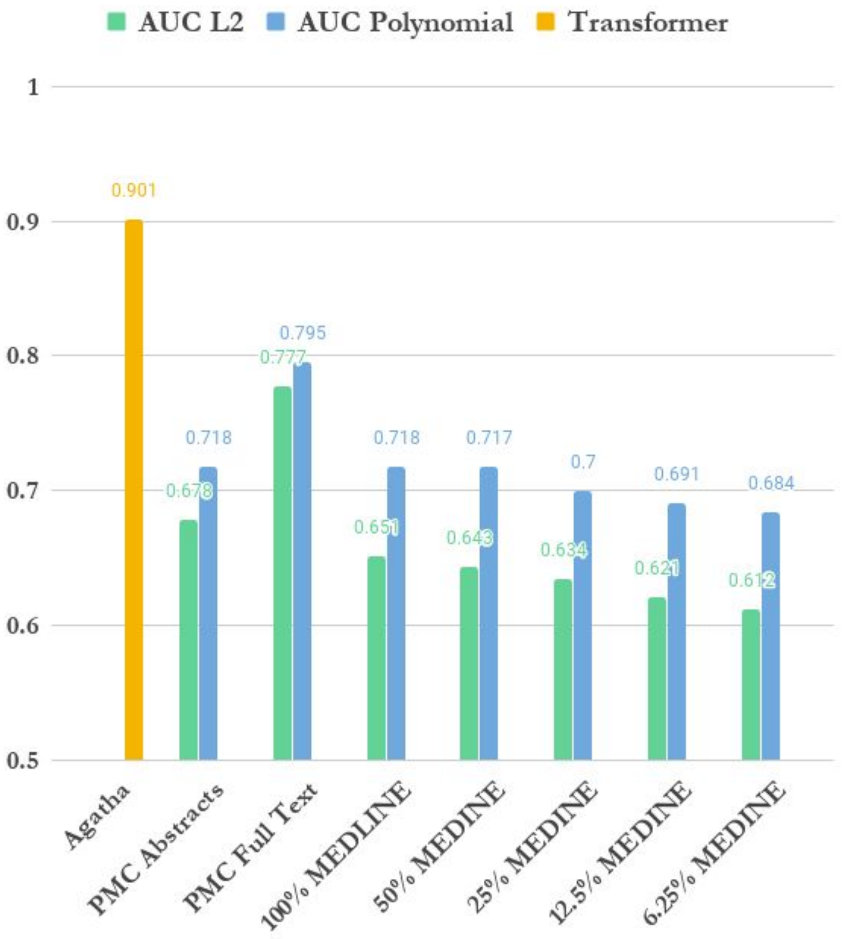
Agatha Deep Learning Model

- Goal: train a transformer encoder to accept two query terms and produce ranking criteria
- Objective: Margin Ranking Loss
- Model: Transformer Encoder
- Graph Embedding



Agatha Validation Results

- Trained on same holdout as Moliere experiments (2015)
  - Used only abstracts
- Same set of predicates
- 100's queries per minute



Beyond the Moliere Benchmark

- Moliere benchmark had significant issues
  - Balanced classes
  - Non-representative negative samples
- New validation task
  - Subdomain all-pairs recommendation
- Procedure:
  - Identify popular types of predicates
  - Find 100 most popular new findings within each predicate type
  - Predict all pairs of queries within popular entities
  - Rank
  - Compute recommendation system metrics

Gene to Cell Function

Top 100 predicates of this type.

- Area under curves:
  - PR: 0.44
  - ROC: 0.62
- Top ranked predicate is positive
- Half of the top-10 are positive
- Each one-to-many query on average:
  - 5.7 of top 10 are positive
  - Positive result within first two

Gene to Neoplastic Process

Top 100 predicates of this type.

- Area under curves:
  - PR: 0.34
  - ROC: 0.65
- Second ranked predicate is positive
- Half of the top-10 are positive
- Each one-to-many query on average:
  - 4.5 of top 10 are positive
  - Positive result within first two

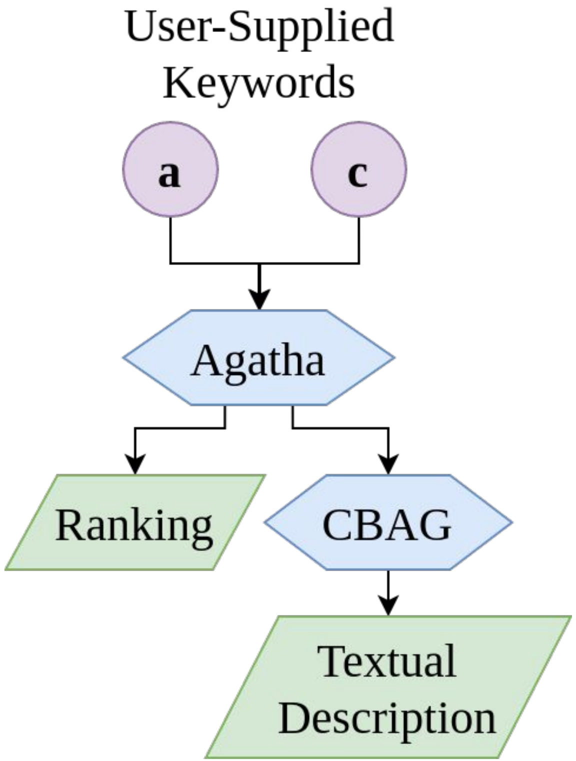
Contribution Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - **Conditional biomedical abstract generation**

CBAG: Conditional Biomedical Abstract Generation
Sybrandt, Safro

Motivation: Abstract Generation

- More interpretable hypothesis generation
- Want to explore a few connections thoroughly
- Present information in familiar way



Background Language Model

- Probability of element given previous

$$\Pr(s) = \prod_{i=1}^{n} \Pr(s_i | s_1, \ldots, s_{i-1})$$

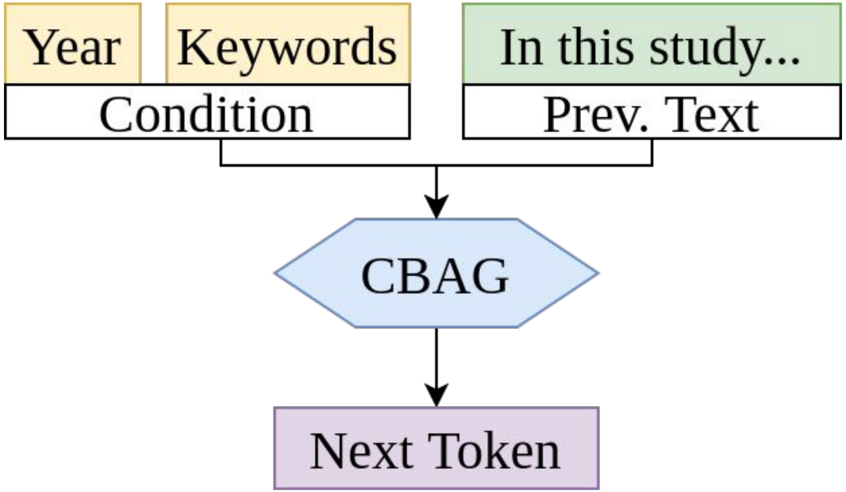
- Conditional model adds extra dependency

$$\Pr(s|c) = \prod_{i=1}^{n} \Pr(s_i | s_1, \ldots, s_{i-1}, c)$$

- Generate text by iteratively sampling Pr

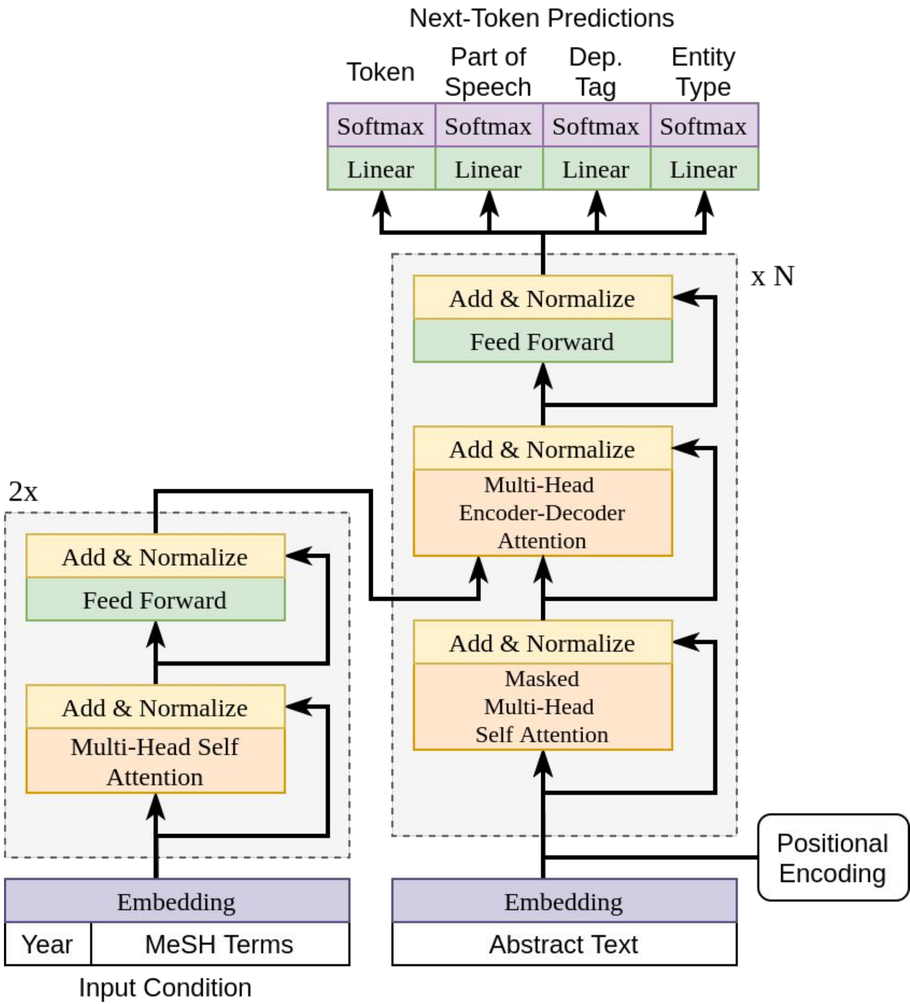
CBAG Overview

- Input condition:
  - Year of publication
  - Author-supplied keywords
- Input text:
  - All prior tokens
- Desired output:
  - Probability of next token
- Repeatedly sample to generate text



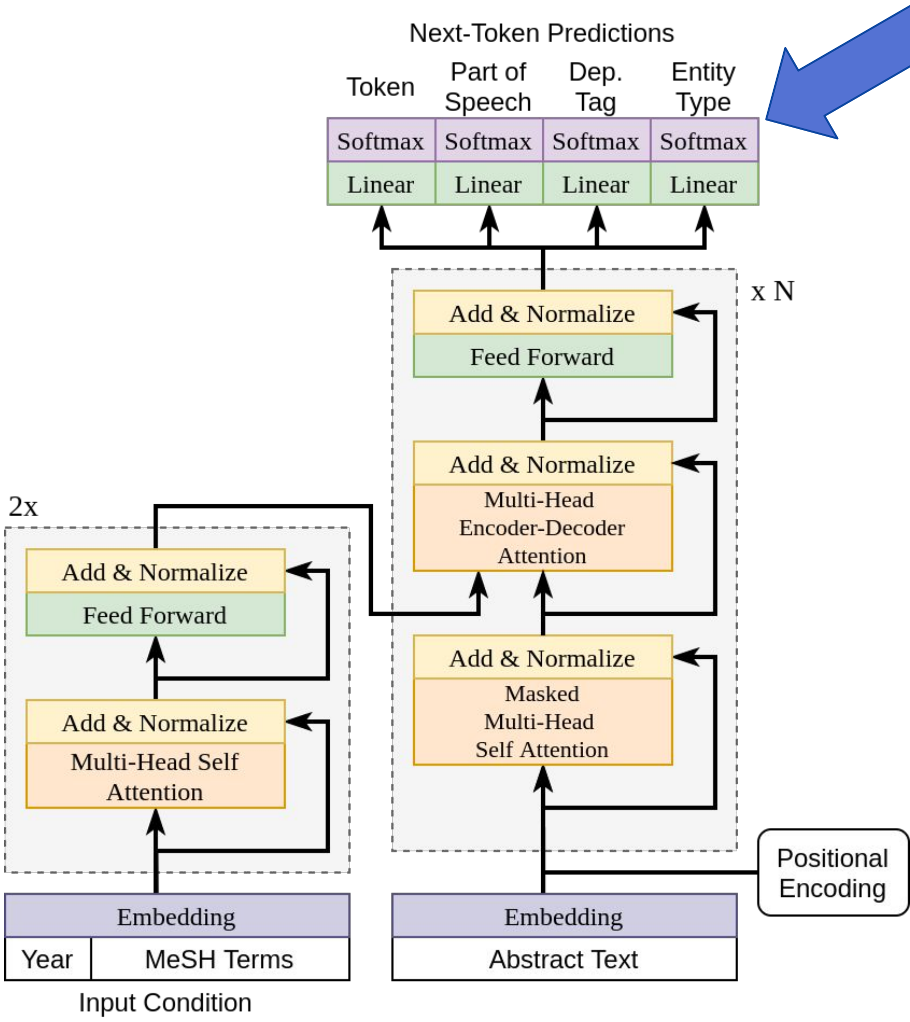
CBAG Model

- Multi-Task Objective
  - Predicts text and biomedical domain info
- Multi-Head
  - Self Attention
  - Masked Attention
  - Encoder-Decoder Attention
- Text Tokenization
  - Subword Regularization



CBAG Components

- **Multi-Task Objective**
  - Predicts text and biomedical domain info
- Multi-Head
  - Self Attention
  - Masked Attention
  - Encoder-Decoder Attention
- Text Tokenization
  - Subword Regularization



CBAG Objective

$$\mathcal{L}(t, p, d, e, c) = \overbrace{L_T}^{\text{Text}}(t, t, c) + \overbrace{L_P}^{\text{PoS}}(p, t, c) + \overbrace{L_D}^{\text{Dep}}(d, t, c) + \overbrace{L_E}^{\text{Entity}}(e, t, c)$$

CBAG Objective

Text PoS Dep Entity

$$\mathcal{L}(t,p,d,e,c) = L_T(t,t,c) + L_P(p,t,c) + L_D(d,t,c) + L_E(e,t,c)$$

Label Text Cond

Negative Log Likelihood

$$L_{[\cdot]}(\ell,t,c) = \sum_{i=1}^{n} -h_{\ell_i}^{(i)} + \log\left(\sum_{j\neq i}\exp\left(h_j^{(i)}\right)\right)$$

$$\text{where } h^{(i)} = \text{softmax}\left(\mathcal{H}\left(\{t_1,\ldots,t_{i-1}\},c\right)W_{[\cdot]}\right)$$

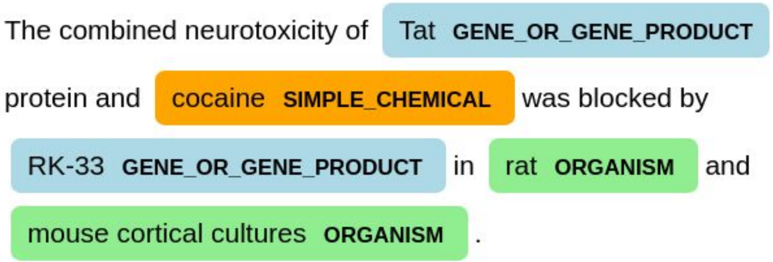
Model Prediction

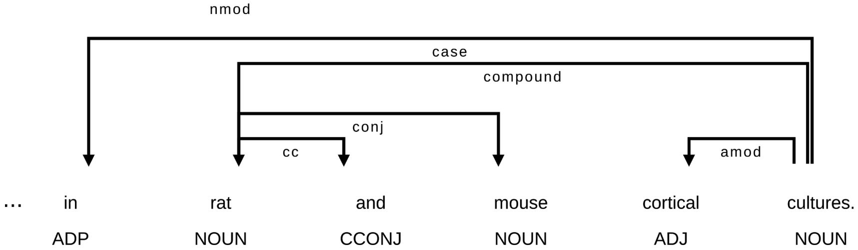
Hidden Output

Task weight

Annotations

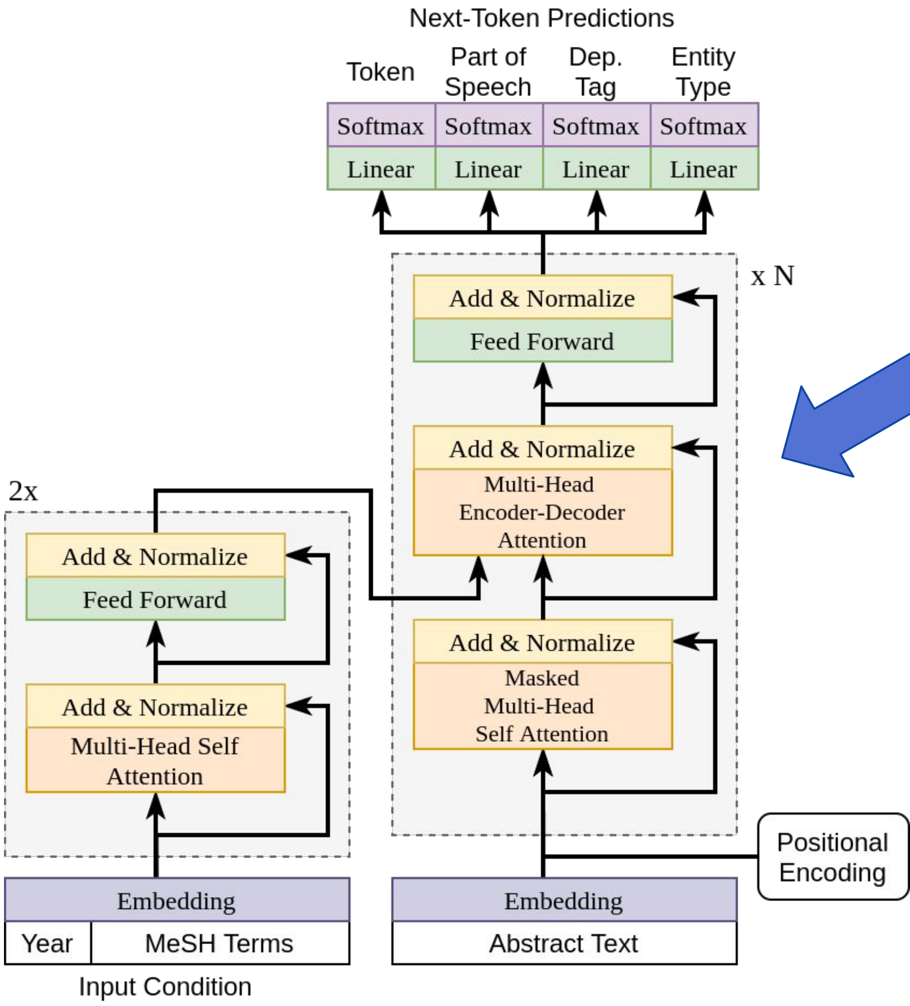
- We add domain-specific information through annotations
- No large annotation datasets
- Rely on pretrained models
  - Scispacy
- Types:
  - Part of Speech
  - Entities
  - Dependency Tags





CBAG Components

- Multi-Task Objective
  - Predicts text and biomedical domain info
- **Multi-Head**
  - Self Attention
  - Masked Attention
  - Encoder-Decoder Attention
- Text Tokenization
  - Subword Regularization



Multi-Headed Attention

- Attention: learned weighted averages

Think: if key matches query

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\intercal}{\sqrt{d_k}}\right) V$$

... then add in value

Multi-Headed Attention

- Attention: learned weighted averages

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

- Multi Headed Self Attention:

$$\text{MultiHead}(X, Y) = [h_1; \ldots; h_k]W^{(4)}$$

$$\text{where } h_i = \text{Attention}\left(XW_i^{(1)}, YW_i^{(2)}, YW_i^{(3)}\right)$$

Query from X
Keys and Values from Y

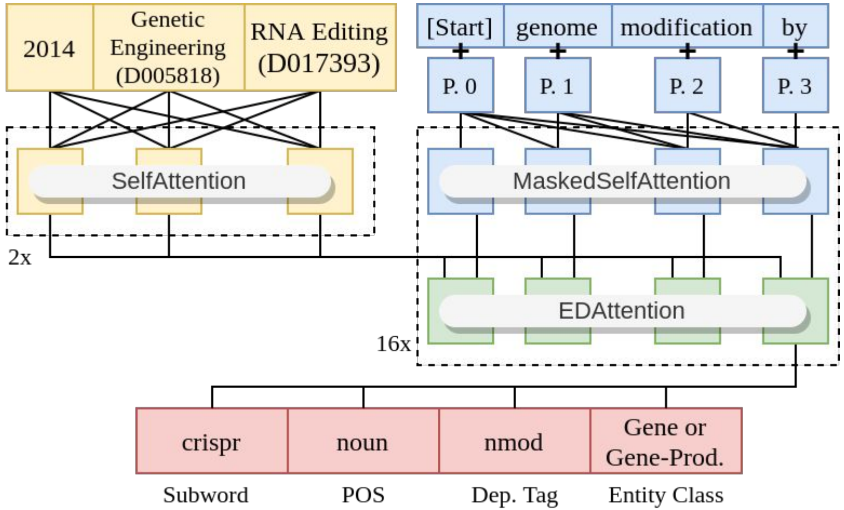
Model Details

$$\mathcal{H}(t, c) = D_d$$
$$D_{i+1} = \mathcal{D}(D_i, E_e) \text{ and } D_0 = t + \text{PE}$$
$$E_{i+1} = \mathcal{E}(E_i) \text{ and } E_0 = c$$

$$\mathcal{E}(X) = \text{LayerNorm}(\text{FF}(\alpha) + \alpha)$$
$$\alpha = \text{LayerNorm}(\text{MultiHead}(X, X) + X)$$

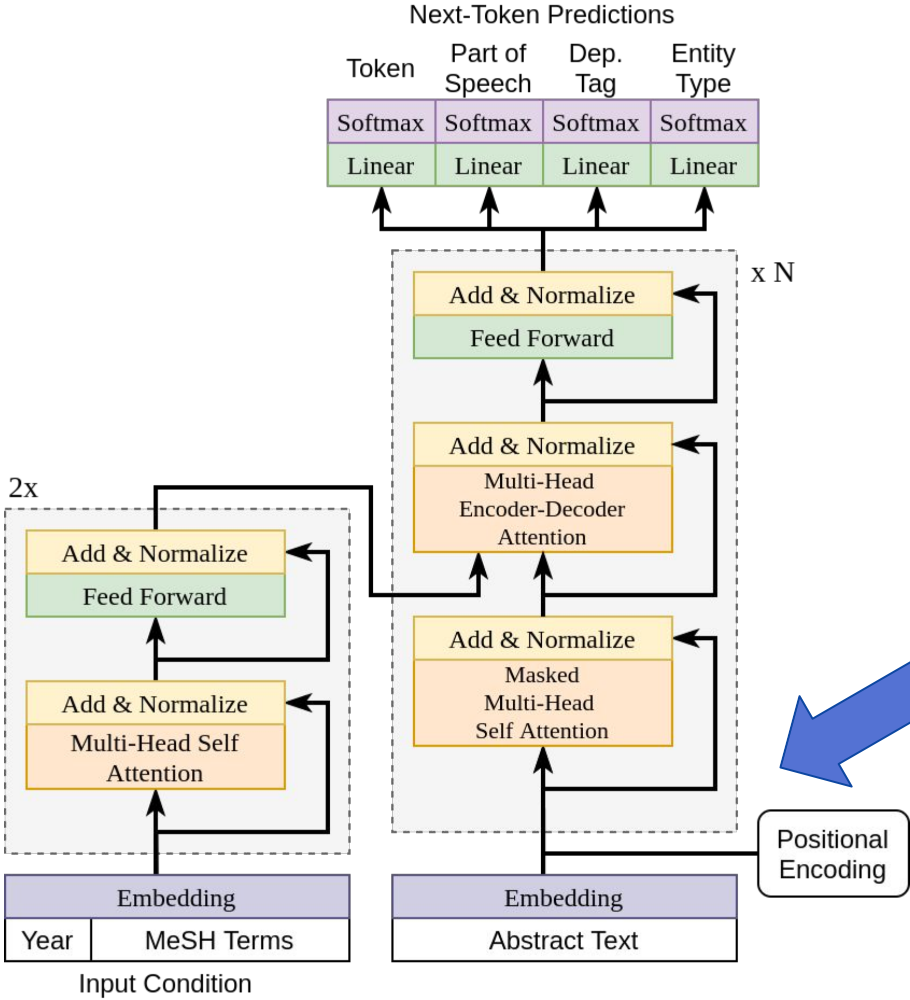
$$\mathcal{D}(X, Y) = \text{LayerNorm}(\text{FF}(\alpha) + \alpha)$$
$$\alpha = \text{LayerNorm}(\text{MultiHead}(\beta, Y) + \beta)$$
$$\beta = \text{LayerNorm}(\text{MultiHead}(X, X) + X)$$

$$\text{FF}(X) = \max(0, XW)W'$$



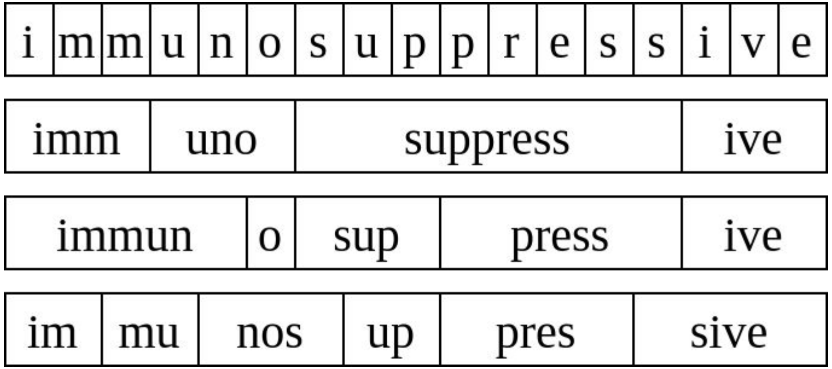
CBAG Components

- Multi-Task Objective
  - Predicts text and biomedical domain info
- Multi-Head
  - Self Attention
  - Masked Attention
  - Encoder-Decoder Attention
- **Text Tokenization**
  - Subword Regularization



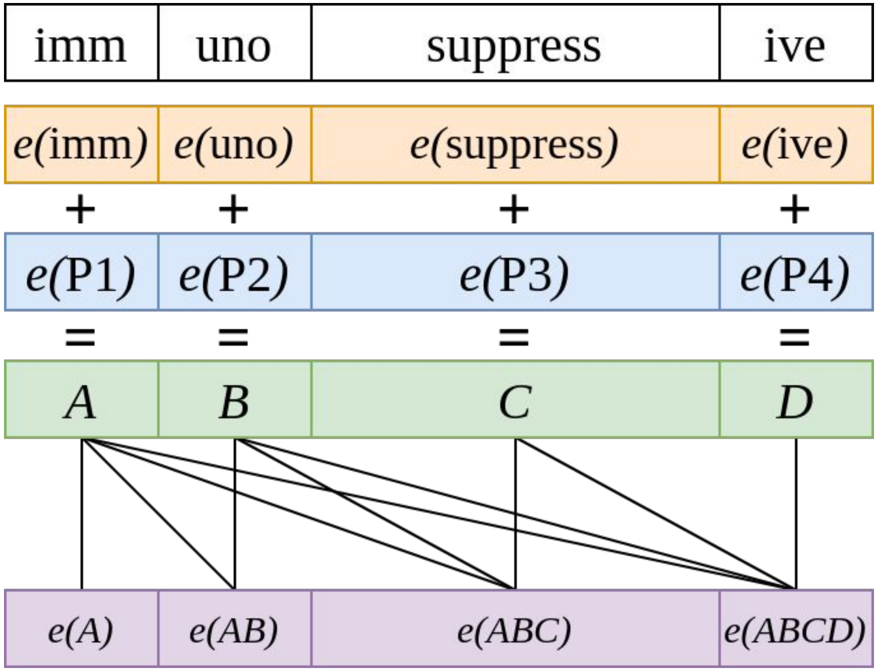
Unigram Subword Regularization

- Goal:
  - Limit number of unique tokens
  - Reduce out-of-vocab words
- WordPiece
  - Find common substrings
  - One for each letter
- Subword regularization
  - Probabilistically tokenize words



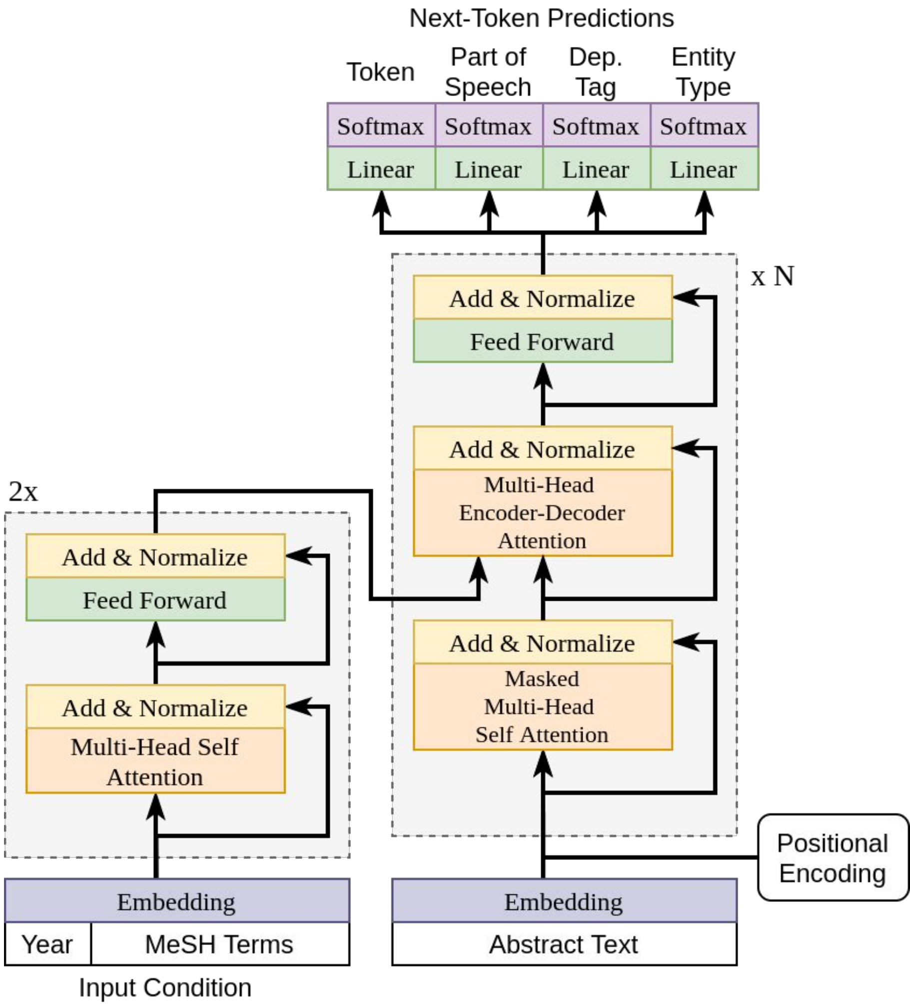
Subword Embedding

- Assert number of subword units
  - 16,000
- Initial random embeddings
- Positional Encoding
  - Sinusoidal embedding function
- Masked Self Attention
  - Each subword gains context
  - Larger words built from subwords



CBAG Components

- Multi-Task Objective
  - Predicts text and biomedical domain info
- Multi-Head
  - Self Attention
  - Masked Attention
  - Encoder-Decoder Attention
- Text Tokenization
  - Subword Regularization



Experimental Design

- Holdout 30% of MEDLINE for testing
- Input:
  - Year
  - Metadata Keywords
  - Title sentence
- Output:
  - Generate text until "end-of-abstract" special token
- Metrics:
  - Compare *n*-gram recall
  - Bleu, METER, CIDEr, etc.
- Baseline: GPT-2

Example Abstract: CBAG

Hierarchically Micro- and Nanopatterned Topographical Cues for Modulation of Cellular Structure and Function.

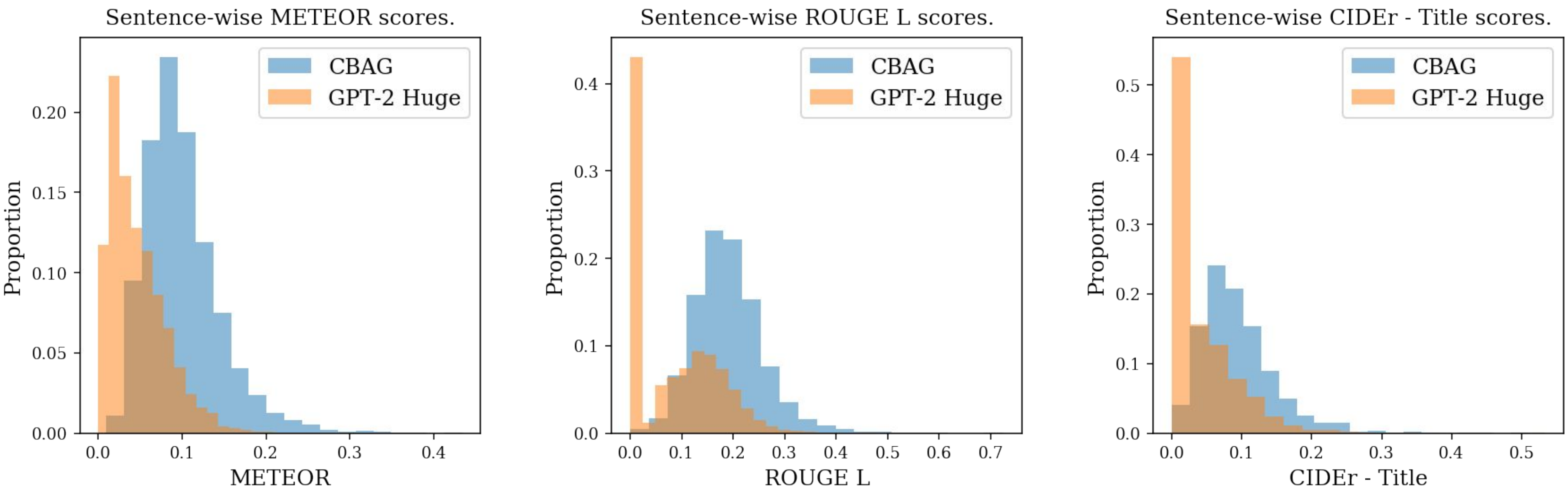
the ability to integrate multiple physiological cues and thereby mediate many cellular functions is critical for many complex life history processes. despite recent advances in **high-throughput imaging of biomolecules** and their spatiotemporal integration into dynamic structure and function, the precise structural organization and temporal structure of tissue architecture remains poorly understood. here, we present an efficient system for temporally and spatially mapping micro- and nanopatterned topographical cues in organ-specific spatial and temporal properties using **multiple imaging modalities.** the **micro- /nanopatterned geometrical cues** can be localized to cell membranes, cells, and proteins. the spatial and temporal dynamics of these local signals are precisely represented by the **cross-correlation function**, which forms the basis of a geometrical model that accurately provides spatiotemporal information about the spatial location and spatial coordinate of the labels and their functionalities. the model is also capable of correlating the properties of neural cells within their network without affecting the spatial and temporal organization of their spatial features, as well as those of their surrounding tissue. as an example of this model, cell types grown as multilayers are described.

Example Abstract: GPT-2

<u>**Hierarchically Micro- and Nanopatterned Topographical Cues for Modulation of Cellular Structure and Function.**</u>

[\n] [\n] The first example of what happens when an **artificial neural network**'see' complex features on a computer screen is provided by the vision of people with sight loss. The most recent results (e.g., see **Krueger's blog**) suggest that a person who has a damaged retinal pathway for **visual light-scattering** called LID is able to'see' objects using low power laser sources. The finding is unexpected because it suggests an adaptive mechanism for recovering limited visual function. The finding has profound implications for understanding how the retina and brain develop, how vision is learned, and how people can potentially **regain impaired vision with new retinal implants.** [\n] The retinotopic map shows which parts of the retina belong to what parts of the visual field (more details in the video below). [\n] The following images are from a retinotopic map showing a person who has lost some peripheral vision due to an injury in his optic nerve and **retinal pigment epithelium**. Note that he can make out details when looking toward the left, and when looking toward the right, details become distorted or are obscured.

Comparison to GPT-2



Varying Condition

Condition	Response
D003270: Contraceptive Agents	...that, during a prospective observational period, the patients were aware of the possibility of adverse cardiac events.
D003634: DDT	...that the aromatic (g)-tse, which is often produced in fruit, is potentially useful to suppress green algae as well as pesticide toxicity.
D004042: Unsaturated Dietary Fats	...that vitamin e levels are associated with early childhood health consequences.
D006046: Gold	...that the nanoparticles provide improved sensitivity to gold nanoparticles, and they are sensitive to ag-b interaction rather than ca-a interaction.
D005395: Fish Oils	...that the combination of pinkland and fish oil intakes (ca-like and ca-like) improves the antioxidant effect of yinneria (tricapsa vul) and that can significantly decrease food intake.

Table 2: Differing generations of the same prompt given various MeSH preconditions. We record the first sentence completing the prompt *"In this study, we found..."*

In Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation

Acknowledgements

- Committee: Ilya Safro, Alexander Herzog, Amy Apon, Brian Dean, Sez Atamturktur
- Co-Authors:
  - Michael Shtutman (U. of South Carolina)
    - Moliere, Validation, Agatha
  - Ruslan Shaydulin
    - Partitioning
  - Ilya Tyagin
    - Agatha
- Funding:
  - GAANN DAISE (US. Dept. of Ed.)
  - NRT RIES (NSF #1633608)

In Summary

- Graph Embedding
  - FOBE & HOBE bipartite embedding
  - Embedding-based coarsening for hypergraph partitioning
- Automatic Hypothesis Generation
  - Moliere: hypothesis generation via topic modeling
  - Validation of hypothesis generation via candidate ranking
  - Evaluation of corpora on generated hypotheses
  - Agatha: deep-learning hypothesis generation
  - Conditional biomedical abstract generation