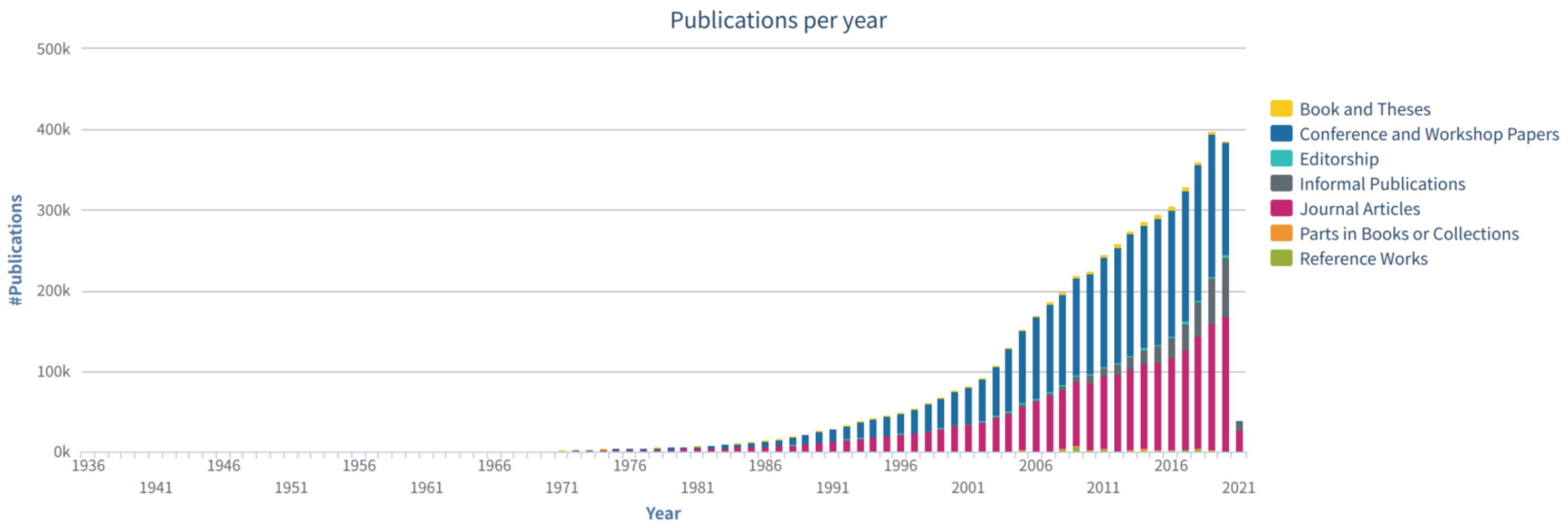
MRT: Tracing the Evolution of Scientific Publications

Da Yin, Weng Lam Tam, Ming Ding, and Jie Tang

清華大學
Tsinghua University

Backgrounds

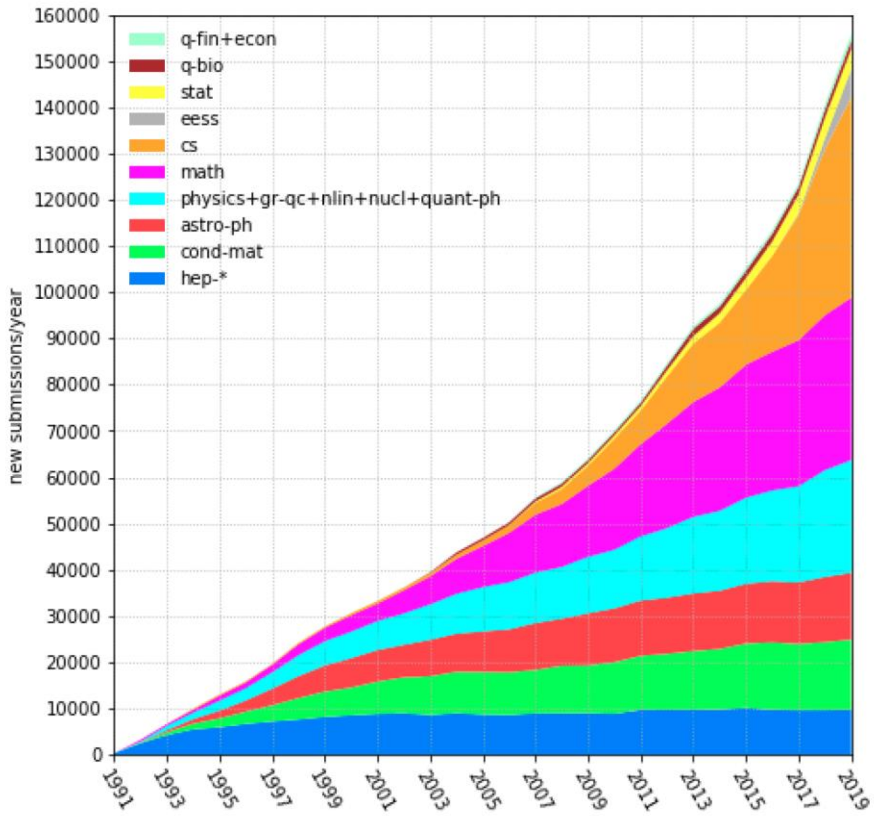
- Science evolution is becoming more and more fast
  - Computer Science: Number of publications in DBLP has grown a lot
    - 2000 (**77k**) -> 2020 (**408k**) +430%
  - For example, top AI conferences accept **over 1,000** papers every year
    - 2020: CVPR (**1,467**), AAAI (**1,591**)



Number of publications in DBLP

Background

- Other research fields
  - Biology, Math, Physics, etc. : the number of arXiv publications also increases a lot at various speed.
  - STM report: The number of all kinds of publications in 2018 reaches over **3 million** and continuously goes up with a rate of **6%** each year.
- Researchers need to digest lots of latest papers!
- **Data mining** techniques can be used to help scholars find useful information

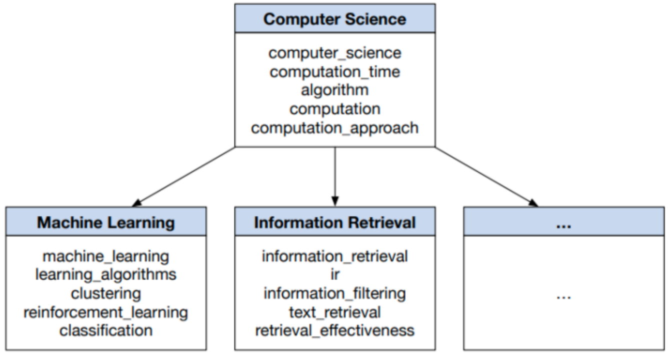


Number of publications in arXiv

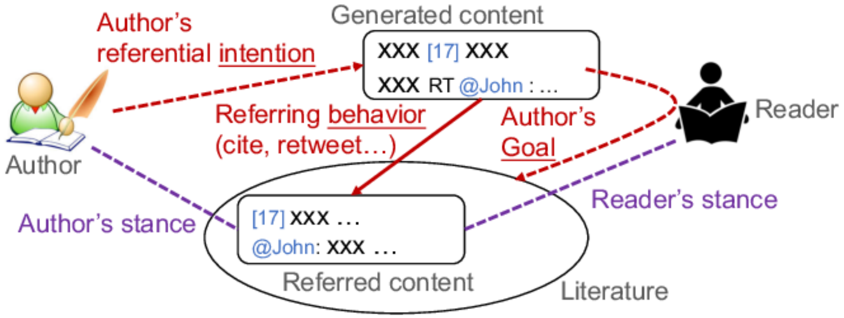
Johnson R, Watkinson A, Mabe M. The stm report[J]. An overview of scientific and scholarly publishing. 5th edition October, 2018

Background

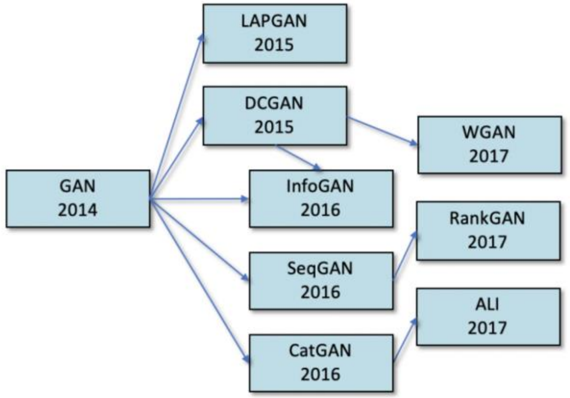
- Previous research on academic data mining
  - Concept extraction: Extract concepts from papers and construct taxonomy
  - Citation analysis: Analyse the roles of citations
  - Algorithm roadmap: Sketch algorithm evolution graph from papers
- Problem: Mainly focus on the over generalized information and lose lots of paper details



Concept extraction



Citation analysis

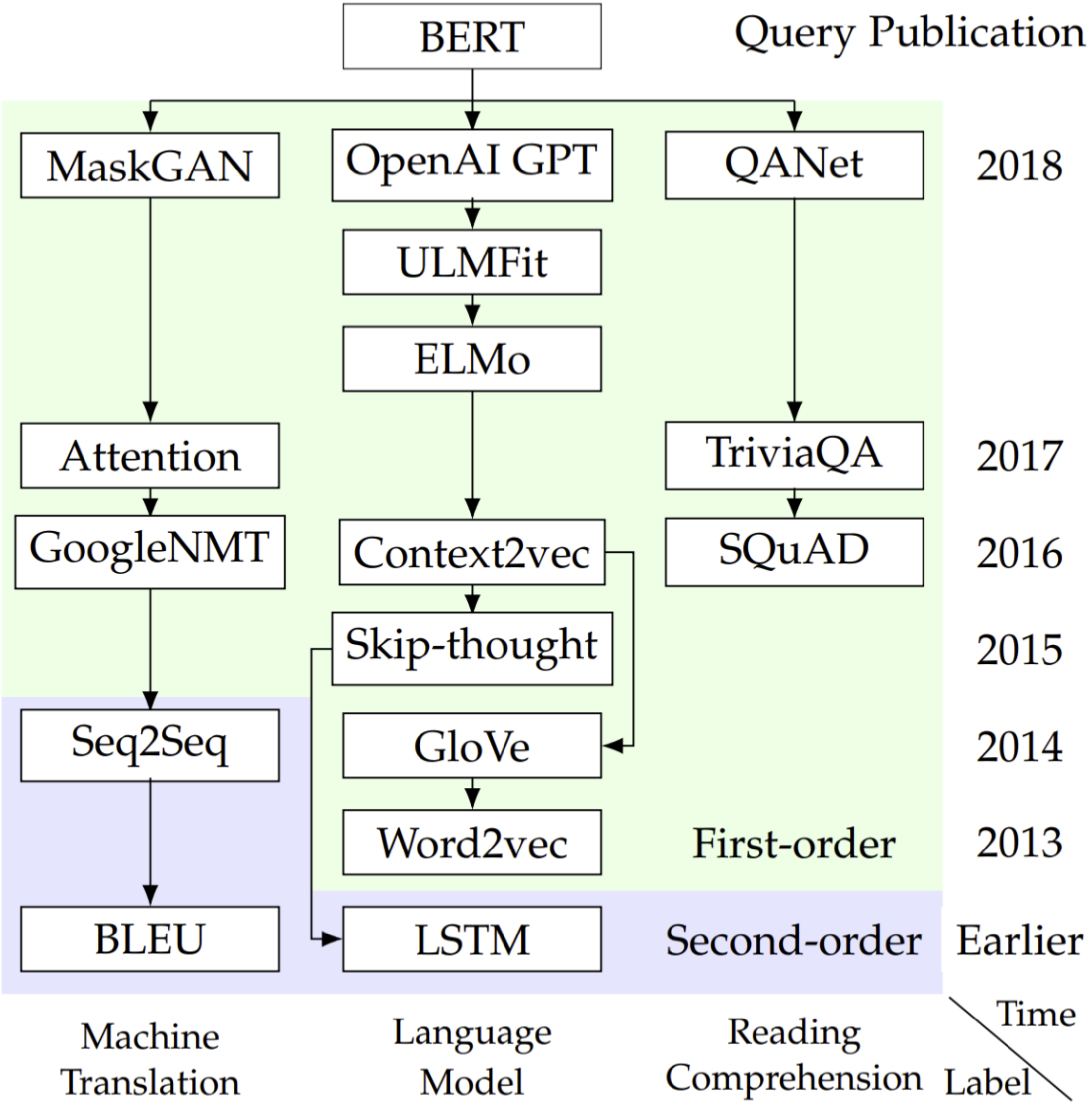


Algorithm roadmap

Zhang C, Tao F, Chen X, et al. Taxogen: Constructing topical concept taxonomy by adaptive term embedding and clustering[C]. KDD. 2018
Yu, W., Yu, M., Zhao, T., & Jiang, M. Identifying Referential Intention with Heterogeneous Contexts[C]. WWW. 2020
Zha H, Chen W, Li K, et al. Mining algorithm roadmap in scientific publications[C]. KDD. 2019

Background

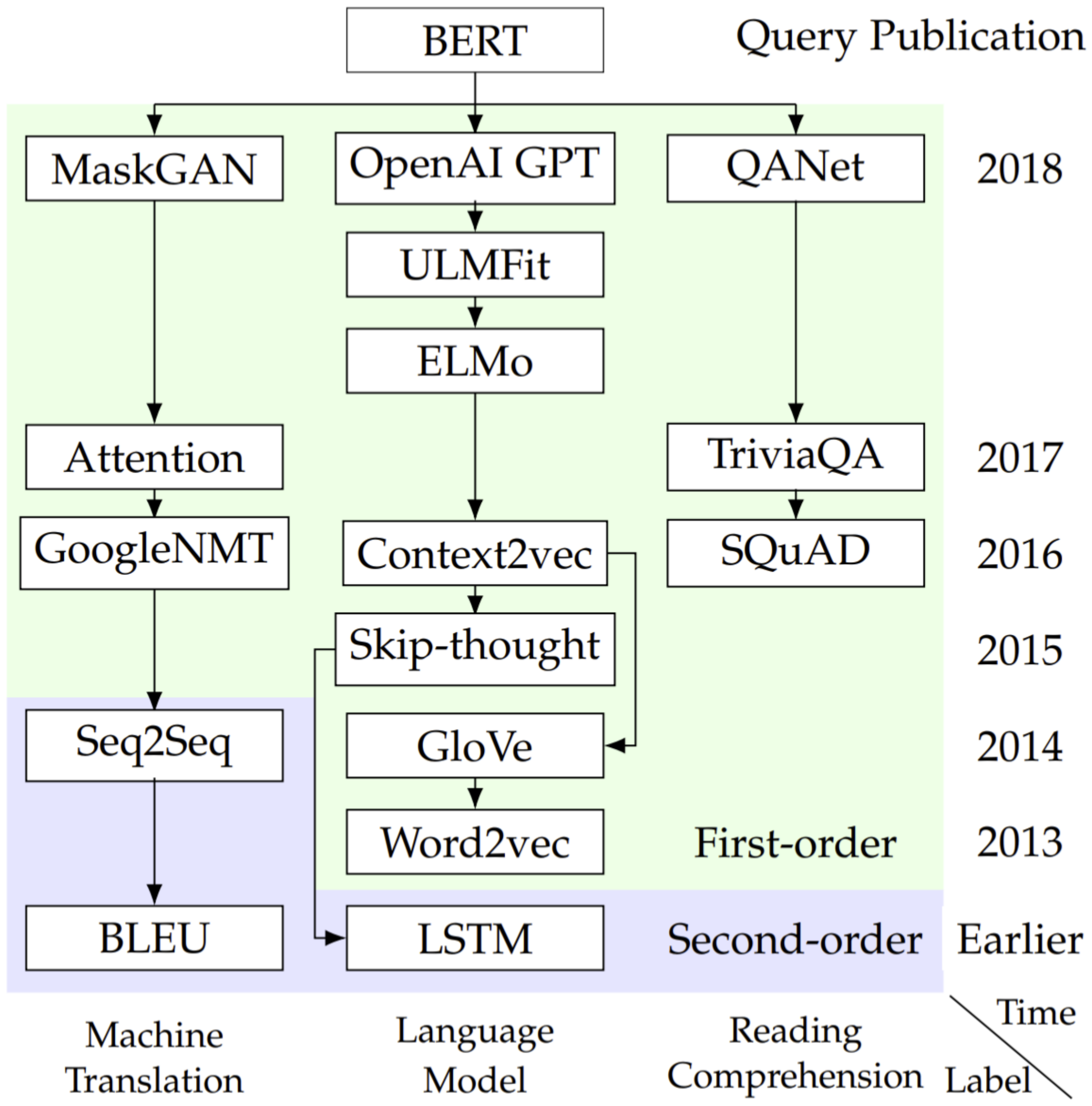
- What researchers need?
- For example, where does BERT's ideas come from?
  - Some ideas come from **Language Model**
    - Pre-training: GPT / ULMFit / ELMo
    - Word Embedding: GloVe, Word2vec
    - Sequence Encoding: LSTM
  - Some ideas come from **Machine Translation**
    - Transformer: Attention
    - MLM: MaskGAN
  - Some come from **Reading Comprehension** ...



Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C]. NAACL-HLT. 2019

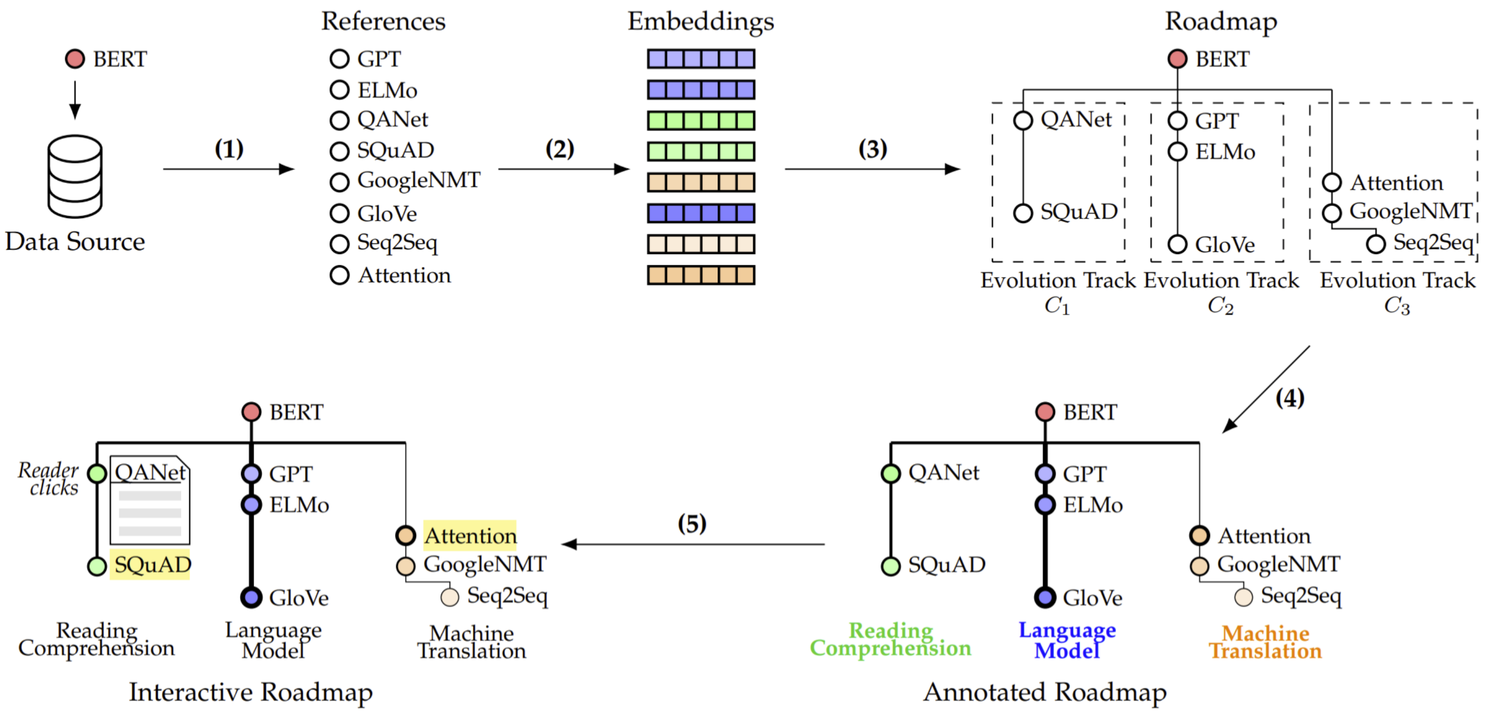
Problem Definition

- Given source publication q and other configurations (N_p, N_t, N_l), generate an evolution roadmap, including:
  - V : N_p nodes, each represents a paper
  - E : $N_p - 1$ edges, represents evolution footprint
  - C : N_t evolution tracks, represents various evolution path. Each track contains N_l labels
  - W : Importance scores, including $N_p - 1$ papers and N_t evolution tracks

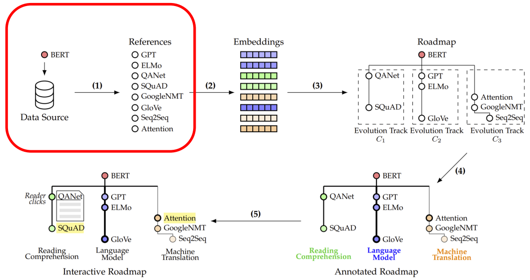


Method

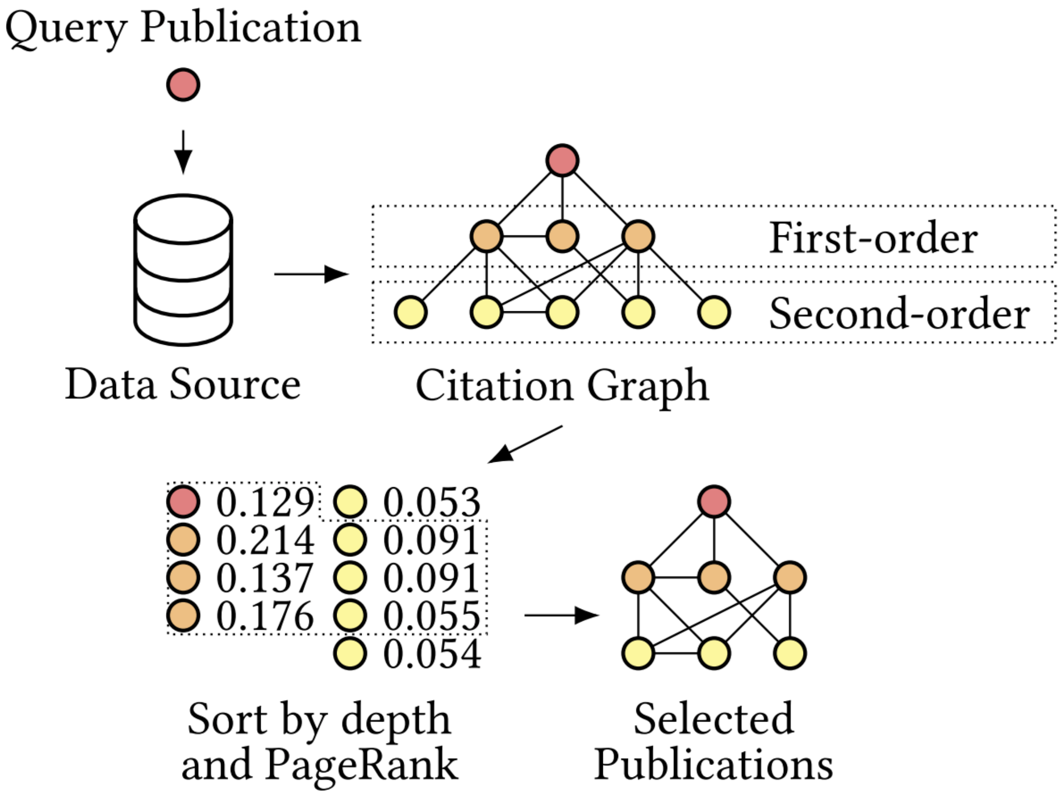
- 1. Fetch reference papers
- 2. Generate paper embeddings
- 3. Generate evolution tracks
- 4. Generate labels and importance scores
- 5. Interact with users



Method

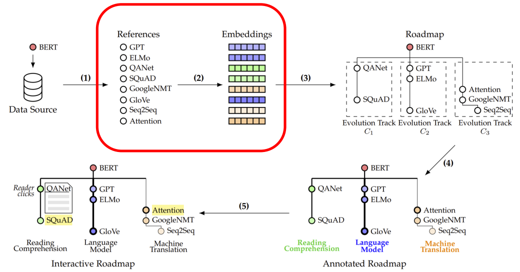


- 1. Fetch reference papers
  - Data source
    - SemanticScholar & AMiner
    - Web API
    - Only metadata (title + abstract)
  - Extend higher order reference papers
  - Build citation graph
  - Use PageRank (or other algorithm) to select papers

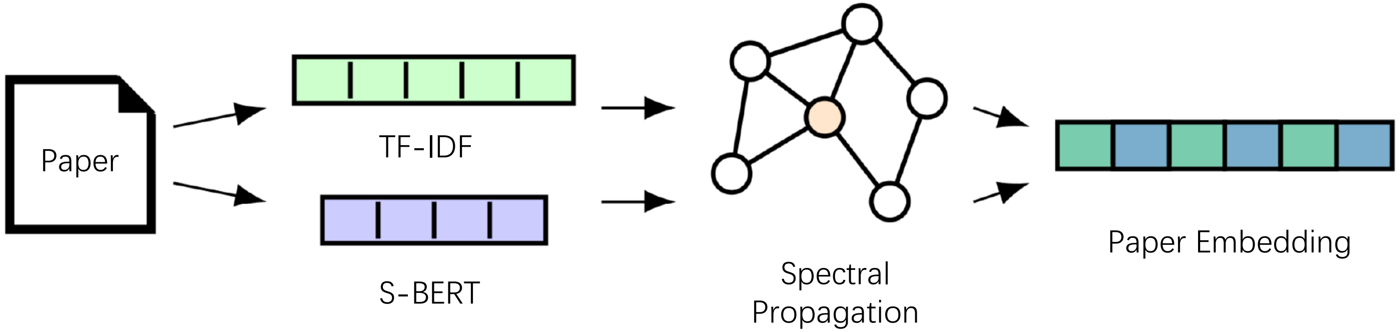


Page L, Brin S, Motwani R, et al. The pagerank citation ranking: Bringing order to the web. [R]. Stanford InfoLab, 1999.

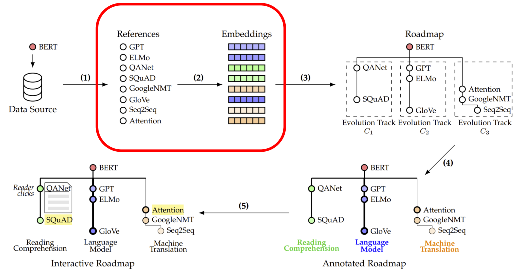
Method



- 2. Generate paper embeddings
  - Use **TF-IDF / S-BERT** to encode paper semantic information (title + abstract)
    - TF-IDF focuses on literal information and is good at identifying keywords
    - Sentence-BERT focuses on latent semantic information
  - Use **spectral propagation** in **ProNE** to incorporate structural information
    - ProNE propagates information to neighbourhoods



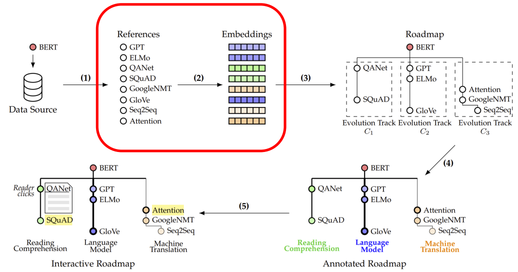
Method



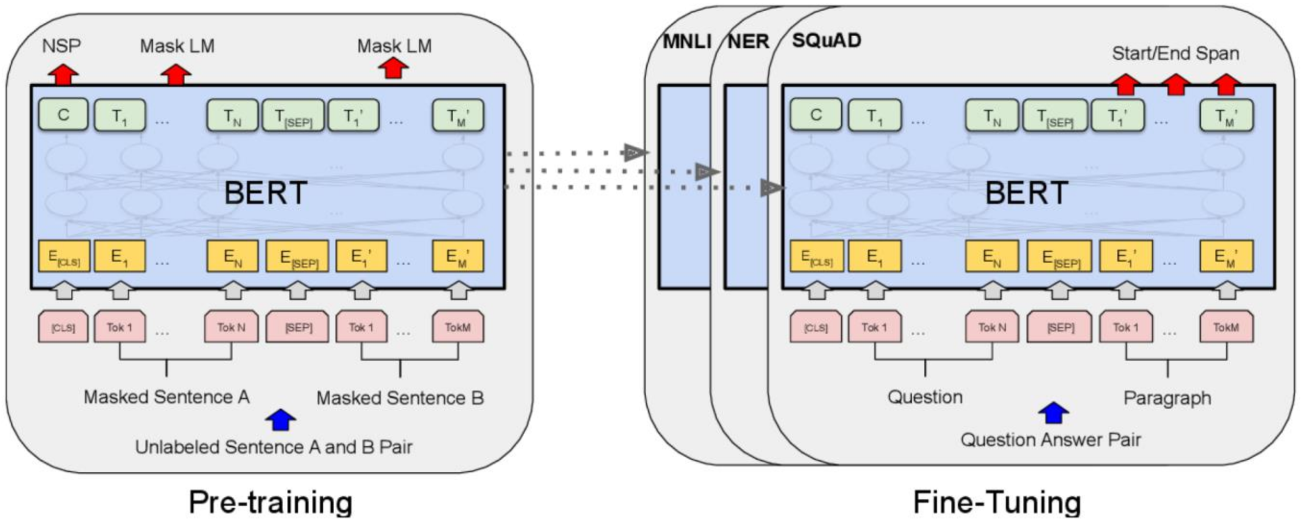
- 2. Generate paper embeddings
  - TF-IDF: Term Frequency – Inverse Document Frequency
    - Lemmatization & N-gram
    - Take n_w most frequent words in subgraph to build TF-IDF document vector

$$\mathtt{TF\text{-}IDF}(word_k|d_i) = \frac{\#word_{ki}}{|d_i|} \log \frac{N_p}{\sum_{i'} 1\{\#word_{ki'} > 0\}}$$

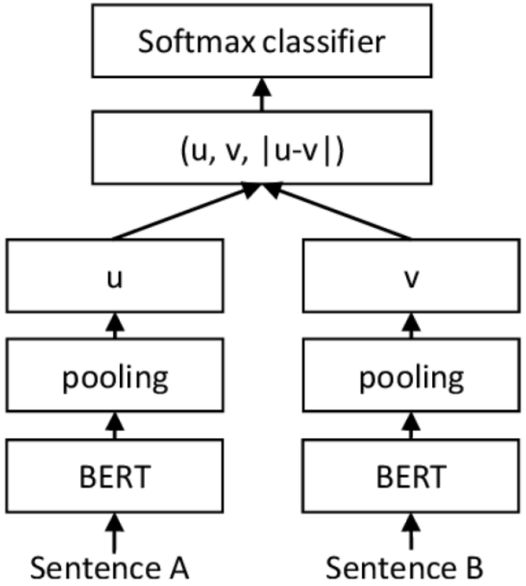
Method



- 2. Generate paper embeddings
  - S-BERT: SentenceBERT
    - Fine-tune sentences on top of pre-trained BERT model
    - Encode latent semantic information



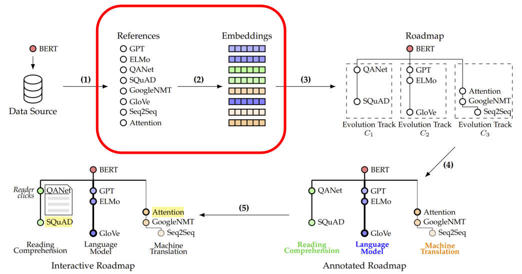
BERT pre-training model



SentenceBERT fine-tuning structure

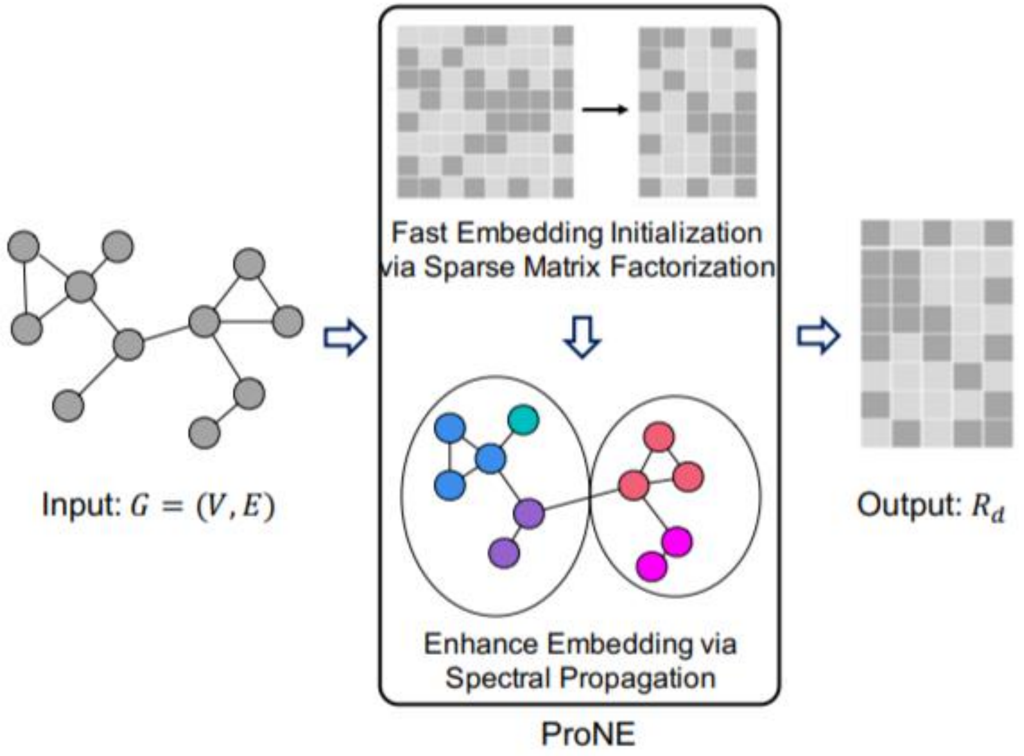
Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019

Method



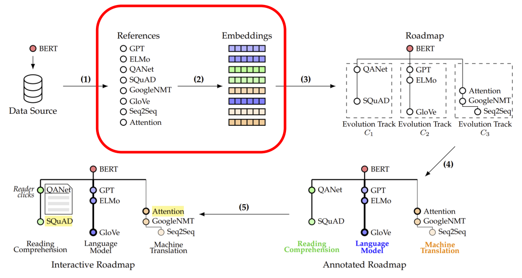
- 2. Generate paper embeddings
  - ProNE
    - Fast matrix factorization to initialize node embeddings
    - Spectral propagation to enhance representation capability on local and global signals
      - Propagation process

$$x \leftarrow D^{-1}A(I_N - \tilde{L})x$$



Zhang J, Dong Y, Wang Y, et al. Prone: fast and scalable network representation learning[C]. IJCAI. 2019: 4278-4284

Method

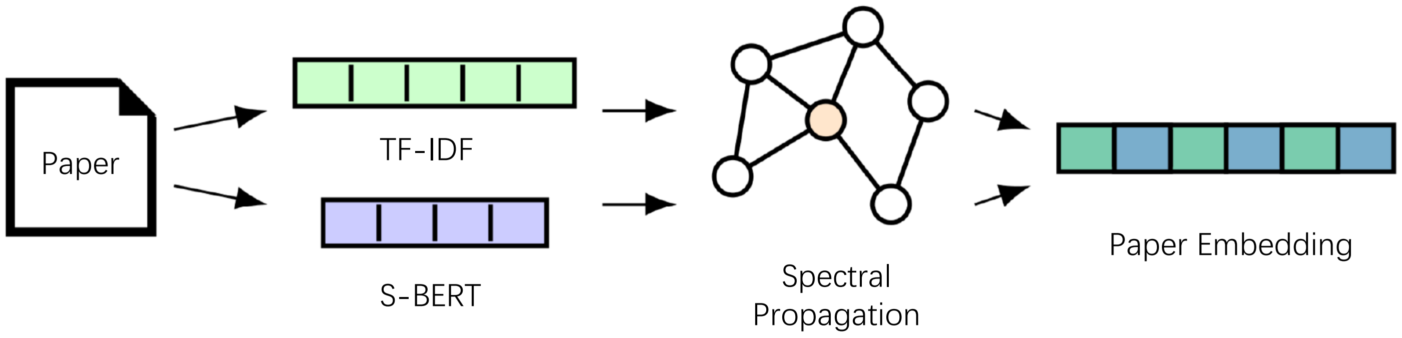


- 2. Generate paper embeddings
    - Propagation

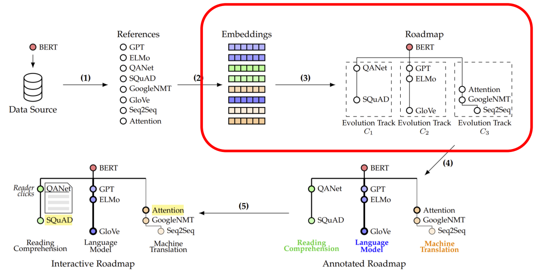
$$\tilde{x}_i^t = \mathtt{TF\text{-}IDF}(p_i),\ \tilde{x}_i^s = \mathtt{S\text{-}BERT}(p_i)$$

$$\hat{x}_i^t = \mathtt{Propagate}(\tilde{x}_i^t, G),\ \hat{x}_i^s = \mathtt{Propagate}(\tilde{x}_i^s, G)$$

$$x_i = \mathtt{Propagate}([\hat{x}_i^t; \hat{x}_i^s], G)$$



Method

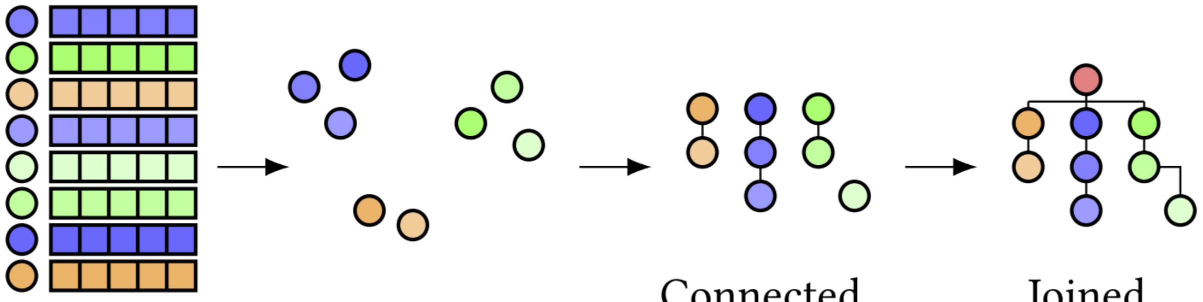


- 3. Generate evolution tracks
  - Use kernel k-means to cluster $N_p - 1$ reference papers into N_t topics

$$\|x_i - m_{C_t}\|^2 = K_{ii} - \frac{2\sum_{p_j \in C_t} K_{ij}}{|C_t|} + \frac{\sum_{p_j, p'_j \in C_t} K_{jj'}}{|C_t|^2}$$

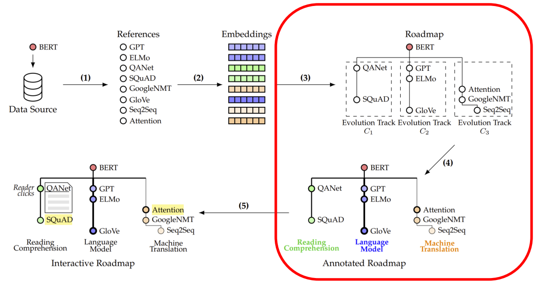
$$K_{ij} = x_i^T x_j + \alpha A_{ij} + \beta \Phi_{ij}$$

  - Connect papers according to their publication date or citation order



Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

Method



- 4. Generate labels and importance scores
  - Label generation
    - First extract label candidates
      - N-gram + Frequency threshold
    - Then sort candidates according to three criteria
      - Label should cover the paper content in current evolution tracks
      - Label should be different from other evolution tracks
      - Label should be related to the source paper

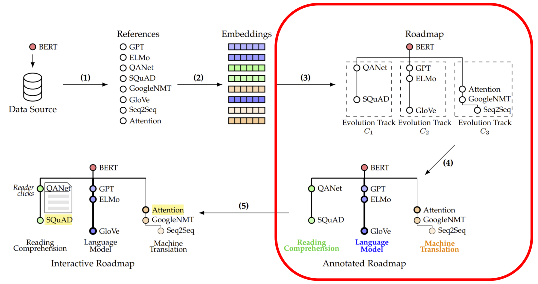
$$
\begin{aligned}
KL(C_t||l) &= \sum_{word_i} p(word_i|C_t) \log \frac{p(word_i|C_t)}{p(word_i|l)} \\
&= -\sum_{word_i} p(word_i|C_t) \log \frac{p(word_i, l|\cdot)}{p(word_i|\cdot)p(l|\cdot)} \\
&\quad + KL(C_t||\cdot) + \sum_{word_i} p(word_i|C_t) \log \frac{p(word_i|l,\cdot)}{p(word_i|l)} \\
&= -\sum_{word_i} p(word_i|C_t) PMI(word_i, l|\cdot) \\
&\quad + KL(C_t||\cdot) - Bias(l,\cdot) \\
&= -\mathbb{E}_{C_t}[PMI(word, l|\cdot)] + KL(C_t||\cdot) - Bias(l,\cdot)
\end{aligned}
$$

$$
Score(l, C_t) = (1 + \frac{\mu}{N_t - 1})\mathbb{E}_{C_t}[PMI(word, l|\cdot)]
$$

$$
- \frac{\mu}{N_t - 1} \sum_{j=1}^{N_t} \mathbb{E}_{C_j}[PMI(word, l|\cdot)] + \phi \mathbb{E}_q[PMI(word, l|\cdot)]
$$

Mei Q, Shen X, Zhai C. Automatic labeling of multinomial topic models[C]. KDD. 2007.

Method

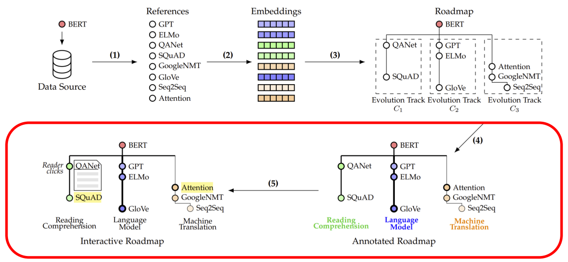


- 4. Generate labels and importance scores
  - Importance scores generation
    - Directly use the kernel weight in clustering
    - Evolution track importance is the sum of all paper importance scores inside

$$w_{p_i} = K_{i_q i}$$

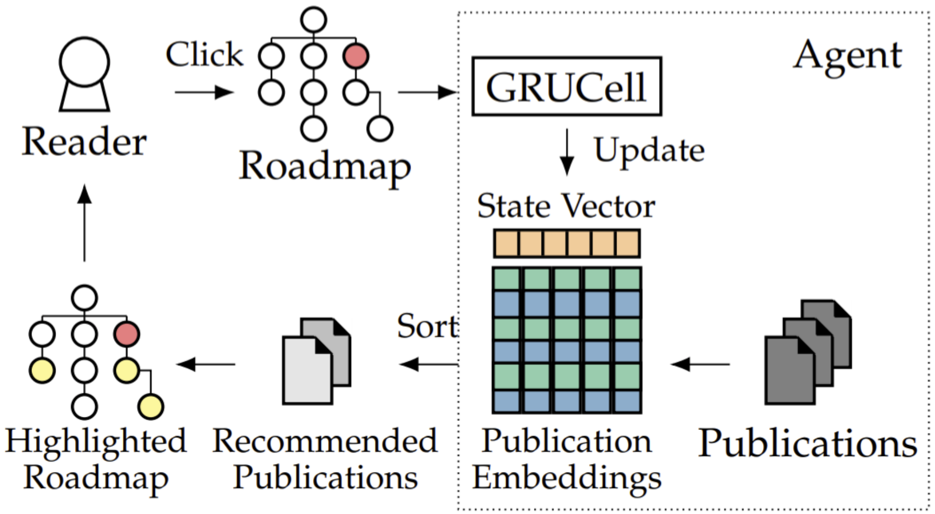
$$w_{C_t} = \sum_{p_i \in C_t} w_{p_i}$$

Method



- 5. Interact with users
  - Design a recommendation module to highlight most related papers
  - Two strategies
    - When no user data available, recommend paper by selecting papers with the closest embeddings
    - When user data available, train a reinforcement learning model to make dynamic recommendation to maximize the expected clicks.

$$\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}[\mathcal{R}(\tau)] \approx \sum_{\tau\sim\pi_\theta} \sum_{t=0}^{|\tau|} \hat{r}_t \nabla_\theta \log \alpha_\theta(p_i|s_t)$$



Chen M, Beutel A, Covington P, et al. Top-k off-policy correction for a reinforce recommender system[C]. WSDM '19. 2018.

Evaluations

- Dataset
  - KDD & ACL 2019~2020 conference papers as source paper
  - Use SemanticScholar data source to generate papers

Main configurations for experiments

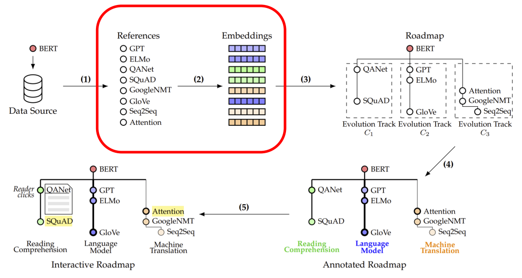
Symbol	Description	Value
N_p	Number of publications for each roadmap	100
N_t	Number of evolution tracks	6
N_l	Number of labels for each evolution track	5
k	Number of recommended publications	5

Dataset statistics for evaluations.

Dataset	Papers[1]	Retrieved References[2]	Citation Links[3]
KDD	534	126,499	1,663,063
ACL	679	88,876	3,202,684

[1] *Papers* refer to the publications used as the query publication q. This is also the number of *evolution roadmaps* we tested.
[2] *Retrieved References* refer to the first-order and second-order references we retrieved from Semantic Scholar, which are not necessarily inside the same conference with the query publications.
[3] *Citation Links* indicate how many links are considered between publications. This is the number of links we used while using PageRank to select related papers.

Evaluations



- Neighborhood Similarity
  - Evaluate the quality of paper embeddings
  - Use neighborhood similarity as ground truth
    - If two papers share similar neighborhoods (have lots of reference or cited papers in common), they should have close paper embeddings
  - Use Spearman correlation coefficient to measure

$$\mathcal{N}(p_i) = \{p \mid \mathtt{cite}(p, p_i) \vee \mathtt{cite}(p_i, p)\} \cup \{p_i\}$$

$$sim_{\mathcal{N}}(p_i, p_j) = \frac{|\mathcal{N}(p_i) \cap \mathcal{N}(p_j)|}{\sqrt{|\mathcal{N}(p_i)| \cdot |\mathcal{N}(p_j)|}}$$

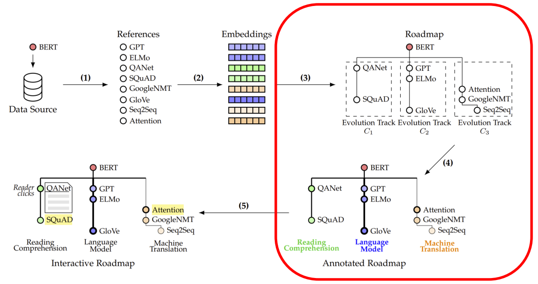
Neighborhood Similarity Experiment

Method	KDD	ACL
TF-IDF[1]	0.50	0.49
S-BERT[2]	0.41	0.36
ProNE[3]	0.72	0.75
node2vec[4]	0.65	0.64
TF-IDF+S-BERT	0.41	0.36
TF-IDF+ProNE	0.78	0.79
S-BERT+ProNE	0.75	0.77
TF-IDF+S-BERT+ProNE	**0.81**	**0.82**

[1] For TF-IDF, we select top frequent 2000 features and use n-grams ranging from 1 to 5.
[2] For S-BERT, we use the pre-trained model of bert-base-nli-stsb-mean-tokens.
[3] For ProNE, the embedding dimension is 32 and the order of Chebyshev expansion is 10, according to [29].
[4] For node2vec, the embedding dimension is 32. Walk length and number of walks are set to be 20 and 60, respectively. The window size is 5.

Evaluation

- Co-mention and MST Trials
  - Evaluate the quality of roadmap structure
  - Co-mention: reference papers mentioned together in the source paper should be clustered together
  - MST: Connecting papers into timelines should not break too much close relationships between papers



They either rely on pattern-based methods [14, 32] which extract hierarchical relation leveraging linguistic features, or clustering-based methods [11, 42], which cluster concepts to induce an implicit hierarchy.

Example: [14] and [32] is strongly related, and weakly related to [11]

Co-mention and MST Trials

Method	Co-mention* KDD	Co-mention* ACL	MST KDD	MST ACL
w/o supervision				
Hierarchical	0.63, 0.48	0.66, 0.51	0.55	0.57
Spectral	0.62, 0.48	0.65, 0.51	0.55	0.57
K-means**	**0.73**, **0.57**	0.77, 0.60	0.57	0.59
Kernel k-means	**0.73**, 0.56	**0.78**, **0.61**	0.57	0.59
w/ supervision				
Strong Co-mention	0.81, 0.58	0.85, 0.64	0.57	0.59
Weak Co-mention	0.84, 0.73	0.88, 0.77	0.57	0.59

* The co-mention columns include strong co-mention hit rate (left) and weak co-mention hit rate (right).

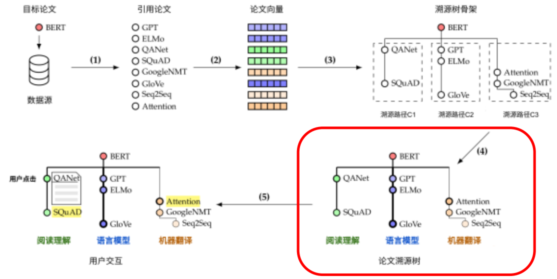
** K-means is also a special case for kernel k-means, setting $\alpha = \beta = 0$.

Evaluation

- Inverse Label Distance and Overlap Rate
  - Evaluate the quality of generated labels
  - ILD: For each evolution track, reference papers inside should be mentioned at a close position to the label
  - Overlap: Different evolution tracks, should have different labels

$$\mathtt{ILD}(G) = \frac{1}{N_t} \sum_{t=0}^{N_t - 1} \max_j \frac{1}{|C_t|} \sum_{p_i \in C_t} \frac{1}{dis_{ij}}$$

$$\mathtt{Overlap}(G) = 1 - \frac{|\{l_{tj} \mid \forall t, j\}|}{N_t N_l}$$



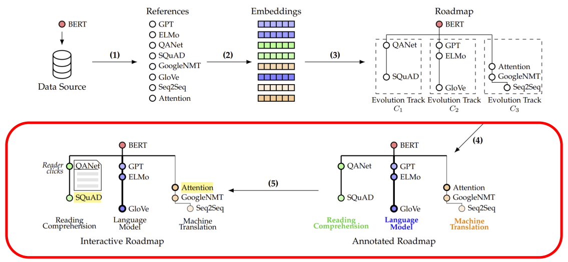
Shortcut Connections. Practices and theories that lead to shortcut connections [2, 34, 49] have been studied for a long time. An early practice of training multi-layer perceptrons (MLPs) is to add a linear layer connected from the network input to the output [34, 49]. In [44, 24], a few intermediate layers are directly connected to auxiliary classifiers for addressing vanishing/exploding gradients. The papers of [39, 38, 31, 47] propose methods for centering layer responses, gradients, and propagated errors, implemented by shortcut connections. In [44], an "inception" layer is composed of a shortcut branch and a few deeper branches.

Example: [2, 34, 49] is closely related to Shortcut Connections.

Inverse Label Distance and Overlap Rate for labeling

Method	ILD		Overlap	
	KDD	ACL	KDD	ACL
Baseline Methods				
Frequency	0.68	0.69	0.14	0.21
TF-IDF	0.66	0.64	**0.07**	**0.09**
Proposed Methods				
$\mu = 0.8, \phi = 0.1$	0.75	0.71	0.13	0.16
$\mu = 0.0, \phi = 0.1$	0.78	0.73	0.40	0.43
$\mu = 0.8, \phi = 0.0$	0.73	0.69	0.11	0.14
$\mu = 0.8, \phi = 0.5$	**0.79**	**0.76**	0.24	0.27

Evaluation



- User Feedback
  - Importance Evaluation
    - Papers with more clicks should receive higher important scores
  - Recommendation Evaluation
    - The CTR for the recommended papers
  - Human Evaluation
    - 3.68/5 (Baseline) vs. 3.82/5 (Proposed)

Importance Evaluation with User Click

Method	Spearman	NDCG@5	NDCG@20
Citation Number	-0.23	0.19	0.28
Out-degrees	-0.15	0.21	0.36
In-degrees	0.36	0.56	0.65
PageRank	0.38	0.61	0.70
Importance Score	**0.41**	**0.87**	**0.79**

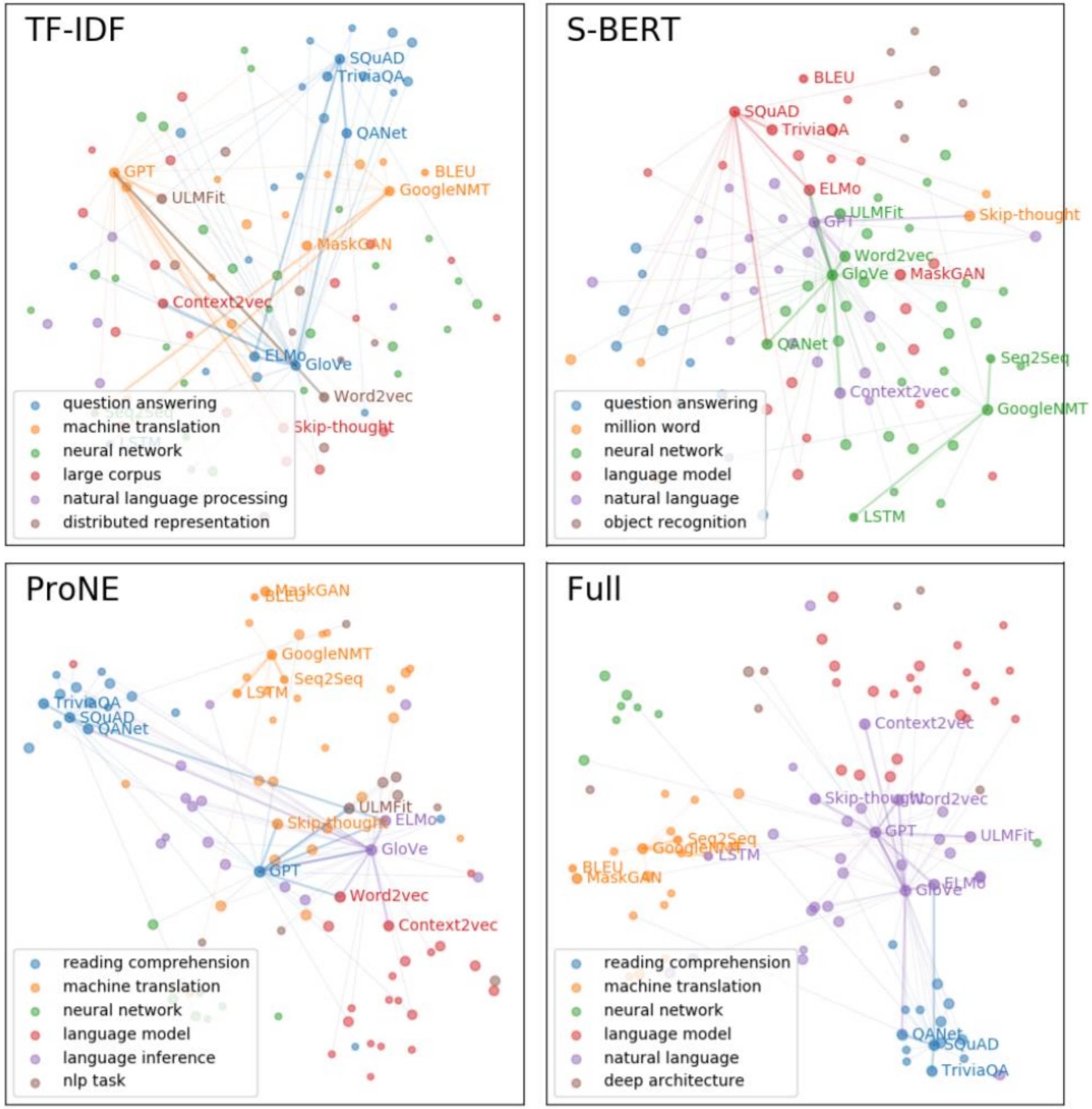
The out-degrees, in-degrees and PageRank scores are all calculated based on the subgraph of citation network. The subgraph has N_p papers as nodes and all their internal citation links.

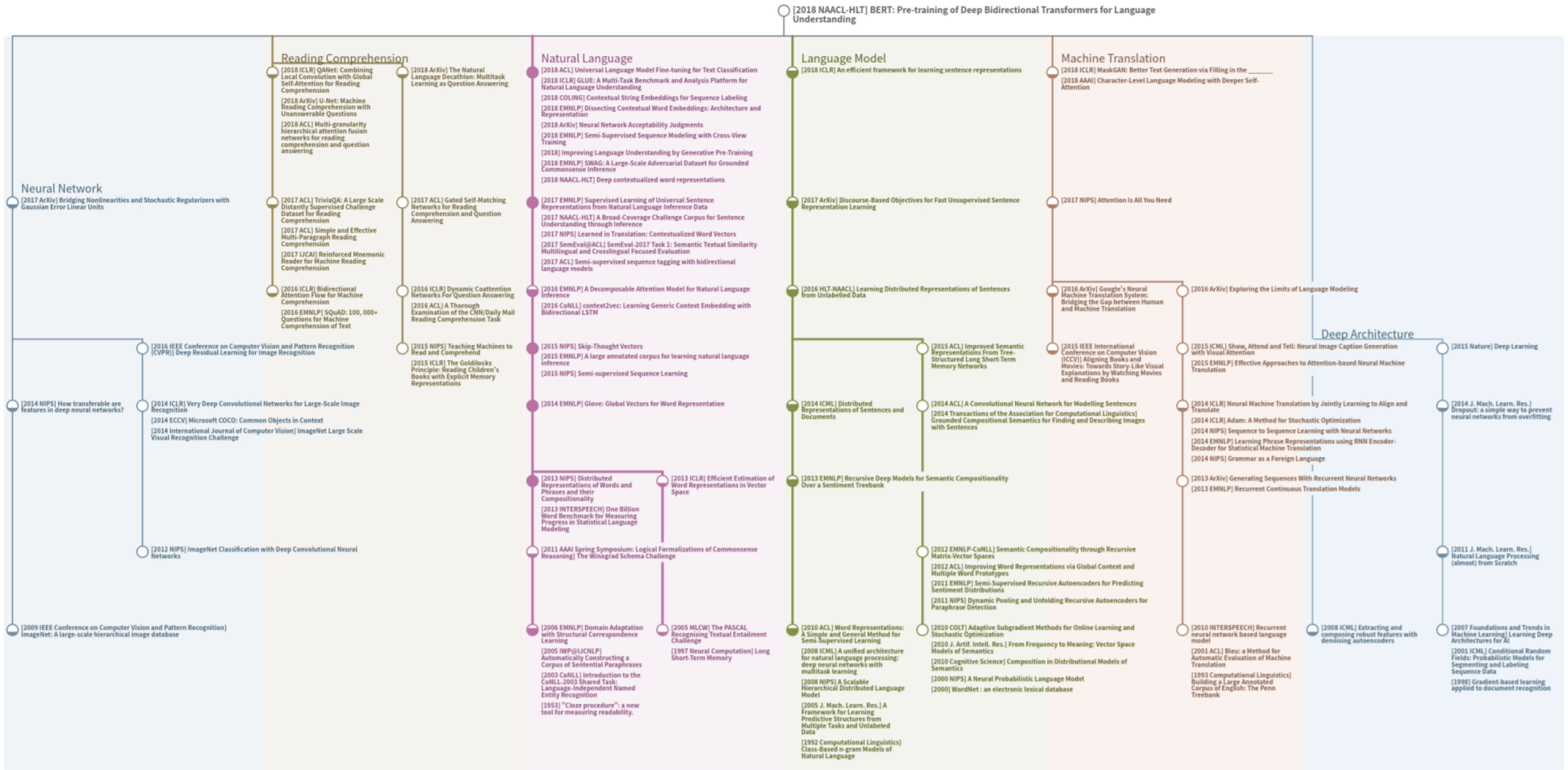
Average Rewards for Dynamic Recommendation

Roadmap	Models	
	Baseline	REINFORCE
BERT	0.32	**0.66**
GAN	0.28	**0.40**
ResNet	0.67	**0.78**
GraphSage	0.75	**0.83**

Case Study

- Paper Embeddings
  - TF-IDF embedding cannot align NLP with “natural language processing” and therefore cannot categorize ULMFit properly.
  - S-BERT cluster QANet into “machine learning” due to its use of lots of machine translation ideas such as backtranslation
  - ProNE is hard to deal papers with high citations such as GPT or GloVe

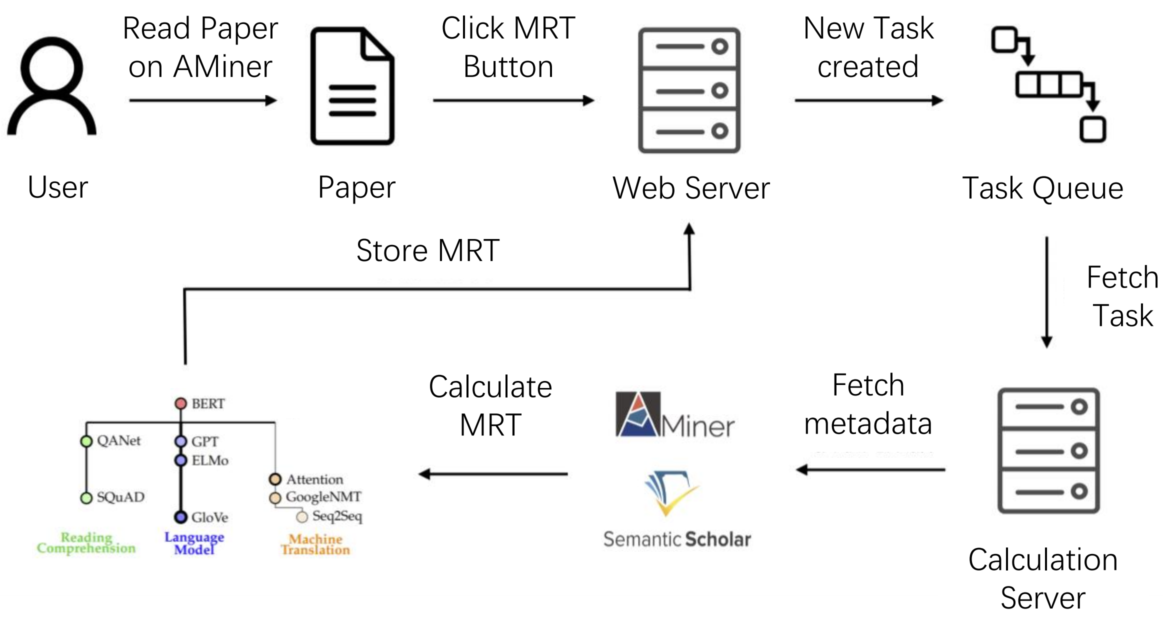




[2018 NAACL-HLT] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Neural Network
[2017 ArXiv] Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units
[2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)] Deep Residual Learning for Image Recognition
[2014 NIPS] How transferable are features in deep neural networks?
[2014 ICLR] Very Deep Convolutional Networks for Large-Scale Image Recognition
[2014 ECCV] Microsoft COCO: Common Objects in Context
[2014 International Journal of Computer Vision] ImageNet Large Scale Visual Recognition Challenge
[2012 NIPS] ImageNet Classification with Deep Convolutional Neural Networks
[2009 IEEE Conference on Computer Vision and Pattern Recognition] ImageNet: A large-scale hierarchical image database
Reading Comprehension
[2018 ICLR] QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension
[2018 ArXiv] U-Net: Machine Reading Comprehension with Unanswerable Questions
[2018 ACL] Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering
[2018 ArXiv] The Natural Language Decathlon: Multitask Learning as Question Answering
[2017 ACL] TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension
[2017 ACL] Simple and Effective Multi-Paragraph Reading Comprehension
[2017 IJCAI] Reinforced Mnemonic Reader for Machine Reading Comprehension
[2017 ACL] Gated Self-Matching Networks for Reading Comprehension and Question Answering
[2016 ICLR] Bidirectional Attention Flow for Machine Comprehension
[2016 EMNLP] SQuAD: 100, 000+ Questions for Machine Comprehension of Text
[2016 ICLR] Dynamic Coattention Networks For Question Answering
[2016 ACL] A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task
[2015 NIPS] Teaching Machines to Read and Comprehend
[2015 ICLR] The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations
Natural Language
[2018 ACL] Universal Language Model Fine-tuning for Text Classification
[2018 ICLR] GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding
[2018 COLING] Contextual String Embeddings for Sequence Labeling
[2018 EMNLP] Dissecting Contextual Word Embeddings: Architecture and Representation
[2018 ArXiv] Neural Network Acceptability Judgments
[2018 EMNLP] Semi-Supervised Sequence Modeling with Cross-View Training
[2018] Improving Language Understanding by Generative Pre-Training
[2018 EMNLP] SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference
[2018 NAACL-HLT] Deep contextualized word representations
[2017 EMNLP] Supervised Learning of Universal Sentence Representations from Natural Language Inference Data
[2017 NAACL-HLT] A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference
[2017 NIPS] Learned in Translation: Contextualized Word Vectors
[2017 SemEval@ACL] SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation
[2017 ACL] Semi-supervised sequence tagging with bidirectional language models
[2016 EMNLP] A Decomposable Attention Model for Natural Language Inference
[2016 CoNLL] context2vec: Learning Generic Context Embedding with Bidirectional LSTM
[2015 NIPS] Skip-Thought Vectors
[2015 EMNLP] A large annotated corpus for learning natural language inference
[2015 NIPS] Semi-supervised Sequence Learning
[2014 EMNLP] Glove: Global Vectors for Word Representation
[2013 NIPS] Distributed Representations of Words and Phrases and their Compositionality
[2013 INTERSPEECH] One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling
[2013 ICLR] Efficient Estimation of Word Representations in Vector Space
[2011 AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning] The Winograd Schema Challenge
[2006 EMNLP] Domain Adaptation with Structural Correspondence Learning
[2005 IWP@IJCNLP] Automatically Constructing a Corpus of Sentential Paraphrases
[2003 CoNLL] Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition
[1953] "Cloze procedure": a new tool for measuring readability.
[2005 MLCW] The PASCAL Recognising Textual Entailment Challenge
[1997 Neural Computation] Long Short-Term Memory
Language Model
[2018 ICLR] An efficient framework for learning sentence representations
[2017 ArXiv] Discourse-Based Objectives for Fast Unsupervised Sentence Representation Learning
[2016 HLT-NAACL] Learning Distributed Representations of Sentences from Unlabelled Data
[2015 ACL] Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks
[2014 ICML] Distributed Representations of Sentences and Documents
[2014 ACL] A Convolutional Neural Network for Modelling Sentences
[2014 Transactions of the Association for Computational Linguistics] Grounded Compositional Semantics for Finding and Describing Images with Sentences
[2013 EMNLP] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
[2012 EMNLP-CoNLL] Semantic Compositionality through Recursive Matrix-Vector Spaces
[2012 ACL] Improving Word Representations via Global Context and Multiple Word Prototypes
[2011 EMNLP] Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions
[2011 NIPS] Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection
[2010 ACL] Word Representations: A Simple and General Method for Semi-Supervised Learning
[2008 ICML] A unified architecture for natural language processing: deep neural networks with multitask learning
[2008 NIPS] A Scalable Hierarchical Distributed Language Model
[2005 J. Mach. Learn. Res.] A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data
[1992 Computational Linguistics] Class-Based n-gram Models of Natural Language
[2010 COLT] Adaptive Subgradient Methods for Online Learning and Stochastic Optimization
[2010 J. Artif. Intell. Res.] From Frequency to Meaning: Vector Space Models of Semantics
[2010 Cognitive Science] Composition in Distributional Models of Semantics
[2000 NIPS] A Neural Probabilistic Language Model
[2000] WordNet : an electronic lexical database
Machine Translation
[2018 ICLR] MaskGAN: Better Text Generation via Filling in the ______
[2018 AAAI] Character-Level Language Modeling with Deeper Self-Attention
[2017 NIPS] Attention Is All You Need
[2016 ArXiv] Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation
[2016 ArXiv] Exploring the Limits of Language Modeling
[2015 IEEE International Conference on Computer Vision (ICCV)] Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books
[2015 ICML] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
[2015 EMNLP] Effective Approaches to Attention-based Neural Machine Translation
[2014 ICLR] Neural Machine Translation by Jointly Learning to Align and Translate
[2014 ICLR] Adam: A Method for Stochastic Optimization
[2014 NIPS] Sequence to Sequence Learning with Neural Networks
[2014 EMNLP] Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
[2014 NIPS] Grammar as a Foreign Language
[2013 ArXiv] Generating Sequences With Recurrent Neural Networks
[2013 EMNLP] Recurrent Continuous Translation Models
[2010 INTERSPEECH] Recurrent neural network based language model
[2001 ACL] Bleu: a Method for Automatic Evaluation of Machine Translation
[1993 Computational Linguistics] Building a Large Annotated Corpus of English: The Penn Treebank
Deep Architecture
[2015 Nature] Deep Learning
[2014 J. Mach. Learn. Res.] Dropout: a simple way to prevent neural networks from overfitting
[2011 J. Mach. Learn. Res.] Natural Language Processing (almost) from Scratch
[2008 ICML] Extracting and composing robust features with denoising autoencoders
[2007 Foundations and Trends in Machine Learning] Learning Deep Architectures for AI
[2001 ICML] Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data
[1998] Gradient-based learning applied to document recognition

System Deploy

- Deployed to AMiner
  - Over 7,000 users
  - About 20,000 access (Mar. 2021)
- Async online service
  - Single MRT generation requires tens of seconds
  - Mostly spends on accessing Web API to retrieve paper data
  - When cache is available, MRT can be calculated in 2~3 seconds with the help of GPU
  - If S-BERT is disabled, the MRT can be generated more fast even without GPU



Average Running Time for Each Algorithm

Algorithm	Time(s)
Select reference papers (PageRank)	$0.51^{0.25}$
Encode papers (TF-IDF, S-BERT, ProNE)	$1.39^{0.33}$
Cluster papers (Kernel k-means)	$0.48^{0.34}$
Generate labels (Automatic Labeling)	$0.29^{0.09}$