Part IV: Weakly-Supervised Text Classification: Embeddings with Less Human Effort

KDD 2021 Tutorial

On the Power of Pre-Trained Text Representations: Models and Applications in Text Mining

Yu Meng, Jiaxin Huang, Yu Zhang, Jiawei Han

Computer Science, University of Illinois at Urbana-Champaign

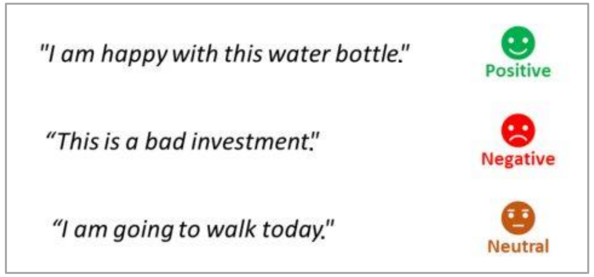
August 14, 2020

Outline

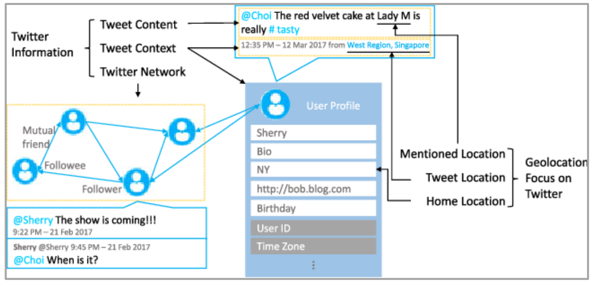
- What Weakly-Supervised Text Classification Is, and Why It Matters
- Flat Text Classification
  - Embedding: WeSTClass [CIKM’18]
  - Pre-trained LM: ConWea [ACL’20], LOTClass [EMNLP’20], X-Class [NAACL’21]
- Text Classification with Taxonomy Information
  - Embedding: WeSHClass [AAAI’19]
  - Pre-trained LM: TaxoClass [NAACL’21]
- Text Classification with Metadata Information
  - Embedding: MetaCat [SIGIR’20], HIMECat [WSDM’21]

Text Classification

- Given a set of text units (e.g., documents, sentences) and a set of categories, the task is to assign relevant category/categories to each text unit
- Text Classification has a lot of downstream applications

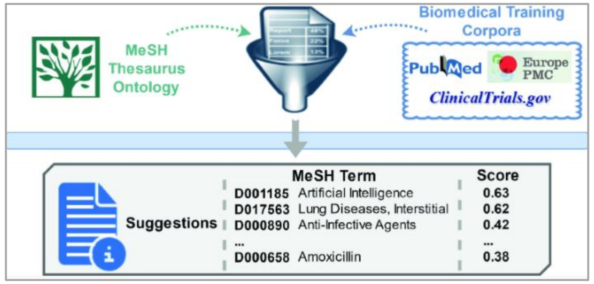


Sentiment Analysis

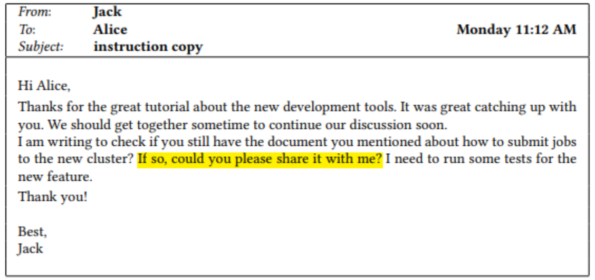


Location Prediction

News Topic Classification



Paper Topic Classification



From: **Jack**
To: **Alice** **Monday 11:12 AM**
Subject: **instruction copy**

Hi Alice,

Thanks for the great tutorial about the new development tools. It was great catching up with you. We should get together sometime to continue our discussion soon.

I am writing to check if you still have the document you mentioned about how to submit jobs to the new cluster? If so, could you please share it with me? I need to run some tests for the new feature.

Thank you!

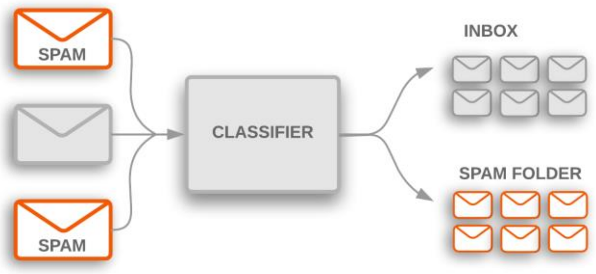
Best,
Jack

Email Intent Identification

Hate Speech Detection

Different Text Classification Settings: Single-Label vs. Multi-Label

- **Single-label**: Each document belongs to one category.
  - Ex. Spam Detection



- **Multi-label**: Each document has multiple relevant labels.
  - Ex. Paper Topic Classification

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Abstract

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5 (7.7 point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

Related Topics

Question answering | Language model | Natural language understanding | Named-entity recognition | SemEval | Inference | Winograd Schema Challenge | Sequence labeling | Artificial intelligence | Computer science | Transformer (machine learning model) | View Less

https://academic.microsoft.com/paper/2963341956/

Different Text Classification Settings: Flat vs. Hierarchical

- **Flat**: All labels are at the same granularity level.
  - Ex. Sentiment Analysis of E-Commerce Reviews (1-5 stars)

It works, it's nice, comfortable, and easy to type on. Not loud (unless you're a key pounder)

This keyboard works. It's comfortable, sensitive enough for touch typers, very quiet by comparison to other mechanicals (unless, of course, you're a 'key pounder'), and the lit keys are excellent for people like me who tend to prefer to work in a cave-like environment.

https://www.amazon.com/gp/product/B089YFHYYS/

- **Hierarchical**: Labels are organized into a hierarchy representing their parent-child relationship.
  - Ex. Paper Topic Classification (the arXiv category taxonomy)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

Subjects: Computation and Language (cs.CL)
Cite as: arXiv:1810.04805 [cs.CL]
(or arXiv:1810.04805v2 [cs.CL] for this version)

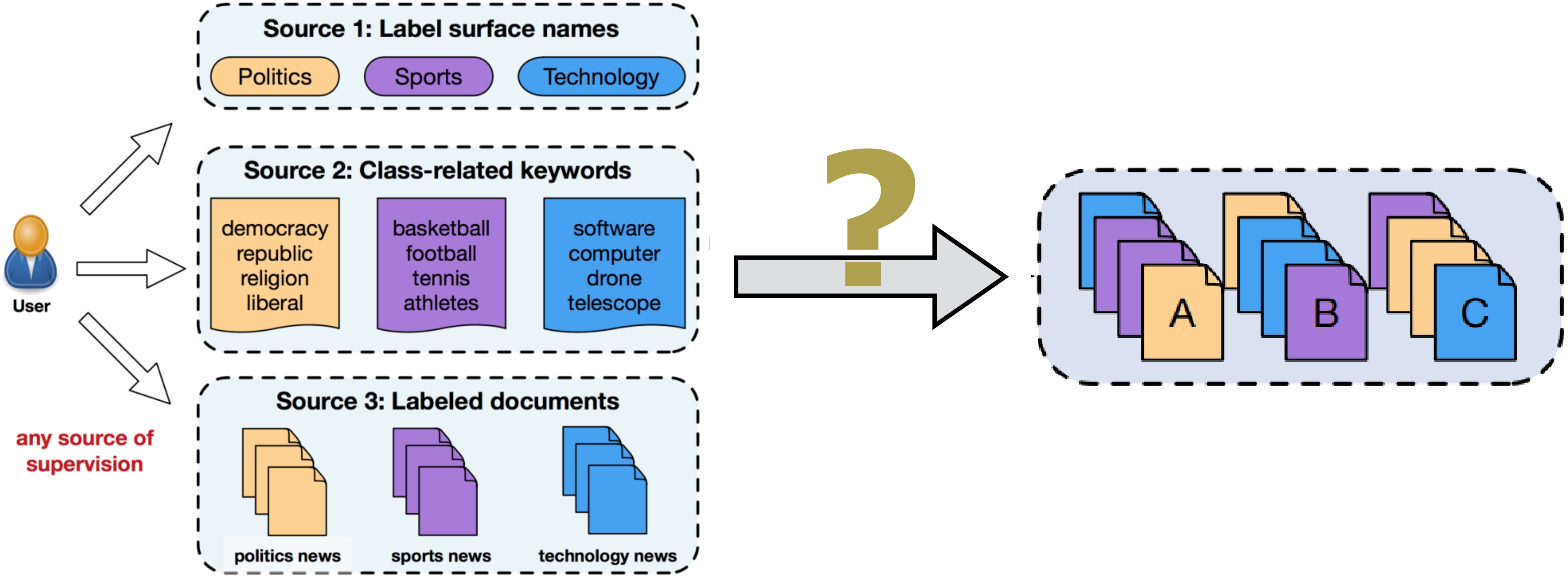
https://arxiv.org/abs/1810.04805

Weakly-Supervised Text Classification: Motivation

- Supervised text classification models (especially recent deep neural models) rely on a significant number of manually labeled training documents to achieve good performance.
- Collecting such training data is usually expensive and time-consuming. In some domains (e.g., scientific papers), annotations must be acquired from domain experts, which incurs additional cost.
- While users cannot afford to label sufficient documents for training a deep neural classifier, they can provide a small amount of seed information:
  - Category-related keywords
  - A small number of labeled documents

Weakly-Supervised Text Classification: Definition

- Text classification without massive human-annotated training data
  - **Keyword-level weak supervision**: label names or a few relevant keywords
  - **Document-level weak supervision**: a small set of labeled docs



General Ideas to Perform Weakly-Supervised Text Classification

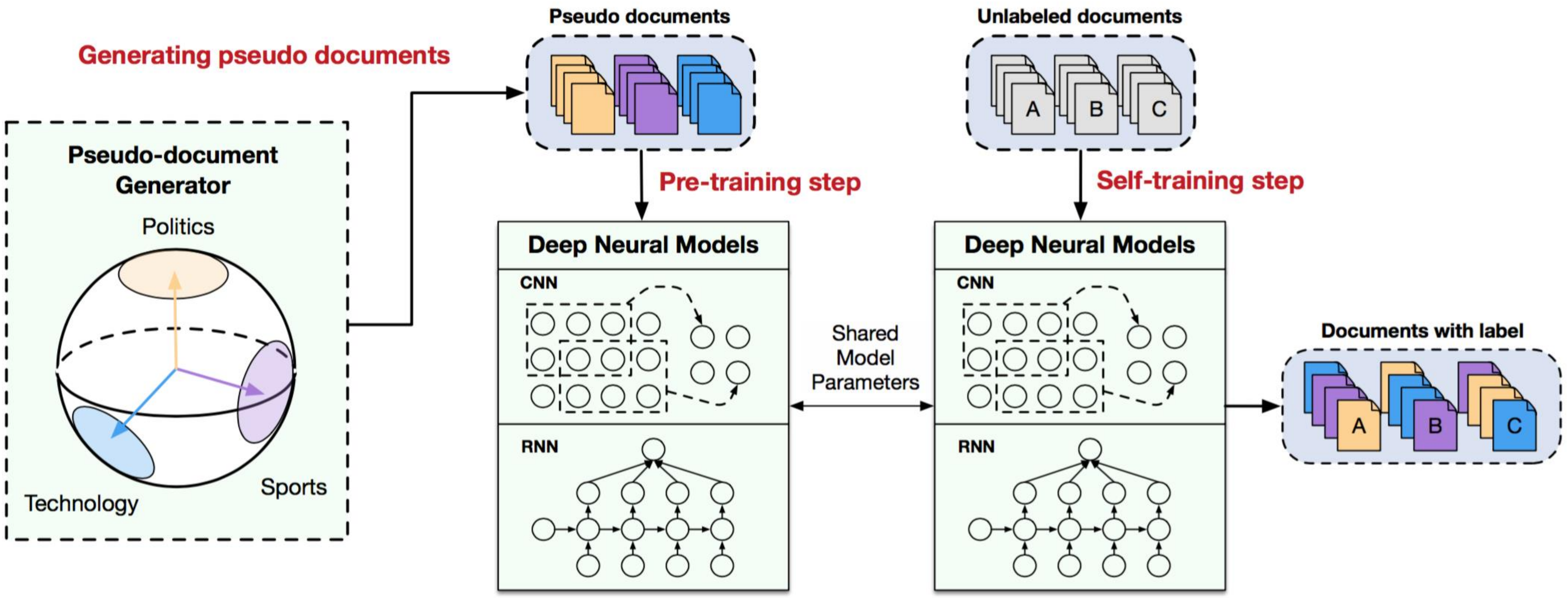
- Joint representation learning
  - Put words, labels, and/or documents into the same latent space using embedding learning or pre-trained language models
- Pseudo training data generation
  - Retrieve some unlabeled documents or synthesize some artificial documents using text embeddings or contextualized representations
  - Give them pseudo labels to train a text classifier
- Transfer the knowledge of pre-trained language models to classification tasks

Outline

- What Weakly-Supervised Text Classification Is, and Why It Matters
- Flat Text Classification
  - Embedding: WeSTClass [CIKM’18]
  - Pre-trained LM: ConWea [ACL’20], LOTClass [EMNLP’20], X-Class [NAACL’21]
- Text Classification with Taxonomy Information
  - Embedding: WeSHClass [AAAI’19]
  - Pre-trained LM: TaxoClass [NAACL’21]
- Text Classification with Metadata Information
  - Embedding: MetaCat [SIGIR’20], HIMECat [WSDM’21]

WeSTClass: Pseudo Training Data + Self-Training

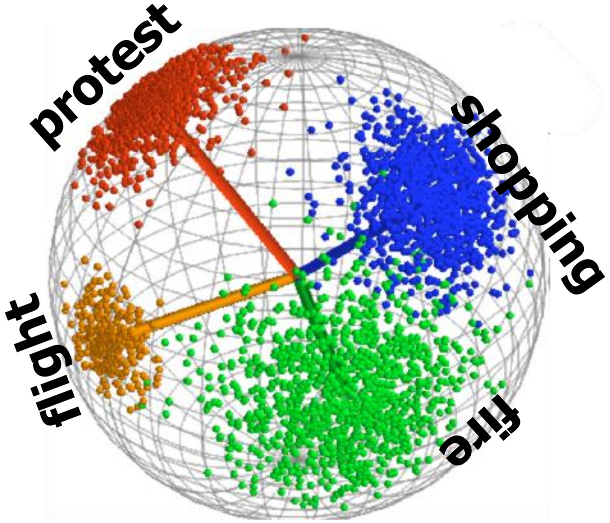
- Embed all words (including label names and keywords) into the same space
- Pseudo document generation: generate pseudo documents from seeds
- Self-training: train deep neural nets (CNN, RNN) with bootstrapping



Meng, Y., Shen, J., Zhang, C., & Han, J. “Weakly-supervised neural text classification”, CIKM’18.
Applicable to both keyword-level and document-level supervision.

WeSTClass: Pseudo Document Generation

- Fit a von-Mishes Fisher distribution for each category according to the keywords
  - Category name as supervision? Find nearest words as keywords
  - A few documents as supervision? Retrieve words with high TF-IDF scores
- Sample bag-of-keywords as pseudo documents for each class



Mean direction

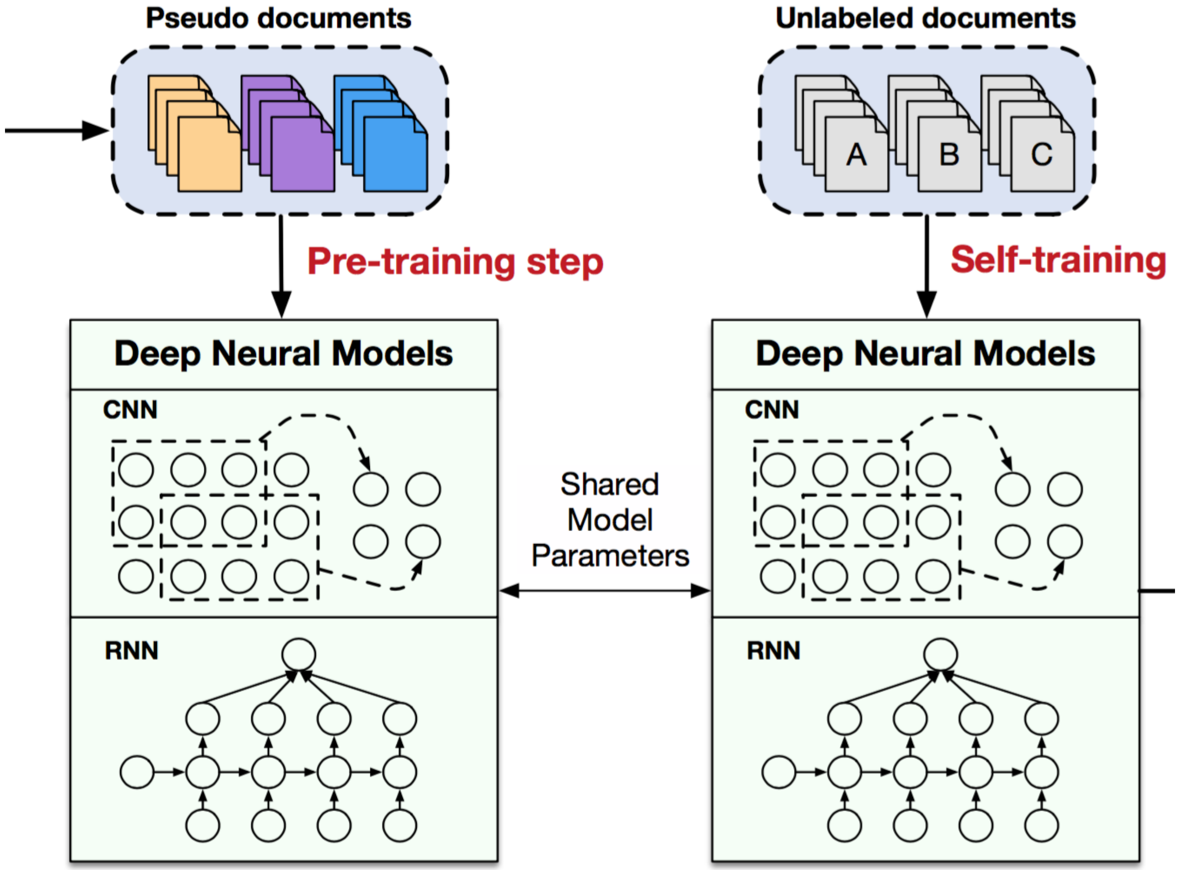
Concentration parameter

$$p(\mathbf{x}|\mu, \kappa) = C_D(\kappa)\exp(\kappa\mu^T\mathbf{x})$$

$$C_D(\kappa) = \frac{\kappa^{D/2-1}}{I_{D/2-1}(\kappa)}$$

WeSTClass: Self-Training Deep Neural Nets

- **Pre-training**: Use pseudo documents to initialize DNNs (e.g., CNN, RNN)
- **Self-training**: Iteratively refine DNNs in a self-boosting fashion



new score of
label j for document i

$$l_{ij} = \frac{y_{ij}^2 / f_j}{\sum_{j'} y_{ij'}^2 / f_{j'}}$$

WeSTClass: Experiment Results

- Datasets: (1) NYT, (2) AG's News, (3) Yelp
- Evaluation: use different types of weak supervision and measure accuracies

Macro-F1 scores:

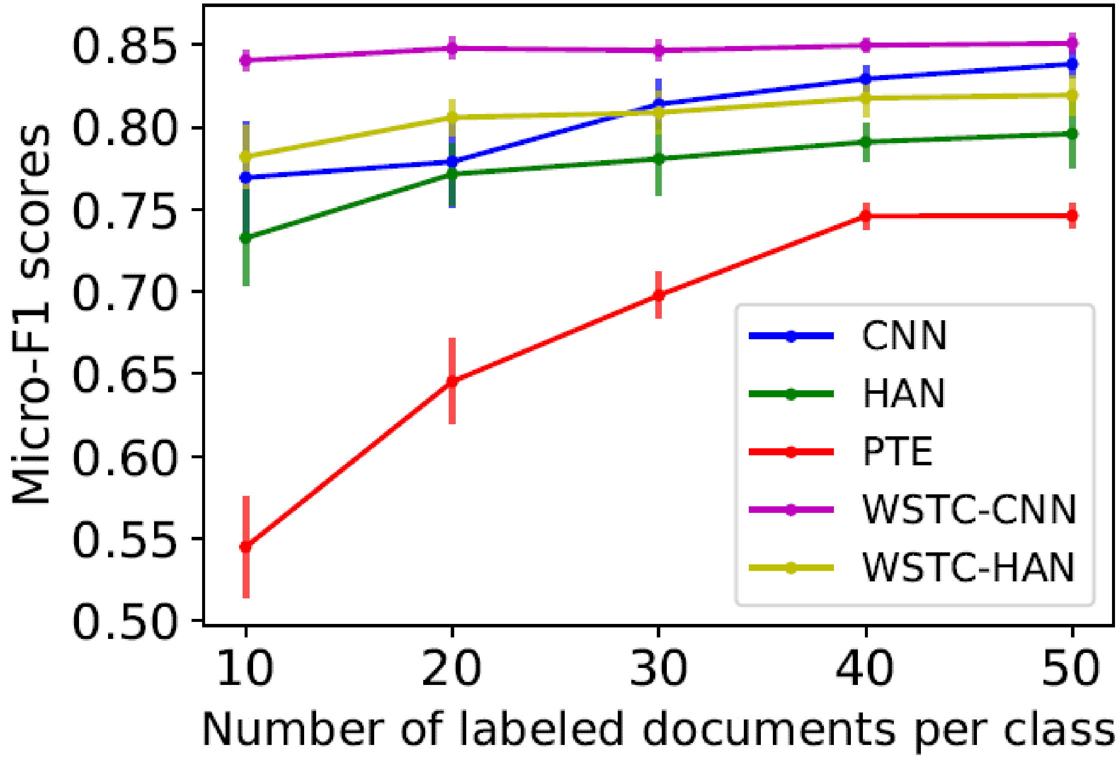
Methods	The New York Times			AG's News			Yelp Review		
	LABELS	KEYWORDS	DOCS	LABELS	KEYWORDS	DOCS	LABELS	KEYWORDS	DOCS
IR with tf-idf	0.319	0.509	-	0.187	0.258	-	0.533	0.638	-
Topic Model	0.301	0.253	-	0.496	0.723	-	0.333	0.333	-
Dataless	0.484	-	-	0.688	-	-	0.337	-	-
UNEC	0.690	-	-	0.659	-	-	0.602	-	-
PTE	-	-	0.834 (0.024)	-	-	0.542 (0.029)	-	-	0.658 (0.042)
HAN	0.348	0.534	0.740 (0.059)	0.498	0.621	0.731 (0.029)	0.519	0.631	0.686 (0.046)
CNN	0.338	0.632	0.702 (0.059)	0.758	0.770	0.766 (0.035)	0.523	0.633	0.634 (0.096)
NoST-HAN	0.515	0.213	0.823 (0.035)	0.590	0.727	0.745 (0.038)	0.731	0.338	0.682 (0.090)
NoST-CNN	0.701	0.702	0.833 (0.013)	0.534	0.759	0.759 (0.032)	0.639	0.740	0.717 (0.058)
WeSTClass-HAN	0.754	0.640	0.832 (0.028)	0.816	0.820	0.782 (0.028)	**0.769**	0.736	0.729 (0.040)
WeSTClass-CNN	**0.830**	**0.837**	**0.835 (0.010)**	**0.822**	**0.821**	**0.839 (0.007)**	0.735	**0.816**	**0.775 (0.037)**

Micro-F1 scores:

Methods	The New York Times			AG's News			Yelp Review		
	LABELS	KEYWORDS	DOCS	LABELS	KEYWORDS	DOCS	LABELS	KEYWORDS	DOCS
IR with tf-idf	0.240	0.346	-	0.292	0.333	-	0.548	0.652	-
Topic Model	0.666	0.623	-	0.584	0.735	-	0.500	0.500	-
Dataless	0.710	-	-	0.699	-	-	0.500	-	-
UNEC	0.810	-	-	0.668	-	-	0.603	-	-
PTE	-	-	0.906 (0.020)	-	-	0.544 (0.031)	-	-	0.674 (0.029)
HAN	0.251	0.595	0.849 (0.038)	0.500	0.619	0.733 (0.029)	0.530	0.643	0.690 (0.042)
CNN	0.246	0.620	0.798 (0.085)	0.759	0.771	0.769 (0.034)	0.534	0.646	0.662 (0.062)
NoST-HAN	0.788	0.676	0.906 (0.021)	0.619	0.736	0.747 (0.037)	0.740	0.502	0.698 (0.066)
NoST-CNN	0.767	0.780	0.908 (0.013)	0.553	0.766	0.765 (0.031)	0.671	0.750	0.725 (0.050)
WeSTClass-HAN	0.901	0.859	0.908 (0.019)	0.816	0.822	0.782 (0.028)	**0.771**	0.737	0.729 (0.040)
WeSTClass-CNN	**0.916**	**0.912**	**0.911 (0.007)**	**0.823**	**0.823**	**0.841 (0.007)**	0.741	**0.816**	**0.776 (0.037)**

WeSTClass: Effect of # Labeled Documents

- Compare the performances of five methods on the AG's News dataset by varying the number of labeled documents per class and



Outline

- What Weakly-Supervised Text Classification Is, and Why It Matters
- Flat Text Classification
  - Embedding: WeSTClass [CIKM'18]
  - Pre-trained LM: ConWea [ACL'20], LOTClass [EMNLP'20], X-Class [NAACL'21]
- Text Classification with Taxonomy Information
  - Embedding: WeSHClass [AAAI'19]
  - Pre-trained LM: TaxoClass [NAACL'21]
- Text Classification with Metadata Information
  - Embedding: MetaCat [SIGIR'20], HIMECat [WSDM'21]

Language Models for Weakly-Supervised Classification

- The previous approaches only use the local corpus
- Fail to take advantage of the general knowledge source (e.g., Wikipedia)
- Why general knowledge?
  - Humans can classify texts with general knowledge
  - Common linguistic features to understand texts better
  - Compensate for potential data scarcity of the local corpus
- How to use general knowledge?
  - Neural language models (e.g., BERT) are pre-trained on large-scale general knowledge texts
  - Their learned semantic/syntactic features can be transferred to downstream tasks

ConWea: Disambiguating User-Provided Keywords

- User-provided seed words may be ambiguous.
- Example:

Class	Seed words
Soccer	soccer, goal, penalty
Law	law, judge, court

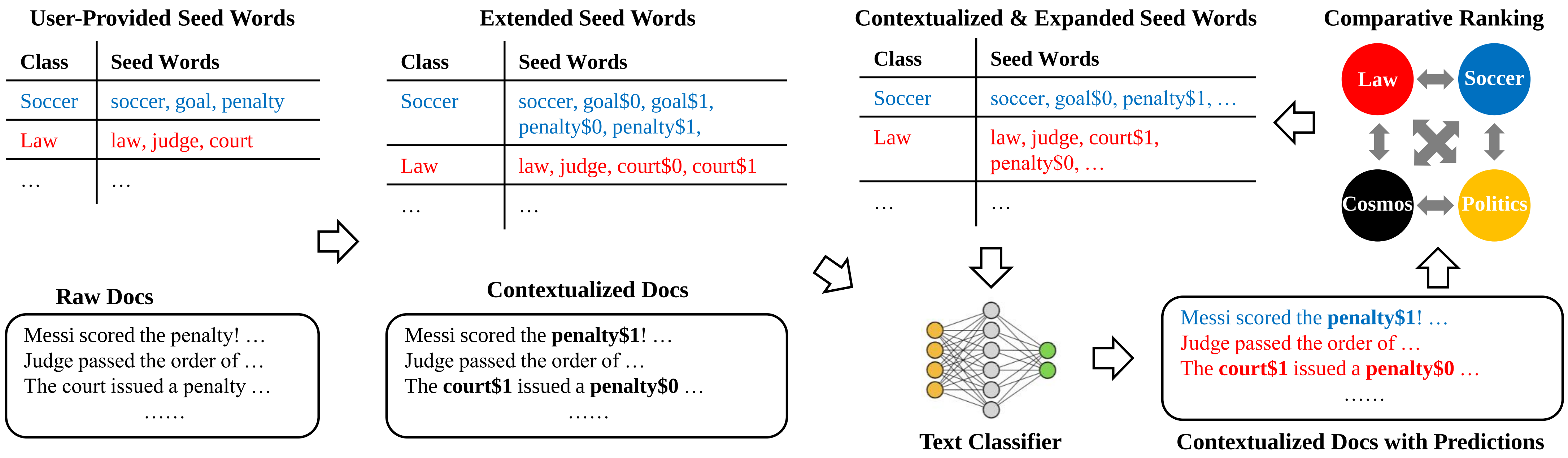
  - Classify the following sentences:
    - Messi scored the penalty.
    - John was issued a death penalty.
- Disambiguate the "senses" based on contextualized representations

Mekala, D. & Shang, J. "Contextualized Weak Supervision for Text Classification", ACL'20. **Keywords as supervision.**

ConWea-related slides credit to Jingbo Shang

ConWea: Clustering for Disambiguation

- For each word, find all its occurrences in the input corpus
  - Run BERT to get their contextualized representations
  - Run a clustering method (e.g., K-Means) to obtain clusters for different "senses"



ConWea: Experiment Results

- Ablations:
  - ConWea-NoCon: Variant of ConWea trained without contextualization.
  - ConWea-NoExpan: Variant of ConWea trained without seed expansion.
  - ConWea-WSD: Variant of ConWea with contextualization replaced by a word sense disambiguation algorithm.

		NYT				20 Newsgroup			
		5-Class (Coarse)		**25-Class** (Fine)		**6-Class** (Coarse)		**20-Class** (Fine)	
	Methods	Micro-F_1	Macro-F_1	Micro-F_1	Macro-F_1	Micro-F_1	Macro-F_1	Micro-F_1	Macro-F_1
Baselines	IR-TF-IDF	0.65	0.58	0.56	0.54	0.49	0.48	0.53	0.52
	Dataless	0.71	0.48	0.59	0.37	0.50	0.47	0.61	0.53
	Word2Vec	0.92	0.83	0.69	0.47	0.51	0.45	0.33	0.33
	Doc2Cube	0.71	0.38	0.67	0.34	0.40	0.35	0.23	0.23
	WeSTClass	0.91	0.84	0.50	0.36	0.53	0.43	0.49	0.46
	ConWea	**0.95**	**0.89**	**0.91**	**0.79**	**0.62**	**0.57**	**0.65**	**0.64**
Ablations	ConWea-NoCon	0.91	0.83	0.89	0.74	0.53	0.50	0.58	0.57
	ConWea-NoExpan	0.92	0.85	0.76	0.66	0.58	0.53	0.58	0.57
	ConWea-WSD	0.83	0.78	0.72	0.64	0.52	0.46	0.49	0.47
Upper bound	HAN-Supervised	0.96	0.92	0.94	0.82	0.90	0.88	0.83	0.83

LOTClass: Find Similar Meaning Words with Label Names

- Find topic words based on label names
  - Overcome the low semantic coverage of label names
- Use language models to predict what words can replace the label names
  - Interchangeable words are likely to have similar meanings

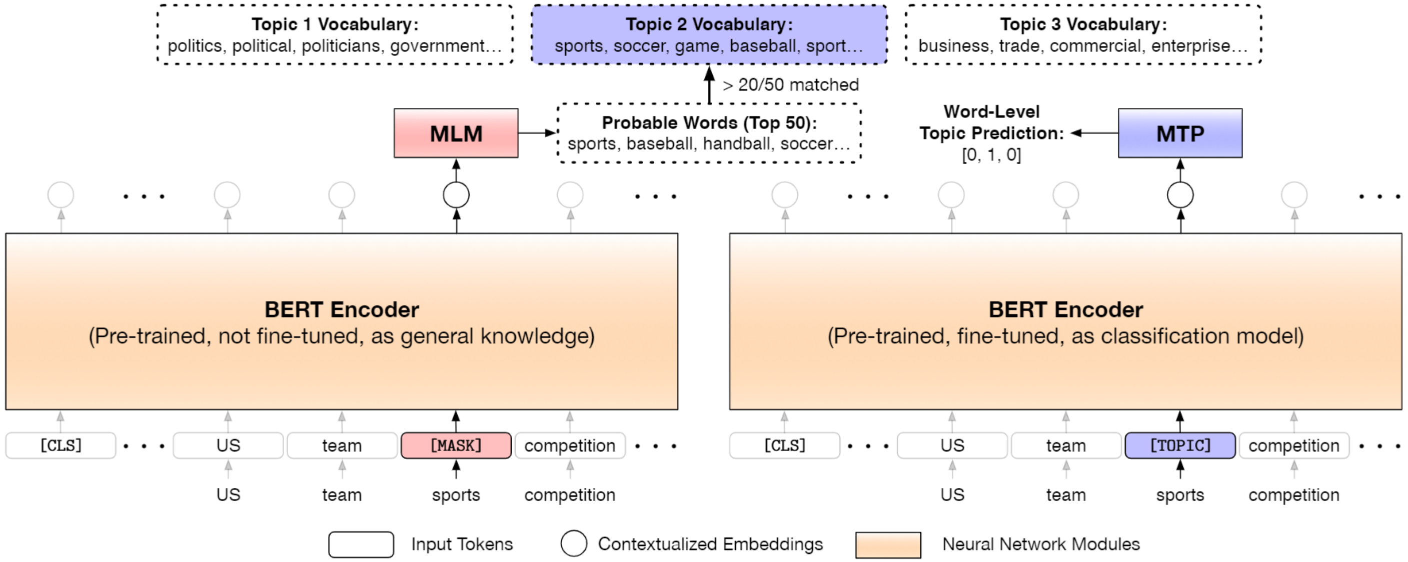
Sentence	Language Model Prediction
The oldest annual US team **sports** competition that includes professionals is not in baseball, or football or basketball or hockey. It's in soccer.	sports, baseball, handball, soccer, basketball, football, tennis, sport, championship, hockey, ...
Samsung's new SPH-V5400 mobile phone **sports** a built-in 1-inch, 1.5-gigabyte hard disk that can store about 15 times more data than conventional handsets, Samsung said.	has, with, features, uses, includes, had, is, contains, featured, have, incorporates, requires, offers, ...

Table 1: BERT language model prediction (sorted by probability) for the word to appear at the position of "sports" under different contexts. The two sentences are from *AG News* corpus.

Meng, Y., Zhang, Y., Huang, J., Xiong, C., Ji, H., Zhang, C., & Han, J. "Text Classification Using Label Names Only: A Language Model Self-Training Approach", EMNLP'20. **Category names as supervision.**

LOTClass: Contextualized Word-Level Topic Prediction

- Context-free matching of topic words is inaccurate
  - "Sports" does not always imply the topic "sports"
- Contextualized topic prediction:
  - Predict a word's implied topic under specific contexts
  - We regard a word as "topic indicative" only when its top replacing words have enough overlap with the topic vocabulary.



LOTClass: Experiment Results

- Achieve around 90% accuracy on four benchmark datasets by only using at most 3 words (1 in most cases) per class as the label name
  - Outperforming previous weakly-supervised approaches significantly
  - Comparable to state-of-the-art semi-supervised models

Supervision Type	Methods	AG News	DBPedia	IMDB	Amazon
Weakly-Sup.	Dataless (Chang et al., 2008)	0.696	0.634	0.505	0.501
	WeSTClass (Meng et al., 2018)	0.823	0.811	0.774	0.753
	BERT w. simple match	0.752	0.722	0.677	0.654
	Ours w/o. self train	0.822	0.850	0.844	0.781
	Ours	**0.864**	**0.889**	**0.894**	**0.906**
Semi-Sup.	UDA (Xie et al., 2019)	0.869	0.986	0.887	0.960
Supervised	char-CNN (Zhang et al., 2015)	0.872	0.983	0.853	0.945
	BERT (Devlin et al., 2019)	0.944	0.993	0.937	0.972

How Powerful Are Vanilla BERT Representations in Category Prediction?

- An average of BERT representations of all tokens in a sentence/document preserves domain information well

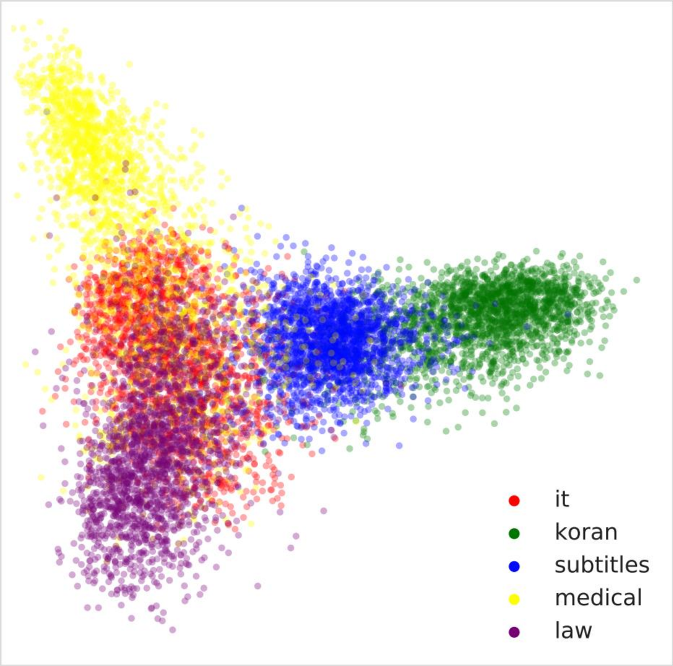


Figure 1: A 2D visualization of average-pooled BERT hidden-state sentence representations using PCA. The colors represent the domain for each sentence.

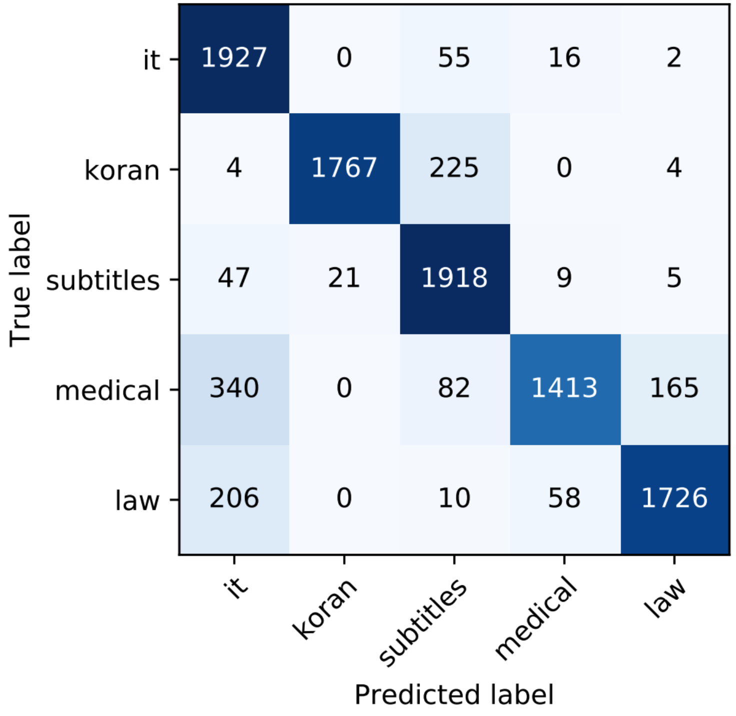
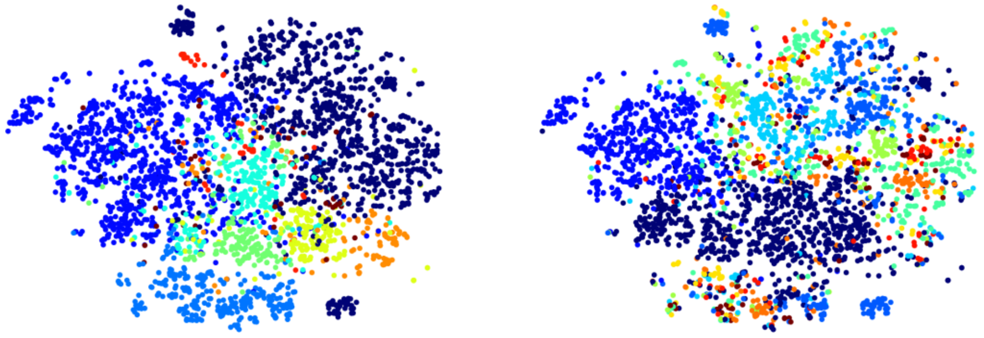


Figure 2: A confusion matrix for clustering with k=5 using BERT-base.

Aharoni, R., & Goldberg, Y. "Unsupervised domain clusters in pretrained language models." ACL'20.

X-Class: Class-Oriented BERT Representations

- A simple idea for text classification
  - Learn representations for documents
  - Set the number of clusters as the number of classes
  - Hope their clustering results are almost the same as the desired classification
- However, the same corpus could be classified differently



(a) NYT-Topics (b) NYT-Locations

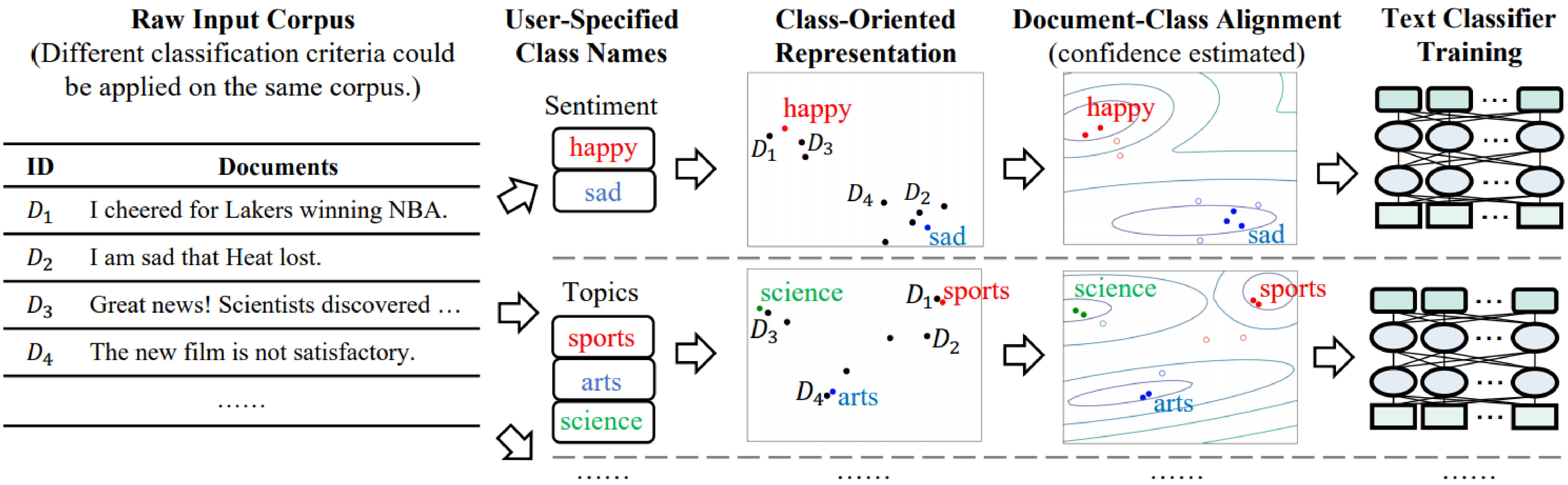
Figure 1: Visualizations of News using Average BERT Representations. Colors denote different classes.

Wang, Z., Mekala, D., & Shang, J. “X-Class: Text Classification with Extremely Weak Supervision”, NAACL’21. **Category Names as supervision.**

X-Class-related slides credit to Jingbo Shang

X-Class: Class-Oriented BERT Representations

- Clustering for classification based on class-oriented representations



X-Class: Experiment Results

- WeSTClass & ConWea consume at least 3 seed words per class
- LOTClass & X-Class use category names only

	AGNews	20News	NYT-Small	NYT-Topic	NYT-Location	Yelp	DBpedia
Corpus Domain	News	News	News	News	News	Reviews	Wikipedia
Class Criterion	Topics	Topics	Topics	Topics	Locations	Sentiment	Ontology
# of Classes	4	5	5	9	10	2	14
# of Documents	120,000	17,871	13,081	31,997	31,997	38,000	560,000
Imbalance	1.0	2.02	16.65	27.09	15.84	1.0	1.0

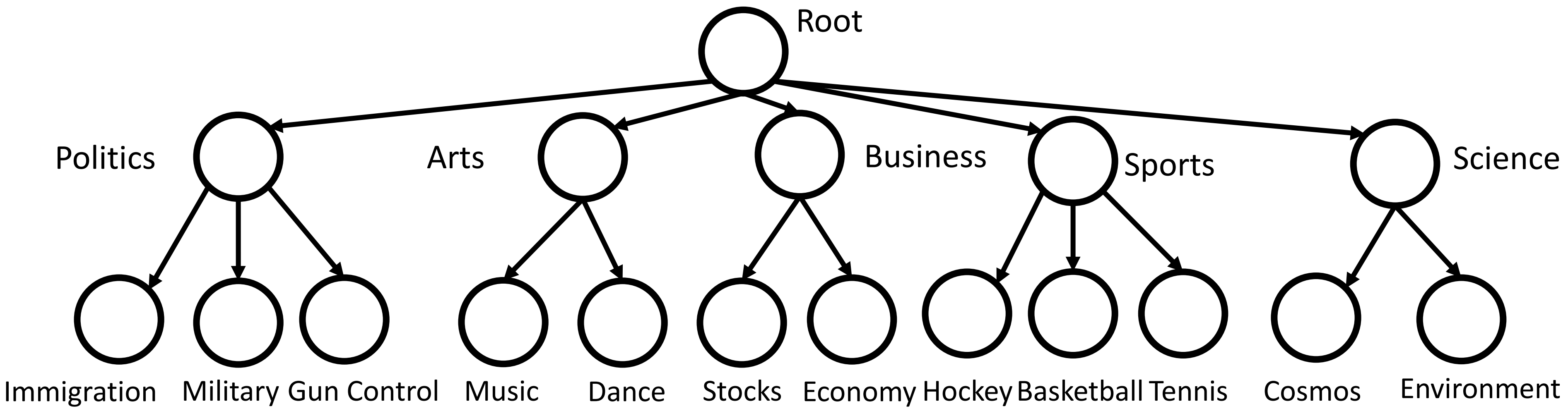
Model	AGNews	20News	NYT-Small	NYT-Topic	NYT-Location	Yelp	DBpedia
Supervised	93.99/93.99	96.45/96.42	97.95/95.46	94.29/89.90	95.99/94.99	95.7/95.7	98.96/98.96
WeSTClass	82.3/82.1	71.28/69.90	91.2/83.7	68.26/57.02	63.15/53.22	81.6/81.6	81.1/ N/A
ConWea	74.6/74.2	75.73/73.26	95.23/90.79	**81.67/71.54**	85.31/83.81	71.4/71.2	N/A
LOTClass	**86.89/86.82**	73.78/72.53	78.12/56.05	67.11/43.58	58.49/58.96	87.75/87.68	86.66/85.98
X-Class	84.8/84.65	**81.36/80.6**	**96.67/92.98**	80.6/69.92	**90.5/89.81**	**88.36/88.32**	**91.33/91.14**
X-Class-Rep	77.92/77.03	75.14/73.24	92.13/83.94	77.85/65.38	86.7/87.36	77.87/77.05	74.06/71.75
X-Class-Align	83.1/83.05	79.28/78.62	96.34/92.08	79.64/67.85	88.58/88.02	87.16/87.1	87.37/87.28

Outline

- What Weakly-Supervised Text Classification Is, and Why It Matters
- Flat Text Classification
  - Embedding: WeSTClass [CIKM'18]
  - Pre-trained LM: ConWea [ACL'20], LOTClass [EMNLP'20], X-Class [NAACL'21]
- Text Classification with Taxonomy Information
  - Embedding: WeSHClass [AAAI'19]
  - Pre-trained LM: TaxoClass [NAACL'21]
- Text Classification with Metadata Information
  - Embedding: MetaCat [SIGIR'20], HIMECat [WSDM'21]

WeSHClass: Weakly-Supervised Hierarchical Text Classification

- The hierarchy has a **tree** structure. Each document is associated with **one path** starting from the root node. (E.g., the main subject of each arXiv paper.)



- Keyword-level weak supervision: The name of each node in the taxonomy, or a few keywords for each leaf category

- Document-level weak supervision: A few labeled documents for each leaf category

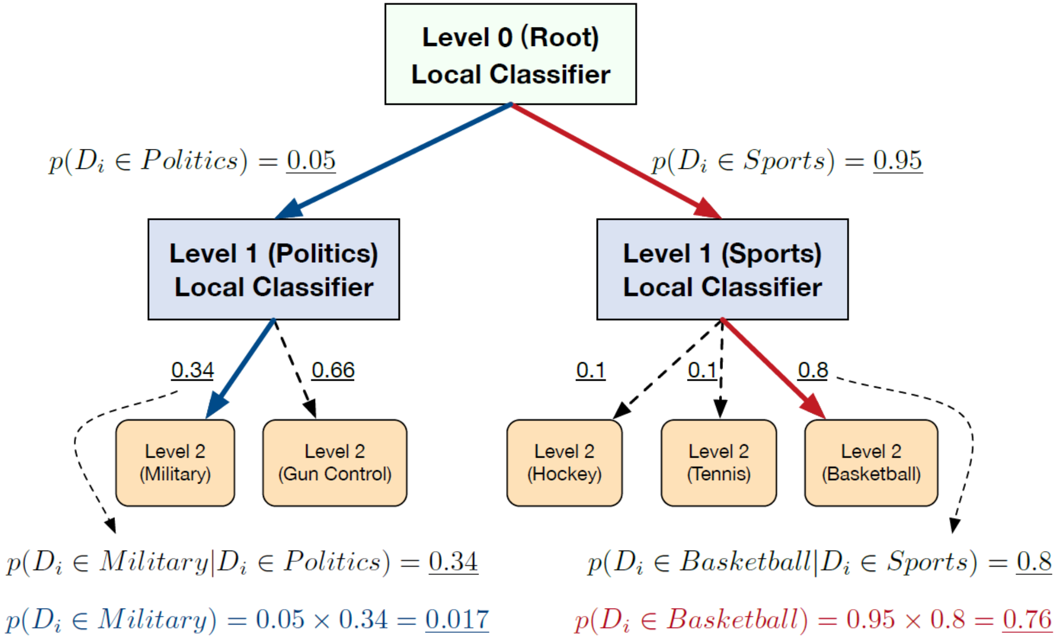
Meng, Y., Shen, J., Zhang, C., & Han, J. "Weakly-Supervised Hierarchical Text Classification", AAAI'19.
Applicable to both keyword-level and document-level supervision.

WeSHClass: Hierarchical Classification Model

- Local Classifier Pre-training
  - Generate β pseudo documents per class (recall WeSTClass) to pre-train the local classifier
  - A naive way of creating the label for a pseudo document D_i^*:
    - Directly use the associated class label it is generated from; one-hot encodings;
    - Problem: classifier overfitting to pseudo documents
  - Instead, use pseudo labels:
    - $l_{ij} = \begin{cases} (1-\alpha) + \alpha/m & D_i^* \text{ is generated from class } j \\ \alpha/m & \text{otherwise} \end{cases}.$
    - α accounts for the "noises" in pseudo documents; it is evenly split into all m classes
  - Pre-training is performed by minimizing KL divergence loss to pseudo labels

WeSHClass: Hierarchical Classification Model

- Global Classifier Per Level
  - At each level k in the class taxonomy, construct a global classifier by ensembling all local classifiers from root to level k
  - Use unlabeled documents to bootstrap the global classifier



WeSHClass: Hierarchical Classification Model

- Global Classifier Construction
  - The multiplication operation can be explained by the conditional probability formula:

$$p(D_i \in C_{child}) = p(D_i \in C_{child} \mid D_i \in C_{parent})p(D_i \in C_{parent})$$

  - All local classifiers from root to to level k are fine-tuned simultaneously via back-propagation during self-training; misclassifications at higher levels can be corrected
- Global Classifier Self-training
  - Step 1: Use the pre-trained global classifier to classify all unlabeled documents in the corpus;
  - Step 2: Compute pseudo labels based on current predictions:

$$l_{ij} = \frac{y_{ij}^2/f_j}{\sum_{j'} y_{ij'}^2/f_{j'}}$$ where $f_j = \sum_i y_{ij}$ and y_{ij} is the current prediction

  - Step 3: Minimize KL divergence loss to pseudo labels
  - Iterate between Steps 2 and 3 until less than $\delta\%$ of documents in the corpus have class assignment changes

WeSHClass: Experiment Results

- Datasets
  - New York Times; arXiv; Yelp Review
- Evaluation: Micro-F1 and Macro-F1 among all classes

Methods	NYT				arXiv				Yelp Review			
	KEYWORDS		DOCS		KEYWORDS		DOCS		KEYWORDS		DOCS	
	Macro	Micro	Macro Avg. (Std.)	Micro Avg. (Std.)	Macro	Micro	Macro Avg. (Std.)	Micro Avg. (Std.)	Macro	Micro	Macro Avg. (Std.)	Micro Avg. (Std.)
Hier-Dataless	0.593	0.811	-	-	0.374	0.594	-	-	0.284	0.312	-	-
Hier-SVM	-	-	0.142 (0.016)	0.469 (**0.012**)	-	-	0.049 (**0.001**)	0.443 (**0.006**)	-	-	0.220 (0.082)	0.310 (0.113)
CNN	-	-	0.165 (0.027)	0.329 (0.097)	-	-	0.124 (0.014)	0.456 (0.023)	-	-	0.306 (0.028)	0.372 (0.028)
WeSTClass	0.386	0.772	0.479 (0.027)	0.728 (0.036)	0.412	0.642	0.264 (0.016)	0.547 (0.009)	0.348	0.389	0.345 (0.027)	0.388 (0.033)
No-global	0.618	0.843	0.520 (0.065)	0.768 (0.100)	0.442	0.673	0.264 (0.020)	0.581 (0.017)	0.391	0.424	0.369 (0.022)	0.403 (0.016)
No-vMF	0.628	0.862	0.527 (0.031)	0.825 (0.032)	0.406	0.665	0.255 (0.015)	0.564 (0.012)	0.410	0.457	0.372 (0.029)	0.407 (0.015)
No-self-train	0.550	0.787	0.491 (0.036)	0.769 (0.039)	0.395	0.635	0.234 (0.013)	0.535 (0.010)	0.362	0.408	0.348 (0.030)	0.382 (0.022)
Our method	**0.632**	**0.874**	**0.532 (0.015)**	**0.827 (0.012)**	**0.452**	**0.692**	**0.279** (0.010)	**0.585** (0.009)	**0.423**	**0.461**	**0.375 (0.021)**	**0.410 (0.014)**

Outline

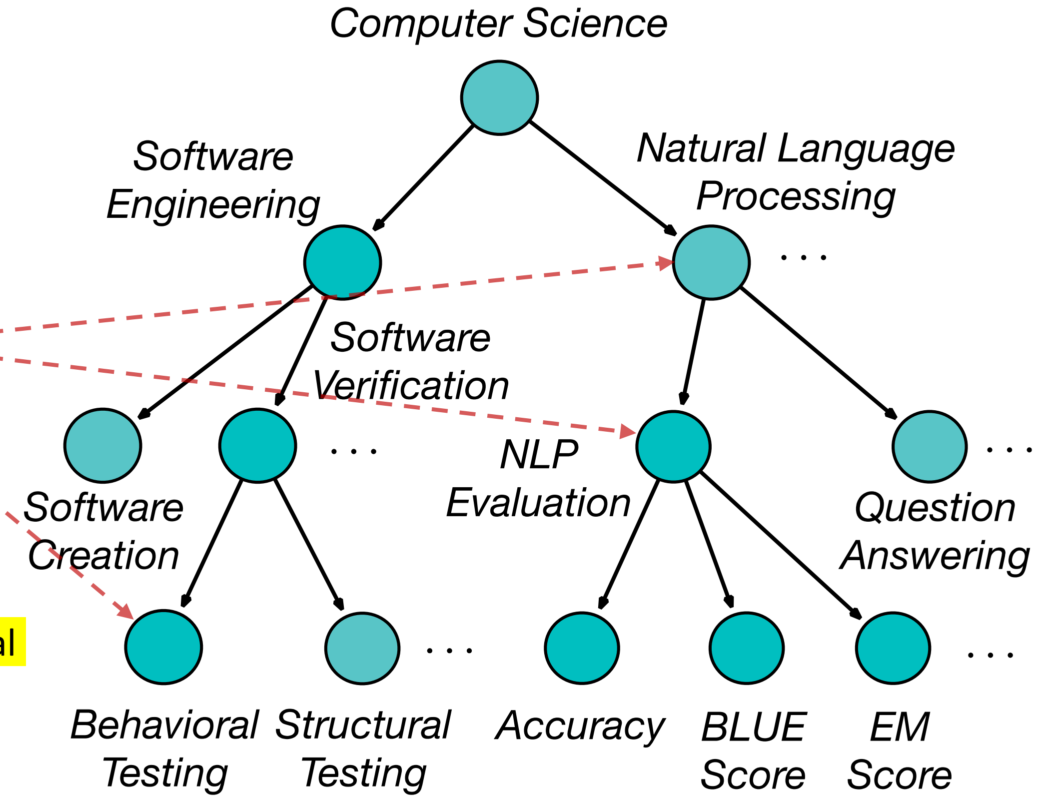
- What Weakly-Supervised Text Classification Is, and Why It Matters
- Flat Text Classification
  - Embedding: WeSTClass [CIKM’18]
  - Pre-trained LM: ConWea [ACL’20], LOTClass [EMNLP’20], X-Class [NAACL’21]
- Text Classification with Taxonomy Information
  - Embedding: WeSHClass [AAAI’19]
  - Pre-trained LM: TaxoClass [NAACL’21]
- Text Classification with Metadata Information
  - Embedding: MetaCat [SIGIR’20], HIMECat [WSDM’21]

TaxoClass: Weakly-supervised Hierarchical Multi-Label Text Classification

- The taxonomy is a directed acyclic graph (DAG)
- Each paper can have multiple categories distributed on different paths
- Category names can be phrases and may not appear in the corpus

Document

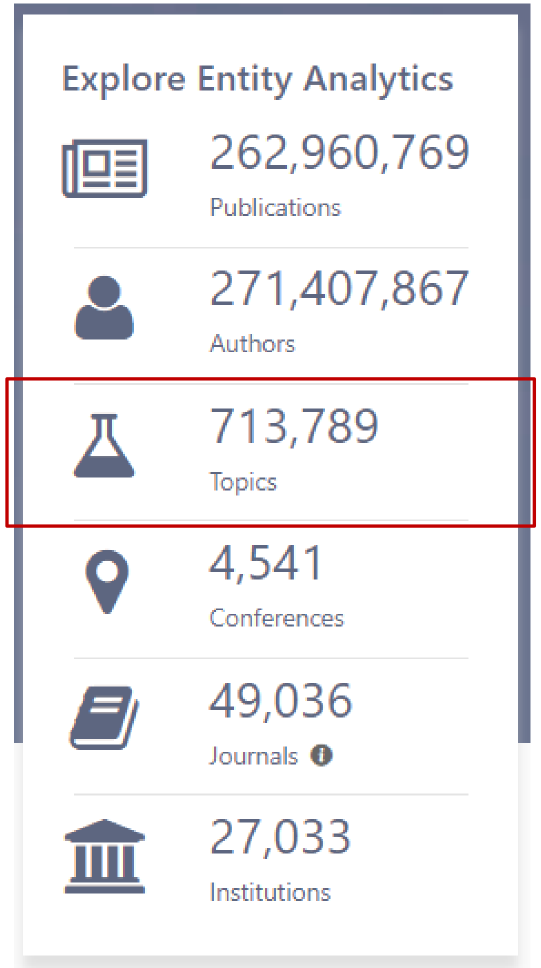
Measuring held-out accuracy often overestimates the performance of ***NLP*** models… Inspired by principles of ***behavioral testing*** in software engineering, we introduce CheckList, a task-agnostic methodology for ***testing NLP models***…



Shen, J., Qiu, W., Meng, Y., Shang, J., Ren, X., & Han, J., "TaxoClass: Hierarchical Multi-Label Text Classification Using Only Class Names", NAACL'21.
Category names as supervision.

TaxoClass: Why Category Names Only?

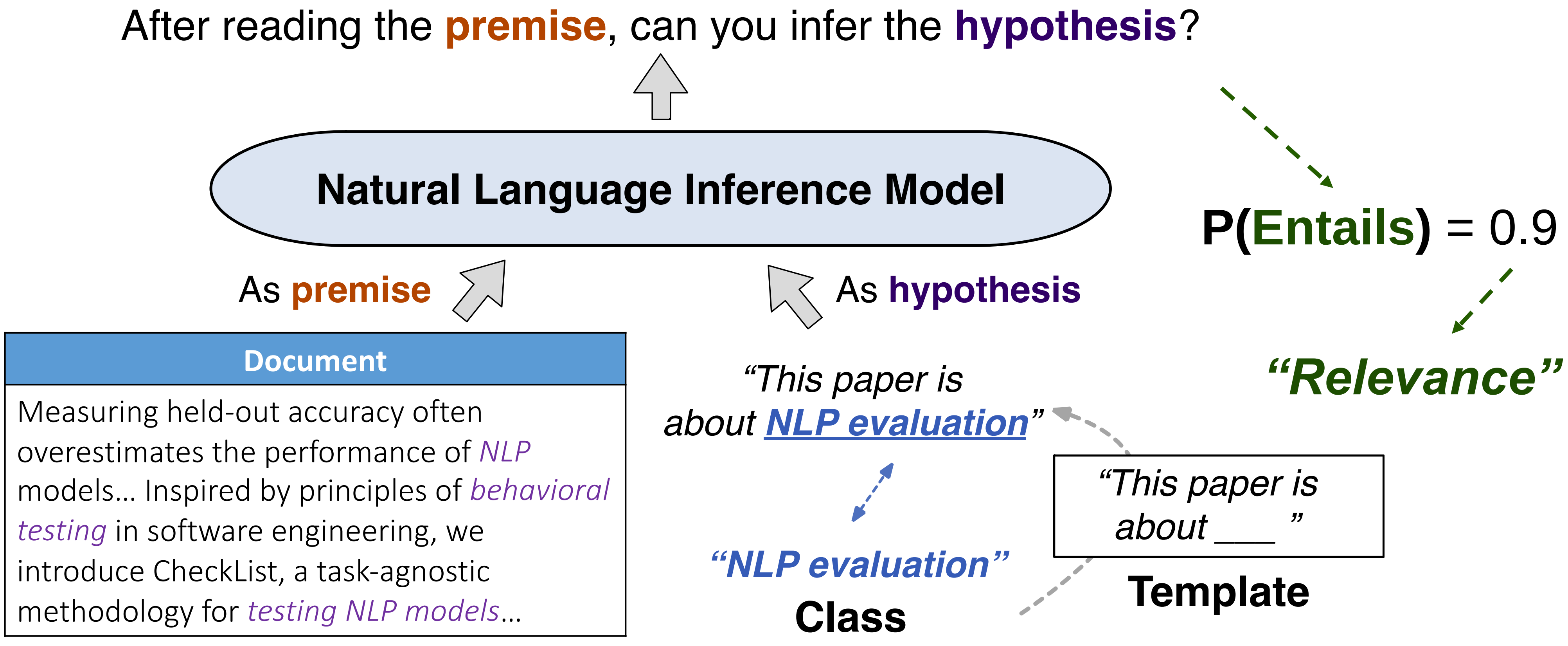
- Taxonomies for multi-label text classification are often big.
  - Amazon Product Catalog: $\times\ 10^4$ categories
  - MeSH Taxonomy (for medical papers): $\times\ 10^4$ categories
  - Microsoft Academic Taxonomy: $\times\ 10^5$ labels
- Impossible for users to provide even a small set of (e.g., 3) keywords/labeled documents for each category



https://academic.microsoft.com/home

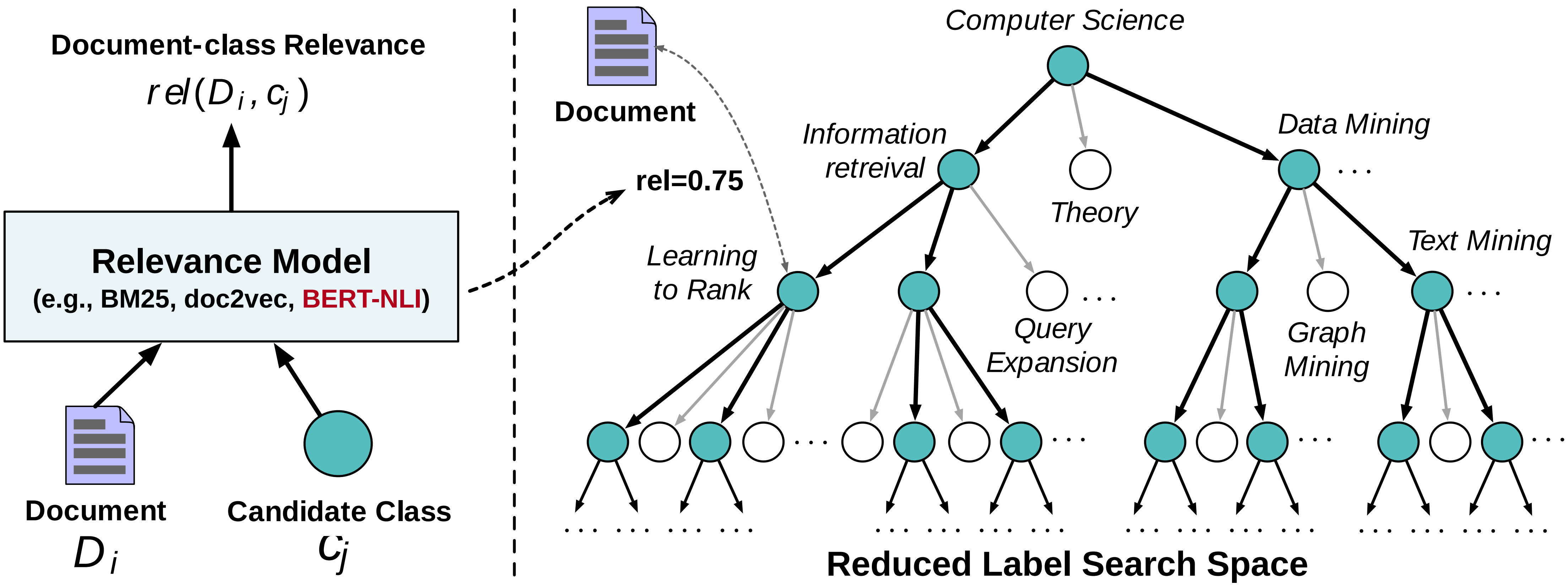
TaxoClass: Document-Class Relevance Calculation

- How to use the knowledge from pre-trained LMs?
- Relevance model: BERT/RoBERTa fine-tuned on the NLI task
  - https://huggingface.co/roberta-large-mnli



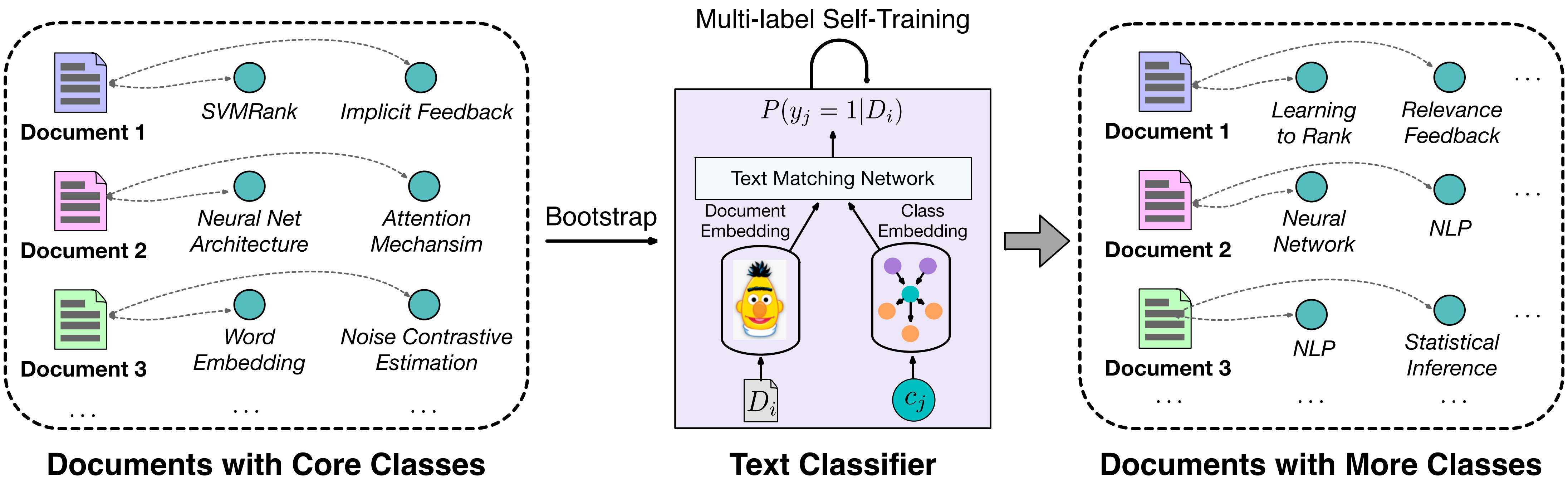
TaxoClass: Top-Down Exploration

- How to use the taxonomy?
- Shrink the label search space with top-down exploration
  - Use a relevance model to filter out completely irrelevant classes



TaxoClass: Identify Core Classes and More Classes

- Identify document core classes in reduced label search space
- Generalize from core classes with bootstrapping and self-training



TaxoClass: Experiment Results

	Methods	Amazon Example-F1	Amazon P@1	DBPedia Example-F1	DBPedia P@1
Weakly-supervised multi-class classification method	**WeSHClass** (Meng et al., AAAI'19)	0.246	0.577	0.305	0.536
Semi-supervised methods using 30% of training set	**SS-PCEM** (Xiao et al., WebConf'19)	0.292	0.537	0.385	0.742
	Semi-BERT (Devlin et al., NAACL'19)	0.339	0.592	0.428	0.761
Zero-shot method	**Hier-0Shot-TC** (Yin et al., EMNLP'19)	0.474	0.714	0.677	0.787
	TaxoClass (ours)	**0.593**	**0.812**	**0.816**	**0.894**

- **vs. WeSHClass**: better model document-class relevance
- **vs. SS-PCEM, Semi-BERT**: better leverage supervision signals from taxonomy
- **vs. Hier-0Shot-TC**: better capture domain-specific information from core classes

Amazon: 49K product reviews (29.5K training + 19.7K testing), 531 classes
DBPedia: 245K Wiki articles (196K training + 49K testing), 298 classes

Example-F1 $= \frac{1}{N}\sum_{i=1}^{N} \frac{2|true_i \cap pred_i|}{|true_i|+|pred_i|}$, **P@1** $= \frac{\#docs\ with\ top-1\ pred\ dorrect}{\#total\ docs}$

Outline

- What Weakly-Supervised Text Classification Is, and Why It Matters
- Flat Text Classification
  - Embedding: WeSTClass [CIKM’18]
  - Pre-trained LM: ConWea [ACL’20], LOTClass [EMNLP’20], X-Class [NAACL’21]
- Text Classification with Taxonomy Information
  - Embedding: WeSHClass [AAAI’19]
  - Pre-trained LM: TaxoClass [NAACL’21]
- Text Classification with Metadata Information
  - Embedding: MetaCat [SIGIR’20], HIMECat [WSDM’21]

MetaCat: Incorporating Metadata for Categorization

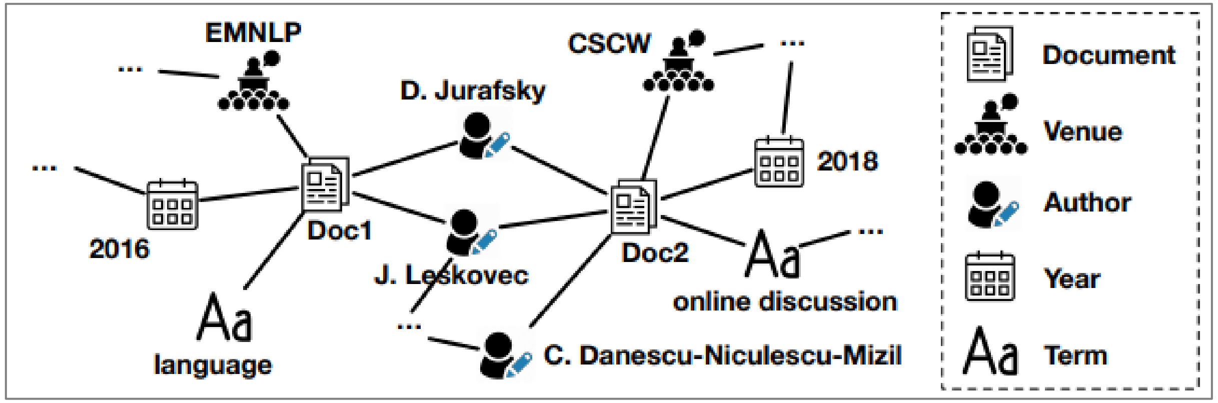
- Metadata is prevalent in many text sources
  - **GitHub repositories**: User, Tag
  - **Tweets**: User, Hashtag
  - **Amazon reviews**: User, Product
  - **Scientific papers**: Author, Venue
- How to leverage these heterogenous signals in the categorization process?

Zhang, Y., Meng, Y., Huang, J., Xu, F.F., Wang, X., & Han, J. "Minimally Supervised Categorization of Text with Metadata", SIGIR'20.
A few labeled documents as supervision.

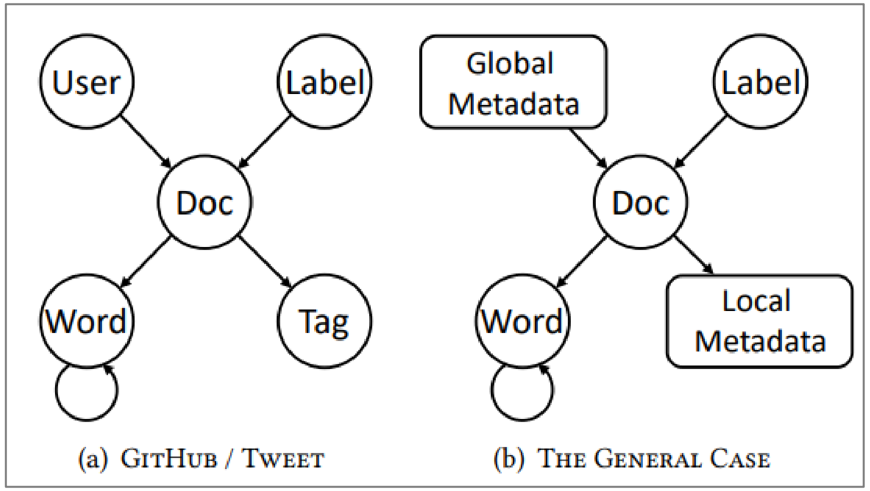
(a) GitHub Repository (b) Tweet (c) Amazon Review

MetaCat: The Underlying Generative Process

- Two categories of metadata:
  - **Global metadata**: user/author, product
    - "Causes" the generation of documents. (E.g., User/Author -> Document)
  - **Local metadata**: tag/hashtag
    - "Describes" the documents. (E.g., Document -> Tag)
  - We can also say "labels" are global, and "words" are local



A network view of corpus with metadata



A generative-process view of corpus with metadata

MetaCat: The Underlying Generative Process

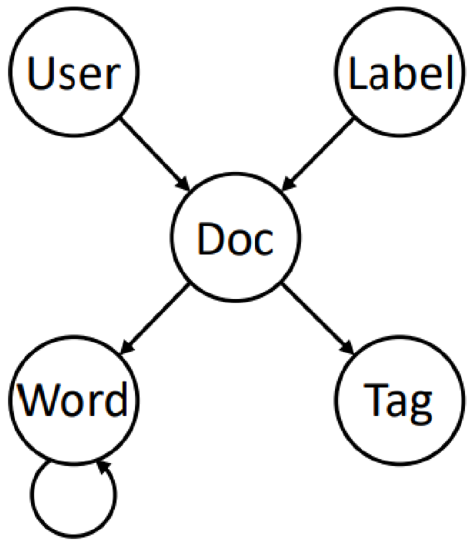
- We use GitHub/Tweet as a specific example to illustrate the process.
- Step 1: **User** (Global Metadata) & **Label** -> **Document**

  $p(d|u,l) \propto \exp(\boldsymbol{e}_d^T \boldsymbol{e}_u) \cdot \exp(\boldsymbol{e}_d^T \boldsymbol{e}_l)$

- Step 2: Document -> Word

  $p(w|d) \propto \exp(\boldsymbol{e}_w^T \boldsymbol{e}_d)$

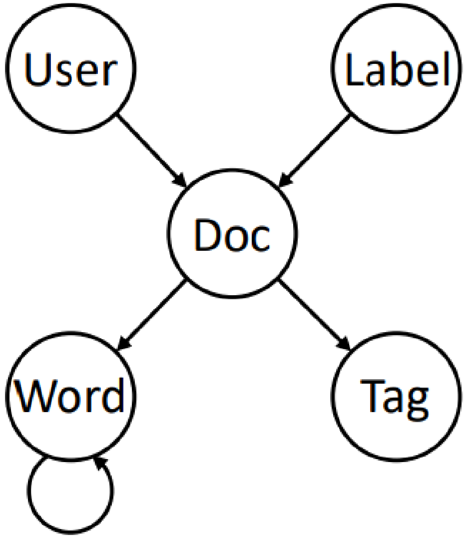
- Step 3: Document -> Tag (Local Metadata)
- Step 4: Word -> Context



(a) GitHub / Tweet

MetaCat: How to use this underlying model?

- **Embedding** Learning Module
  - All embedding vectors $\boldsymbol{e}_u, \boldsymbol{e}_l, \boldsymbol{e}_d, \boldsymbol{e}_t, \boldsymbol{e}_w$ are parameters of the generative process
  - Learn the embedding vectors through maximizing the likelihood of observing all text and metadata
- Training Data **Generation** Module
  - $\boldsymbol{e}_u, \boldsymbol{e}_l, \boldsymbol{e}_d, \boldsymbol{e}_t, \boldsymbol{e}_w$ have been learned
  - Given a label l, generate d, w and t according to the generative process



(a) GitHub / Tweet

MetaCat: Experiment Results

- Metadata is more helpful on smaller corpora.
- Datasets
  - GitHub-Bio: 10 categories; 876 docs
  - GitHub-AI: 14 categories; 1,596 docs
  - GitHub-Sec: 3 categories; 84,950 docs
  - Amazon: 10 categories; 100,000 docs
  - Twitter: 9 categories; 135,619 docs

Table 2: Micro F1 scores of compared algorithms on the five datasets. "–": excessive memory requirements.

Type	Method	GitHub-Bio	GitHub-AI	GitHub-Sec	Amazon	Twitter
Text-based	CNN [12]	0.2227 ± 0.0195	0.2404 ± 0.0404	0.4909 ± 0.0489	0.4915 ± 0.0374	0.3106 ± 0.0613
	HAN [38]	0.1409 ± 0.0145	0.1900 ± 0.0299	0.4677 ± 0.0334	0.4809 ± 0.0372	0.3163 ± 0.0878
	PTE [32]	0.3170 ± 0.0516	0.3511 ± 0.0403	0.4551 ± 0.0249	0.2997 ± 0.0786	0.1945 ± 0.0250
	WeSTClass [23]	0.3680 ± 0.0138	0.5036 ± 0.0287	0.6146 ± 0.0084	0.5312 ± 0.0161	0.3568 ± 0.0178
	PCEM [36]	0.3426 ± 0.0160	0.4820 ± 0.0292	0.5912 ± 0.0341	0.4645 ± 0.0163	0.2387 ± 0.0344
	BERT [4]	0.2680 ± 0.0303	0.2451 ± 0.0273	0.5538 ± 0.0368	0.5240 ± 0.0261	0.3312 ± 0.0860
Graph-based	ESim [27]	0.2925 ± 0.0223	0.4376 ± 0.0323	0.5480 ± 0.0109	0.5320 ± 0.0246	0.3512 ± 0.0226
	Metapath2vec [5]	0.3956 ± 0.0141	0.4444 ± 0.0231	0.5772 ± 0.0594	0.5256 ± 0.0335	0.3516 ± 0.0407
	HIN2vec [6]	0.2564 ± 0.0131	0.3614 ± 0.0234	0.5218 ± 0.0466	0.4987 ± 0.0252	0.2944 ± 0.0614
	TextGCN [39]	0.4759 ± 0.0126	0.6353 ± 0.0059	–	–	0.3361 ± 0.0032
	MetaCat	**0.5258 ± 0.0090**	**0.6889 ± 0.0128**	**0.7243 ± 0.0336**	**0.6422 ± 0.0058**	**0.3971 ± 0.0169**

Table 3: Macro F1 scores of compared algorithms on the five datasets. "–": excessive memory requirements.

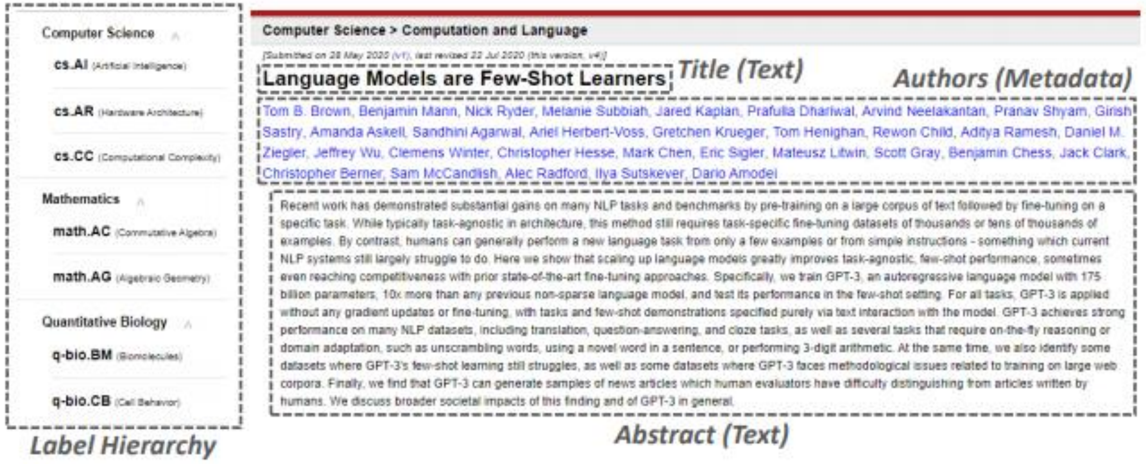
Type	Method	GitHub-Bio	GitHub-AI	GitHub-Sec	Amazon	Twitter
Text-based	CNN [12]	0.1896 ± 0.0133	0.1796 ± 0.0216	0.4268 ± 0.0584	0.5056 ± 0.0376	0.2858 ± 0.0559
	HAN [38]	0.0677 ± 0.0208	0.0961 ± 0.0254	0.4095 ± 0.0590	0.4644 ± 0.0597	0.2592 ± 0.0826
	PTE [32]	0.2630 ± 0.0371	0.3363 ± 0.0250	0.3803 ± 0.0218	0.2563 ± 0.0810	0.1739 ± 0.0190
	WeSTClass [23]	0.3414 ± 0.0129	0.4056 ± 0.0248	0.5497 ± 0.0054	0.5234 ± 0.0147	0.3085 ± 0.0398
	PCEM [36]	0.2977 ± 0.0281	0.3751 ± 0.0350	0.4033 ± 0.0336	0.4239 ± 0.0237	0.2039 ± 0.0472
	BERT [4]	0.1740 ± 0.0164	0.2083 ± 0.0415	0.4956 ± 0.0164	0.4911 ± 0.0544	0.2834 ± 0.0550
Graph-based	ESim [27]	0.2598 ± 0.0182	0.3209 ± 0.0202	0.4672 ± 0.0171	0.5336 ± 0.0220	0.3399 ± 0.0113
	Metapath2vec [5]	0.3214 ± 0.0128	0.3220 ± 0.0290	0.5140 ± 0.0637	0.5239 ± 0.0437	0.3443 ± 0.0208
	HIN2vec [6]	0.2742 ± 0.0136	0.2513 ± 0.0211	0.4000 ± 0.0115	0.4261 ± 0.0284	0.2411 ± 0.0142
	TextGCN [39]	0.4817 ± 0.0078	0.5997 ± 0.0013	–	–	0.3191 ± 0.0029
	MetaCat	**0.5230 ± 0.0080**	**0.6154 ± 0.0079**	**0.6323 ± 0.0235**	**0.6496 ± 0.0091**	**0.3612 ± 0.0067**

HIMECat: Jointly Modeling Metadata and Hierarchy

- How to jointly leverage the label hierarchy, metadata, and text information?

Zhang, Y., Chen, X., Meng, Y., & Han, J. “Hierarchical Metadata-Aware Document Categorization under Weak Supervision”, WSDM’21.
A few labeled documents (for each leaf category) as supervision.

(a) **GitHub Repository.** Label Hierarchy: PaperWithCode Task Taxonomy (https://paperswithcode.com/sota); Text: Description and README; Metadata: User and Tag.



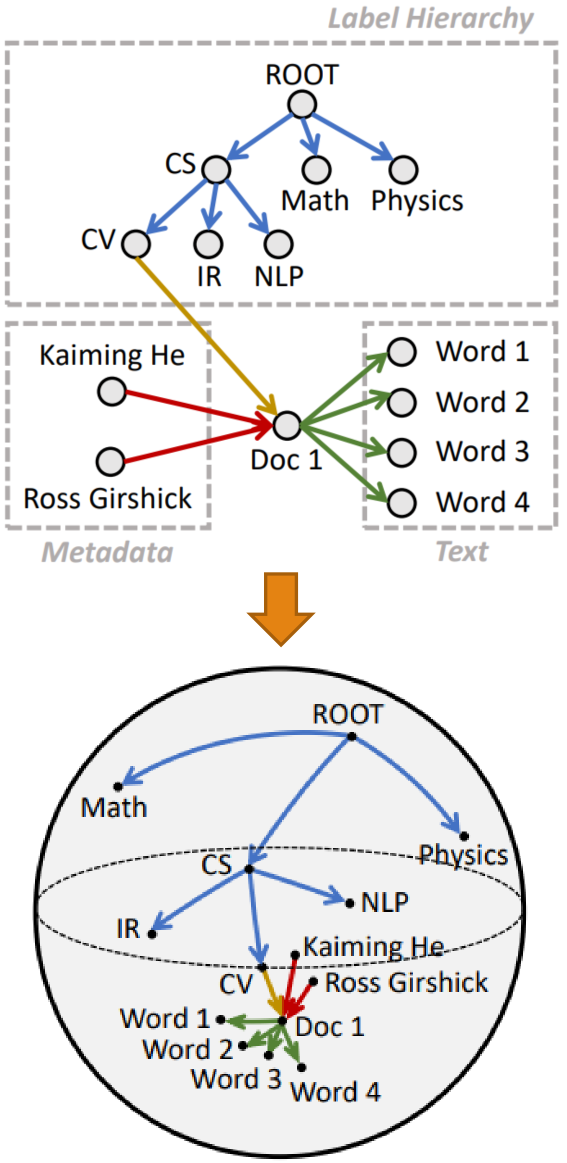
(b) **arXiv Paper.** Label Hierarchy: arXiv Category Taxonomy (https://arxiv.org/category_taxonomy); Text: Title and Abstract; Metadata: Author.

(c) **Amazon Review.** Label Hierarchy: Amazon Product Catalog [24]; Text: Title and Review; Metadata: User and Product.

HIMECat: A Hierarchical Generative Process

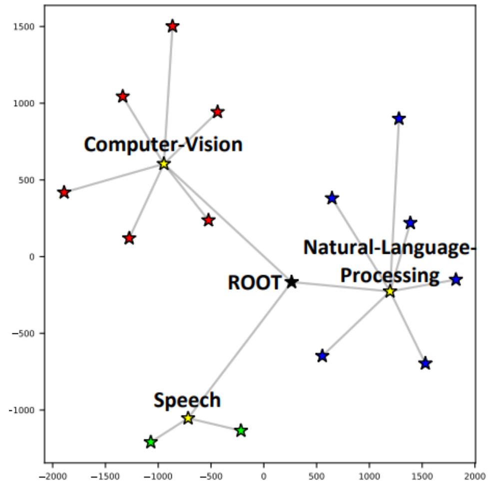
- Step 1: Parent Label -> Child Label
- Step 2: **Leaf** label & Metadata -> Document
- Step 3: Document -> Word

- **Joint Representation Learning**
  - Embeddings are the parameters of the generative process.
  - Maximum likelihood estimation of the parameters when observing the hierarchy, metadata and text
- **Hierarchical Data Augmentation**
  - After representation learning, how to synthesize training data for each class?
  - Follow the generative process

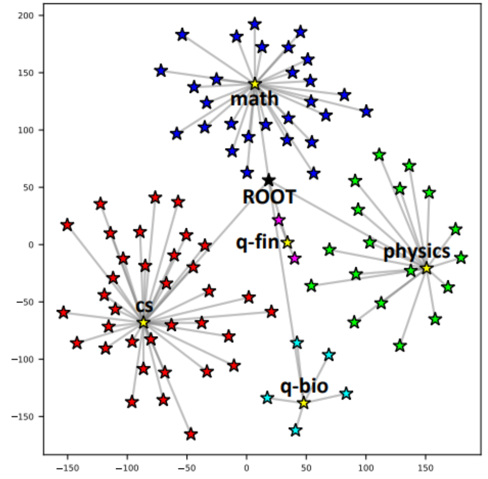


HIMECat: Experimental Results

- Datasets
  - GitHub: 3+14 categories; 1,596 docs
  - ArXiv: 5+88 categories; 25,960 docs
  - Amazon: 18+147 categories; 147,000 docs
- Metrics
  - F1 scores on leaf categories
  - F1 scores on all non-root categories



(a) GitHub



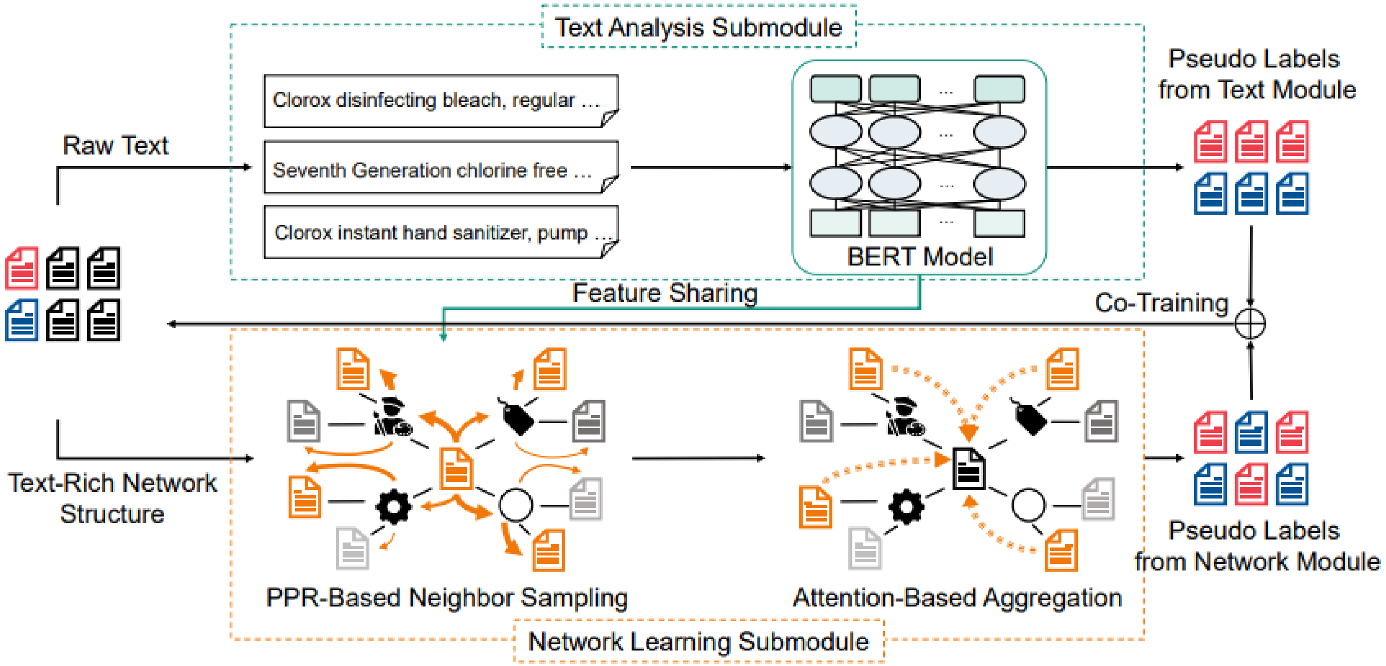
(b) ArXiv

Table 2: {Leaf, Overall}×{Micro, Macro} F1 scores of compared algorithms on the three datasets. *: significantly worse than HiMeCat (p-value < 0.05). **: significantly worse than HiMeCat (p-value < 0.01).

	GitHub				ArXiv				Amazon			
	Leaf		Overall		Leaf		Overall		Leaf		Overall	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
HierSVM [7]	0.1861**	0.1388**	0.4862**	0.2457**	0.0538**	0.0460**	0.4066**	0.0750**	0.0248**	0.0217**	0.2218**	0.0494**
WeSHClass [29]	0.1727**	0.1559**	0.3332**	0.1924**	0.0604**	0.0602**	0.3077**	0.0797**	0.0483**	0.0500**	0.1234**	0.0640**
PCEM [48]	0.2519**	0.1234**	0.5299*	0.1786**	0.1090**	0.0717**	0.4440	0.0963**	0.0675**	0.0439**	0.2189**	0.0659**
HiGitClass [53]	0.3984	0.3902*	0.5073**	0.4084**	0.1738**	0.1656**	0.3928**	0.1880**	0.0903**	0.0876**	0.1677**	0.1040**
MetaCat [51]	0.3762**	0.3403**	0.5411*	0.3863**	0.0790**	0.0768**	0.3071**	0.0935**	0.1008**	0.0994**	0.1703**	0.1083**
Metapath2vec [6]	0.2814**	0.2805**	0.4592**	0.3212**	0.1360**	0.1344**	0.3419**	0.1534**	0.0669**	0.0666**	0.1334**	0.0800**
Poincaré [32]	0.2750**	0.1980**	0.4302**	0.2218**	0.1336**	0.1296**	0.2995**	0.1454**	0.0645**	0.0607**	0.1202**	0.0739**
BERT [5]	0.2889**	0.2561**	0.4675**	0.3007**	0.1316**	0.0812**	0.4203**	0.1100**	0.0891**	0.0520**	0.2361**	0.0771**
HiMeCat	**0.4254**	**0.4209**	**0.5820**	**0.4535**	**0.2038**	**0.1938**	**0.4509**	**0.2191**	**0.1552**	**0.1553**	**0.2748**	**0.1770**

Using Pretrained LMs for Metadata-Aware Text Classification

- Difficulty: How to let pretrained LMs encode metadata/network information
  - The limited vocabulary can hardly identify some types of metadata (e.g., author names, product IDs)
- Related study:
  - Co-training GNN and BERT to deal with metadata and text, respectively



Zhang, X., Zhang, C., Dong, X. L., Shang, J., & Han, J. "Minimally-Supervised Structure-Rich Text Categorization via Learning on Text-Rich Networks", WWW'21.

Summary

Method	Flat vs. Hierarchical	Single-label vs. Multi-label	Supervision Format	Embedding vs. Pretrained LM
WeSTClass	Flat	Single-label	All 3 types	Embedding
ConWea	Flat	Single-label	Keywords	Pretrained LM
LOTClass	Flat	Single-label	Category Names	Pretrained LM
X-Class	Flat & Hierarchical	Single-label & Path	Category Names	Pretrained LM
WeSHClass	Hierarchical	Path	All 3 types	Embedding
TaxoClass	Hierarchical	Multi-label	Category Names	Pretrained LM
MetaCat	Flat	Single-label	A Few Labeled Docs	Embedding
HIMECat	Hierarchical	Path	A Few Labeled Docs	Embedding
LTRN	Flat	Single-label	A Few Labeled Docs	Pretrained LM

References

- Meng, Y., Shen, J., Zhang, C., & Han, J. “Weakly-supervised neural text classification”, CIKM’18
- Mekala, D. & Shang, J. “Contextualized Weak Supervision for Text Classification”, ACL’20
- Meng, Y., Zhang, Y., Huang, J., Xiong, C., Ji, H., Zhang, C., & Han, J. “Text Classification Using Label Names Only: A Language Model Self-Training Approach”, EMNLP’20
- Wang, Z., Mekala, D., & Shang, J. “X-Class: Text Classification with Extremely Weak Supervision”, NAACL’21
- Meng, Y., Shen, J., Zhang, C., & Han, J. “Weakly-Supervised Hierarchical Text Classification”, AAAI’19
- Shen, J., Qiu, W., Meng, Y., Shang, J., Ren, X., & Han, J., “TaxoClass: Hierarchical Multi-Label Text Classification Using Only Class Names”, NAACL’21
- Zhang, Y., Meng, Y., Huang, J., Xu, F.F., Wang, X., & Han, J. “Minimally Supervised Categorization of Text with Metadata”, SIGIR’20
- Zhang, Y., Chen, X., Meng, Y., & Han, J. “Hierarchical Metadata-Aware Document Categorization under Weak Supervision”, WSDM’21
- Zhang, X., Zhang, C., Dong, X. L., Shang, J., & Han, J. "Minimally-Supervised Structure-Rich Text Categorization via Learning on Text-Rich Networks", WWW'21

Q&A