Weakly Supervised Text Classification

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
October 4, 2022
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]
    - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
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
- 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)

[https://academic.microsoft.com/paper/2963341956/](https://academic.microsoft.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/](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 89.5% (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 63.1 (5.1 point absolute improvement).

Subjects: Computation and Language (cs.CL)
Cite as: arXiv:1810.04805 [cs.CL]
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

- **Weakly supervised text classification**
  - While users cannot afford to label sufficient documents for training a deep neural classifier, they can provide *a small amount of seed information*:
    - Category names or 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**: category names or a few relevant keywords
- **Document-level weak supervision**: a small set of labeled docs

![Diagram showing sources of supervision:]

- **Source 1: Label surface names**
  - Politics
  - Sports
  - Technology

- **Source 2: Class-related keywords**
  - democracy, republic, religion, liberal
  - basketball, football, tennis, athletes
  - software, computer, drone, telescope

- **Source 3: Labeled documents**
  - politics news
  - sports news
  - technology news
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]
  - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
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

Fit a von-Mises 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

Mathematical formula:

\[ p(x|\mu, \kappa) = C_D(\kappa) \exp(\kappa \mu^T x) \]

\[ C_D(\kappa) = \frac{\kappa^{D/2-1}}{I_{D/2-1}(\kappa)} \]
Datasets: (1) NYT, (2) AG’s News, (3) Yelp

Evaluation: use different types of weak supervision and measure accuracies

Macro-F1 scores: (i.e., unweighted mean of the F1 scores calculated per class)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LABELS</td>
<td>KEYWORDS</td>
<td>DOCS</td>
</tr>
<tr>
<td>IR with tf-idf</td>
<td>0.319</td>
<td>0.509</td>
<td>-</td>
</tr>
<tr>
<td>Topic Model</td>
<td>0.301</td>
<td>0.253</td>
<td>-</td>
</tr>
<tr>
<td>Dataless</td>
<td>0.484</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UNEC</td>
<td>0.690</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PTE</td>
<td>-</td>
<td>-</td>
<td>0.834 (0.024)</td>
</tr>
<tr>
<td>HAN</td>
<td>0.348</td>
<td>0.534</td>
<td>0.740 (0.059)</td>
</tr>
<tr>
<td>CNN</td>
<td>0.338</td>
<td>0.632</td>
<td>0.702 (0.059)</td>
</tr>
<tr>
<td>NoST-HAN</td>
<td>0.515</td>
<td>0.213</td>
<td>0.823 (0.035)</td>
</tr>
<tr>
<td>NoST-CNN</td>
<td>0.701</td>
<td>0.702</td>
<td>0.833 (0.013)</td>
</tr>
<tr>
<td>WeSTCLASS-HAN</td>
<td>0.754</td>
<td>0.640</td>
<td>0.832 (0.028)</td>
</tr>
<tr>
<td>WeSTCLASS-CNN</td>
<td>0.830</td>
<td>0.837</td>
<td>0.835 (0.010)</td>
</tr>
</tbody>
</table>

Micro-F1 scores: (i.e., calculated using the total # of True Positives (TP), False Positives (FP) and False Negatives (FN), instead of individually for each class)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LABELS</td>
<td>KEYWORDS</td>
<td>DOCS</td>
</tr>
<tr>
<td>IR with tf-idf</td>
<td>0.240</td>
<td>0.346</td>
<td>-</td>
</tr>
<tr>
<td>Topic Model</td>
<td>0.666</td>
<td>0.623</td>
<td>-</td>
</tr>
<tr>
<td>Dataless</td>
<td>0.710</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UNEC</td>
<td>0.810</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PTE</td>
<td>-</td>
<td>-</td>
<td>0.906 (0.020)</td>
</tr>
<tr>
<td>HAN</td>
<td>0.251</td>
<td>0.595</td>
<td>0.849 (0.038)</td>
</tr>
<tr>
<td>CNN</td>
<td>0.246</td>
<td>0.620</td>
<td>0.798 (0.085)</td>
</tr>
<tr>
<td>NoST-HAN</td>
<td>0.788</td>
<td>0.676</td>
<td>0.906 (0.021)</td>
</tr>
<tr>
<td>NoST-CNN</td>
<td>0.767</td>
<td>0.780</td>
<td>0.908 (0.013)</td>
</tr>
<tr>
<td>WeSTCLASS-HAN</td>
<td>0.901</td>
<td>0.859</td>
<td>0.908 (0.019)</td>
</tr>
<tr>
<td>WeSTCLASS-CNN</td>
<td>0.916</td>
<td>0.912</td>
<td>0.911 (0.007)</td>
</tr>
</tbody>
</table>
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]
  - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
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:

<table>
<thead>
<tr>
<th>Class</th>
<th>Seed words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>soccer, goal, penalty</td>
</tr>
<tr>
<td>Law</td>
<td>law, judge, court</td>
</tr>
</tbody>
</table>

- 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, slides credited 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”

User-Provided Seed Words

<table>
<thead>
<tr>
<th>Class</th>
<th>Seed Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>soccer, goal, penalty</td>
</tr>
<tr>
<td>Law</td>
<td>law, judge, court</td>
</tr>
</tbody>
</table>

Extended Seed Words

<table>
<thead>
<tr>
<th>Class</th>
<th>Seed Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>soccer, goal$0$, goal$1$, penalty$0$, penalty$1$,</td>
</tr>
<tr>
<td>Law</td>
<td>law, judge, court$0$, court$1$</td>
</tr>
</tbody>
</table>

Contextualized & Expanded Seed Words

<table>
<thead>
<tr>
<th>Class</th>
<th>Seed Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>soccer, goal$0$, penalty$1$,</td>
</tr>
<tr>
<td>Law</td>
<td>law, judge, court$1$, penalty$0$,</td>
</tr>
</tbody>
</table>

Comparative Ranking

Law ↔ Soccer
Cosmos ↔ Politics

Raw Docs

Messi scored the penalty! …
Judge passed the order of …
The court issued a penalty …
……

Contextualized Docs

Messi scored the penalty$1$! …
Judge passed the order of …
The court$1$ issued a penalty$0$ …
……

Text Classifier

Contextualized Docs with Predictions

Messi scored the penalty$1$!
Judge passed the order of …
The court$1$ issued a penalty$0$ …
……
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

<table>
<thead>
<tr>
<th>Methods</th>
<th>NYT 5-Class (Coarse) Micro-F₁</th>
<th>NYT 5-Class (Fine) Macro-F₁</th>
<th>NYT 6-Class (Coarse) Micro-F₁</th>
<th>NYT 6-Class (Fine) Macro-F₁</th>
<th>NYT 20-Class (Fine) Micro-F₁</th>
<th>NYT 20-Class (Fine) Macro-F₁</th>
<th>20 Newsgroup 5-Class (Coarse) Micro-F₁</th>
<th>20 Newsgroup 5-Class (Fine) Macro-F₁</th>
<th>20 Newsgroup 6-Class (Coarse) Micro-F₁</th>
<th>20 Newsgroup 6-Class (Fine) Macro-F₁</th>
<th>20 Newsgroup 20-Class (Fine) Micro-F₁</th>
<th>20 Newsgroup 20-Class (Fine) Macro-F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR-TF-IDF</td>
<td>0.65</td>
<td>0.58</td>
<td>0.56</td>
<td>0.54</td>
<td>0.49</td>
<td>0.48</td>
<td>0.53</td>
<td>0.52</td>
<td>0.53</td>
<td>0.52</td>
<td>0.53</td>
<td>0.52</td>
</tr>
<tr>
<td>Dataless</td>
<td>0.71</td>
<td>0.48</td>
<td>0.59</td>
<td>0.37</td>
<td>0.50</td>
<td>0.47</td>
<td>0.61</td>
<td>0.53</td>
<td>0.61</td>
<td>0.53</td>
<td>0.61</td>
<td>0.53</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>0.92</td>
<td>0.83</td>
<td>0.69</td>
<td>0.47</td>
<td>0.51</td>
<td>0.45</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Doc2Cube</td>
<td>0.71</td>
<td>0.38</td>
<td>0.67</td>
<td>0.34</td>
<td>0.40</td>
<td>0.35</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
<td>0.23</td>
</tr>
<tr>
<td>WeSTClass</td>
<td>0.91</td>
<td>0.84</td>
<td>0.50</td>
<td>0.36</td>
<td>0.53</td>
<td>0.43</td>
<td>0.49</td>
<td>0.46</td>
<td>0.49</td>
<td>0.46</td>
<td>0.49</td>
<td>0.46</td>
</tr>
<tr>
<td>ConWea</td>
<td><strong>0.95</strong></td>
<td><strong>0.89</strong></td>
<td><strong>0.91</strong></td>
<td><strong>0.79</strong></td>
<td><strong>0.62</strong></td>
<td><strong>0.57</strong></td>
<td><strong>0.65</strong></td>
<td><strong>0.64</strong></td>
<td><strong>0.65</strong></td>
<td><strong>0.64</strong></td>
<td><strong>0.65</strong></td>
<td><strong>0.64</strong></td>
</tr>
<tr>
<td>ConWea-NoCon</td>
<td>0.91</td>
<td>0.83</td>
<td>0.89</td>
<td>0.74</td>
<td>0.53</td>
<td>0.50</td>
<td>0.58</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
</tr>
<tr>
<td>ConWea-NoExpan</td>
<td>0.92</td>
<td>0.85</td>
<td>0.76</td>
<td>0.66</td>
<td>0.58</td>
<td>0.53</td>
<td>0.58</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
</tr>
<tr>
<td>ConWea-WSD</td>
<td>0.83</td>
<td>0.78</td>
<td>0.72</td>
<td>0.64</td>
<td>0.52</td>
<td>0.46</td>
<td>0.49</td>
<td>0.47</td>
<td>0.49</td>
<td>0.47</td>
<td>0.49</td>
<td>0.47</td>
</tr>
<tr>
<td>HAN-Supervised</td>
<td>0.96</td>
<td>0.92</td>
<td>0.94</td>
<td>0.82</td>
<td>0.90</td>
<td>0.88</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
</tbody>
</table>
LOTClass: Label-Name-Only Text Classification

- **Inputs:** A set of label names representing each class + unlabeled documents
- **Method:** Make good use of pre-trained language model (e.g., BERT)
  - **Step 1.** Category understanding via label name replacement (learn *topic vocabulary*)
  - Ex. “sports” → {“soccer”, “basketball”, …} (use pretrained LM to replace category name)

<table>
<thead>
<tr>
<th>Label Name</th>
<th>Category Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>politics</td>
<td>political, politician, government, elections, politician, democracy, democratic, governing, party, leadership, state, election, politically, affairs, issues, governments, voters, debate, cabinet, congress, democrat, president, religion, …</td>
</tr>
<tr>
<td>sports</td>
<td>sports, games, sporting, game, athletics, national, athletic, espn, soccer, basketball, stadium, arts, racing, baseball, tv, hockey, pro, press, team, red, home, bay, kings, city, legends, winning, miracle, olympic, ball, giants, players, champions, boxing, …</td>
</tr>
<tr>
<td>business</td>
<td>business, trade, commercial, enterprise, shop, money, market, commerce, corporate, global, future, sales, general, international, group, retail, management, companies, operations, operation, store, corporation, venture, economic, division, firm, …</td>
</tr>
<tr>
<td>technology</td>
<td>technology, tech, software, technological, device, equipment, hardware, devices, infrastructure, system, knowledge, technique, digital, technical, concept, systems, gear, techniques, functionality, process, material, facility, feature, method, …</td>
</tr>
</tbody>
</table>

- Find topic words based on label names: Overcome the low semantic coverage of label names
- Make good use of pretrained LM (e.g., BERT)
- Result from AGNews dataset

LOTClass: Meaning of Word Is Context-Dependent

- Use language models to predict what words can replace the label names
  - Interchangeable words are likely to have similar meanings
  - Note: The same word with different contents may have very different meanings

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

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.
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

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

![Image](image)

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

Figure 1: Visualizations of News using Average BERT Representations. Colors denote different classes.
X-Class: Class-Oriented BERT Representations

- Clustering for classification based on class-oriented representations

<table>
<thead>
<tr>
<th>Raw Input Corpus</th>
<th>User-Specified Class Names</th>
<th>Class-Oriented Representation</th>
<th>Document-Class Alignment (confidence estimated)</th>
<th>Text Classifier Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID Documents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_1$ I cheered for Lakers winning NBA.</td>
<td>Sentiment: happy, sad</td>
<td>Happy $D_1^*$, $D_3$</td>
<td>Happy sad</td>
<td></td>
</tr>
<tr>
<td>$D_2$ I am sad that Heat lost.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_3$ Great news! Scientists discovered …</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_4$ The new film is not satisfactory.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## X-Class: Experiment Results

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>AGNews</th>
<th>20News</th>
<th>NYT-Small</th>
<th>NYT-Topic</th>
<th>NYT-Location</th>
<th>Yelp</th>
<th>DBpedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>93.99/93.99</td>
<td>96.45/96.42</td>
<td>97.95/95.46</td>
<td>94.29/89.90</td>
<td>95.99/94.99</td>
<td>95.7/95.7</td>
<td>98.96/98.96</td>
</tr>
<tr>
<td>WeSTClass</td>
<td>82.3/82.1</td>
<td>71.28/69.90</td>
<td>91.2/83.7</td>
<td>68.26/57.02</td>
<td>63.15/53.22</td>
<td>81.6/81.6</td>
<td>81.1/ N/A</td>
</tr>
<tr>
<td>ConWea</td>
<td>74.6/74.2</td>
<td>75.73/73.26</td>
<td>95.23/90.79</td>
<td><strong>81.67/71.54</strong></td>
<td>85.31/83.81</td>
<td>71.4/71.2</td>
<td>N/A</td>
</tr>
<tr>
<td>LOTClass</td>
<td><strong>86.89/86.82</strong></td>
<td>73.78/72.53</td>
<td>78.12/56.05</td>
<td>67.11/43.58</td>
<td>58.49/58.96</td>
<td>87.75/87.68</td>
<td>86.66/85.98</td>
</tr>
<tr>
<td>X-Class</td>
<td>84.8/84.65</td>
<td><strong>81.36/80.6</strong></td>
<td><strong>96.67/92.98</strong></td>
<td>80.6/69.92</td>
<td><strong>90.5/89.81</strong></td>
<td><strong>88.36/88.32</strong></td>
<td><strong>91.33/91.14</strong></td>
</tr>
<tr>
<td>X-Class-Rep</td>
<td>77.92/77.03</td>
<td>75.14/73.24</td>
<td>92.13/83.94</td>
<td>77.85/65.38</td>
<td>86.7/87.36</td>
<td>77.87/77.05</td>
<td>74.06/71.75</td>
</tr>
<tr>
<td>X-Class-Align</td>
<td>83.1/83.05</td>
<td>79.28/78.62</td>
<td>96.34/92.08</td>
<td>79.64/67.85</td>
<td>88.58/88.02</td>
<td>87.16/87.1</td>
<td>87.37/87.28</td>
</tr>
</tbody>
</table>
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]
    - Pre-trained LM: MICoL [WWW’22]
  - Text Classification: Innovative Applications
    - Aspect-based Sentiment Analysis [EMNLP’20]
    - Mining Text Outlier in Document Directories [ICDM’20]
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

WeSHClass: Hierarchical Classification Model

- Local Classifier Per Node
- Essentially a flat classification task
- Follow WeSTClass
- 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: Experiment Results

- **Datasets**
  - New York Times; arXiv; Yelp Review

- **Evaluation**: Micro-F1 and Macro-F1 among all classes

<table>
<thead>
<tr>
<th>Methods</th>
<th>NYT KEYWORDS</th>
<th>NYT DOCS</th>
<th>arXiv KEYWORDS</th>
<th>arXiv DOCS</th>
<th>Yelp Review</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Macro</td>
<td>Micro</td>
<td>Macro Avg. (Std.)</td>
<td>Micro Avg. (Std.)</td>
<td>Macro</td>
</tr>
<tr>
<td>Hier-Dataless</td>
<td>0.593</td>
<td>0.811</td>
<td>-</td>
<td>-</td>
<td>0.374</td>
</tr>
<tr>
<td>Hier-SVM</td>
<td>-</td>
<td>-</td>
<td>0.142 (0.016)</td>
<td>0.469 (0.012)</td>
<td>0.049 (0.001)</td>
</tr>
<tr>
<td>CNN</td>
<td>-</td>
<td>-</td>
<td>0.165 (0.027)</td>
<td>0.329 (0.097)</td>
<td>0.124 (0.014)</td>
</tr>
<tr>
<td>WeSTClass</td>
<td>0.386</td>
<td>0.772</td>
<td>0.479 (0.027)</td>
<td>0.728 (0.036)</td>
<td>0.412</td>
</tr>
<tr>
<td>No-global</td>
<td>0.618</td>
<td>0.843</td>
<td>0.520 (0.065)</td>
<td>0.768 (0.100)</td>
<td>0.442</td>
</tr>
<tr>
<td>No-vMF</td>
<td>0.628</td>
<td>0.862</td>
<td>0.527 (0.031)</td>
<td>0.825 (0.032)</td>
<td>0.406</td>
</tr>
<tr>
<td>No-self-train</td>
<td>0.550</td>
<td>0.787</td>
<td>0.491 (0.036)</td>
<td>0.769 (0.089)</td>
<td>0.395</td>
</tr>
<tr>
<td>Our method</td>
<td>0.632</td>
<td>0.874</td>
<td>0.532 (0.015)</td>
<td>0.827 (0.012)</td>
<td>0.452</td>
</tr>
</tbody>
</table>
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]
    - Pre-trained LM: MICoL [WWW’22]

- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
Need for Hierarchical Tagging of Unstructured Documents

- Tag each document with **a set of relevant classes** from **a huge candidate pool**

- **Challenges:**
  - Huge label space, multi-label tagging
  - Limited labeled data—hard for supervised models

- **25,000+ candidate classes**
- **15,000+ candidate classes**
TaxoClass: Weakly-supervised Hierarchical Multi-Label Text Classification

- Taxonomy! — Structure the huge label space by organizing classes hierarchically
- DAG structure: Enable fast label space exploration in a top-down way
- Each paper can have multiple categories distributed on different paths
- Facilitate multi-label tagging by capturing class relations

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](https://huggingface.co/roberta-large-mnli)

After reading the premise, can you infer the hypothesis?

Natural Language Inference Model

P(Entails) = 0.9

"This paper is about NLP evaluation"

"Relevance"

"NLP evaluation" Template

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...
**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

---

**Diagram:**

- Document-class Relevance: $rel(D_i, c_j)$
- Relevance Model: (e.g., BM25, doc2vec, BERT-NLI)
- Reduced Label Search Space
  - $rel = 0.75$
  - Learning to Rank
  - Information retrieval
  - Query Expansion
  - Graph Mining
  - Text Mining
  - Data Mining
  - Computer Science
  - Data Mining
  - Text Mining
  - Query Expansion
  - Graph Mining
  - Computer Science
**TaxoClass: A Weakly-Supervised Classification Method based on Taxonomy**

- Shrink the label search space with top-down exploration
- **Identify document core classes** in reduced label search space

---

“Locally” Most Relevant

“Collection-wise” Most Relevant
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: Case Studies

Inspired by principles of behavioral testing in software engineering, we introduce CheckList, a task-agnostic methodology for testing NLP models...

When our son was about 4 months old, doctor said we could give him crafted cereal so we bought it. It digests well and doesn’t lock up his bowels at all...
## TaxoClass: Performance Comparison

<table>
<thead>
<tr>
<th>Methods</th>
<th>Amazon Example-F1</th>
<th>P@1</th>
<th>DBPedia Example-F1</th>
<th>P@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WeSHClass (Meng et al., AAAI’19)</td>
<td>0.246</td>
<td>0.577</td>
<td>0.305</td>
<td>0.536</td>
</tr>
<tr>
<td>SS-PCEM (Xiao et al., WebConf’19)</td>
<td>0.292</td>
<td>0.537</td>
<td>0.385</td>
<td>0.742</td>
</tr>
<tr>
<td>Semi-BERT (Devlin et al., NAACL’19)</td>
<td>0.339</td>
<td>0.592</td>
<td>0.428</td>
<td>0.761</td>
</tr>
<tr>
<td>Hier-0Shot-TC (Yin et al., EMNLP’19)</td>
<td>0.474</td>
<td>0.714</td>
<td>0.677</td>
<td>0.787</td>
</tr>
<tr>
<td>TaxoClass (NAACL’21)</td>
<td>0.593</td>
<td>0.812</td>
<td>0.816</td>
<td>0.894</td>
</tr>
</tbody>
</table>

- **Example-F1** = \( \frac{1}{N} \sum_{i=1}^{N} \frac{2|\text{true}_i \cap \text{pred}_i|}{|\text{true}_i| + |\text{pred}_i|} \)
- **P@1** = \( \frac{\#\text{docs with top-1 pred correct}}{\#\text{total docs}} \)

- **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
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]
  - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
MetaCat: Leveraging 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 heterogeneous signals in the categorization process?

---

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: How to Use this Underlying Model?

- **Embedding** Learning Module
  - All embedding vectors $e_u, e_l, e_d, e_t, e_w$ are parameters of the generative process
  - Learn the embedding vectors through maximizing the likelihood of observing all text and metadata

- **Training Data** Generation Module
  - $e_u, e_l, e_d, e_t, e_w$ have been learned
  - Given a label $l$, generate $d, w$ and $t$ according to the generative process
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.

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

### Table 3: Macro F1 scores of compared algorithms on the five datasets. “-”: excessive memory requirements.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>GitHub-Bio</th>
<th>GitHub-AI</th>
<th>GitHub-Sec</th>
<th>Amazon</th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-based</td>
<td>CNN [12]</td>
<td>0.1896 ± 0.0133</td>
<td>0.1796 ± 0.0216</td>
<td>0.4268 ± 0.0584</td>
<td>0.5056 ± 0.0376</td>
<td>0.2858 ± 0.0559</td>
</tr>
<tr>
<td></td>
<td>HAN [38]</td>
<td>0.0677 ± 0.0208</td>
<td>0.0961 ± 0.0254</td>
<td>0.4095 ± 0.0590</td>
<td>0.4644 ± 0.0597</td>
<td>0.2592 ± 0.0826</td>
</tr>
<tr>
<td></td>
<td>PTE [32]</td>
<td>0.2630 ± 0.0371</td>
<td>0.3363 ± 0.0250</td>
<td>0.3803 ± 0.0218</td>
<td>0.2563 ± 0.0810</td>
<td>0.1739 ± 0.0190</td>
</tr>
<tr>
<td></td>
<td>WeSTClass [23]</td>
<td>0.3414 ± 0.0129</td>
<td>0.4056 ± 0.0248</td>
<td>0.5497 ± 0.0054</td>
<td>0.5234 ± 0.0147</td>
<td>0.3085 ± 0.0398</td>
</tr>
<tr>
<td></td>
<td>PCEM [36]</td>
<td>0.2977 ± 0.0281</td>
<td>0.3751 ± 0.0350</td>
<td>0.4033 ± 0.0336</td>
<td>0.4239 ± 0.0237</td>
<td>0.2039 ± 0.0472</td>
</tr>
<tr>
<td></td>
<td>BERT [4]</td>
<td>0.1740 ± 0.0164</td>
<td>0.2083 ± 0.0415</td>
<td>0.4956 ± 0.0164</td>
<td>0.4911 ± 0.0544</td>
<td>0.2834 ± 0.0550</td>
</tr>
<tr>
<td>Graph-based</td>
<td>ESim [27]</td>
<td>0.2598 ± 0.0182</td>
<td>0.3209 ± 0.0202</td>
<td>0.4672 ± 0.0171</td>
<td>0.5336 ± 0.0220</td>
<td>0.3399 ± 0.0113</td>
</tr>
<tr>
<td></td>
<td>Metapath2vec [5]</td>
<td>0.3214 ± 0.0128</td>
<td>0.3220 ± 0.0290</td>
<td>0.5140 ± 0.0637</td>
<td>0.5239 ± 0.0437</td>
<td>0.3443 ± 0.0208</td>
</tr>
<tr>
<td></td>
<td>HIN2vec [6]</td>
<td>0.2742 ± 0.0136</td>
<td>0.2513 ± 0.0211</td>
<td>0.4000 ± 0.0115</td>
<td>0.4261 ± 0.0284</td>
<td>0.2411 ± 0.0142</td>
</tr>
<tr>
<td></td>
<td>TextGCN [39]</td>
<td>0.4817 ± 0.0078</td>
<td>0.5997 ± 0.0013</td>
<td>–</td>
<td>–</td>
<td>0.3191 ± 0.0029</td>
</tr>
<tr>
<td></td>
<td>MetaCat</td>
<td>0.5230 ± 0.0080</td>
<td>0.6154 ± 0.0079</td>
<td>0.6323 ± 0.0235</td>
<td>0.6496 ± 0.0091</td>
<td>0.5612 ± 0.0067</td>
</tr>
</tbody>
</table>
HIMECat: Jointly Modeling Metadata and Hierarchy

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


(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

**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).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>GitHub</th>
<th></th>
<th>GitHub</th>
<th></th>
<th>Amazon</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Leaf</td>
<td>Overall</td>
<td>Leaf</td>
<td>Overall</td>
<td>Leaf</td>
<td>Overall</td>
</tr>
<tr>
<td></td>
<td>Micro</td>
<td>Macro</td>
<td>Micro</td>
<td>Macro</td>
<td>Micro</td>
<td>Macro</td>
</tr>
<tr>
<td>HierSVM [7]</td>
<td>0.1861*</td>
<td>0.1388*</td>
<td>0.4862*</td>
<td>0.2457**</td>
<td>0.0538*</td>
<td>0.0460*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeSHClass [29]</td>
<td>0.1727*</td>
<td>0.1559*</td>
<td>0.3332*</td>
<td>0.1924**</td>
<td>0.0604*</td>
<td>0.0602*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCEM [48]</td>
<td>0.2519*</td>
<td>0.1234*</td>
<td>0.5299*</td>
<td>0.1786**</td>
<td>0.1090*</td>
<td>0.0717*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HiGitClass [53]</td>
<td>0.3984*</td>
<td>0.3902*</td>
<td>0.5073*</td>
<td>0.4084*</td>
<td>0.1738*</td>
<td>0.1656*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetaCat [51]</td>
<td>0.3762*</td>
<td>0.3403*</td>
<td>0.5411*</td>
<td>0.3863*</td>
<td>0.0790*</td>
<td>0.0768*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metapath2vec [6]</td>
<td>0.2814*</td>
<td>0.2805*</td>
<td>0.4592*</td>
<td>0.3212**</td>
<td>0.1360*</td>
<td>0.1344*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poincaré [32]</td>
<td>0.2750*</td>
<td>0.1980*</td>
<td>0.4302*</td>
<td>0.2218*</td>
<td>0.1336*</td>
<td>0.1296*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT [5]</td>
<td>0.2889*</td>
<td>0.2561*</td>
<td>0.4675*</td>
<td>0.3007*</td>
<td>0.1316*</td>
<td>0.0812*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIMECat</td>
<td>0.4254*</td>
<td>0.4209*</td>
<td>0.5820*</td>
<td>0.4535*</td>
<td>0.2038*</td>
<td>0.1938*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4509*</td>
<td>0.2191*</td>
<td>0.2748*</td>
<td>0.1770*</td>
<td>0.1552*</td>
<td>0.1553*</td>
</tr>
</tbody>
</table>
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]
  - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
MICOIL: Metadata-Induced Contrastive Learning for Zero-Shot Multi-Label Text Classification

- **Input**
  - A set of labels
  - Each label has its name and description
  - A large set of unlabeled documents associated with metadata (e.g., authors, venue, references) that can connect the documents together

- **Output**
  - A multi-label text classifier
  - Given some new documents, the classifier can predict relevant labels for each document

---

- **Input**
  - A set of labels
  - Each label has its name and description
  - A large set of unlabeled documents associated with metadata (e.g., authors, venue, references) that can connect the documents together

- **Output**
  - A multi-label text classifier
  - Given some new documents, the classifier can predict relevant labels for each document

---

Zhang, Y., Shen, Z., Wu, C., Xie, B., Wang, Y., Wang, K. & Han, J. "Metadata-Induced Contrastive Learning for Zero-Shot Multi-Label Text Classification", WWW’22. **Category names and descriptions as supervision.**
Pre-trained Language Models for Multi-Label Text Classification

- If we could have some labeled documents, ...
- We can use relevant (document, label) pairs to fine-tune the pre-trained LM
- Both Bi-Encoder and Cross-Encoder are applicable

However, we do not have any labeled documents!!!
Contrastive learning:

- Instead of training the model to know “what is what” (e.g., relevant (document, label) pairs), train it to know “what is similar with what” (e.g., similar (document, document) pairs)

- Using metadata to define similar (document, document) pairs

---

MICoL: Experimental Results

- MICoL significantly outperforms text-based contrastive learning baselines
- MICoL is competitive with the supervised SOTA trained on 10K–50K labeled documents

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@1</td>
<td>P@3</td>
</tr>
<tr>
<td>Doc2Vec [31]</td>
<td>0.5697**</td>
<td>0.4613**</td>
</tr>
<tr>
<td>SciBERT [2]</td>
<td>0.6440**</td>
<td>0.5030**</td>
</tr>
<tr>
<td>ZeroShot-Entail [61]</td>
<td>0.6649**</td>
<td>0.5003**</td>
</tr>
<tr>
<td>SPECTER [8]</td>
<td>0.7107**</td>
<td>0.5381**</td>
</tr>
<tr>
<td>EDA [53]</td>
<td>0.6442**</td>
<td>0.4939**</td>
</tr>
<tr>
<td>UDA [57]</td>
<td>0.6291**</td>
<td>0.4848**</td>
</tr>
<tr>
<td>MICoL (Bi-Encoder, $P \rightarrow P \rightarrow P$)</td>
<td>0.7062*</td>
<td>0.5369*</td>
</tr>
<tr>
<td>MICoL (Bi-Encoder, $P \rightarrow (PP) \rightarrow P$)</td>
<td>0.7050*</td>
<td>0.5344*</td>
</tr>
<tr>
<td>MICoL (Cross-Encoder, $P \rightarrow P \rightarrow P$)</td>
<td><strong>0.7177</strong></td>
<td><strong>0.5444</strong></td>
</tr>
<tr>
<td>MICoL (Cross-Encoder, $P \rightarrow (PP) \rightarrow P$)</td>
<td>0.7061</td>
<td>0.5376</td>
</tr>
<tr>
<td>MATCH [68] (10K Training)</td>
<td>0.4423**</td>
<td>0.2851**</td>
</tr>
<tr>
<td>MATCH [68] (50K Training)</td>
<td>0.6215**</td>
<td>0.4280**</td>
</tr>
<tr>
<td>MATCH [68] (100K Training)</td>
<td>0.8321</td>
<td>0.6520</td>
</tr>
<tr>
<td>MATCH [68] (Full, 560K+ Training)</td>
<td>0.9114</td>
<td>0.7634</td>
</tr>
</tbody>
</table>
## Summary: Weakly Supervised Text Classification Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Flat vs. Hierarchical</th>
<th>Single-label vs. Multi-label</th>
<th>Supervision Format</th>
<th>Embedding vs. Pretrained LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>WeSTClass</td>
<td>Flat</td>
<td>Single-label</td>
<td>Both types</td>
<td>Embedding</td>
</tr>
<tr>
<td>ConWea</td>
<td>Flat</td>
<td>Single-label</td>
<td>Category Names</td>
<td>Pretrained LM</td>
</tr>
<tr>
<td>LOTClass</td>
<td>Flat</td>
<td>Single-label</td>
<td>Category Names</td>
<td>Pretrained LM</td>
</tr>
<tr>
<td>X-Class</td>
<td>Flat &amp; Hierarchical</td>
<td>Single-label &amp; Path</td>
<td>Category Names</td>
<td>Pretrained LM</td>
</tr>
<tr>
<td>WeSHClass</td>
<td>Hierarchical</td>
<td>Path</td>
<td>Both types</td>
<td>Embedding</td>
</tr>
<tr>
<td>TaxoClass</td>
<td>Hierarchical</td>
<td>Multi-label</td>
<td>Category Names</td>
<td>Pretrained LM</td>
</tr>
<tr>
<td>MetaCat</td>
<td>Flat</td>
<td>Single-label</td>
<td>A Few Labeled Docs</td>
<td>Embedding</td>
</tr>
<tr>
<td>HIMECat</td>
<td>Hierarchical</td>
<td>Path</td>
<td>A Few Labeled Docs</td>
<td>Embedding</td>
</tr>
<tr>
<td>MICoL</td>
<td>Flat</td>
<td>Multi-label</td>
<td>Category Names</td>
<td>Pretrained LM</td>
</tr>
</tbody>
</table>
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]
    - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
Aspect-based Sentiment Analysis

- Task definition: Given an opinionated document about a target entity (e.g., a laptop, a restaurant or a hotel), the goal is to identify the opinion tuple of <aspect, sentiment> of the document

  S1: Mermaid Inn is an overall good restaurant with really good seafood.  (good, food)
  S2: Eye-pleasing with semi-private booths, place for a date.  (good, ambience)
  S3: It’s to die for!

Pure aspect in red; pure opinion in blue; joint topics are underlined and in purple

- Most previous studies deal with the tasks of aspect extraction and sentiment polarity classification individually or sequentially

- Other methods jointly solve these two sub-tasks by first separating target words from opinion words and then learning joint topic distributions over words

- We learn and regularize the joint topics
If the semantics of each joint topic of <sentiment, aspect> can be automatically captured, machines will be able to identify representative terms of the joint topics such as “semi-private” for <good, ambience>.

Thus, it will benefit both aspect extraction and sentiment classification.

Our general idea is to learn and regularize the joint topics in the embedding space to enhance both tasks.
Our Framework: JASen (Joint Aspect Sentiment Analysis)

- Step 1: Leverage the in-domain training corpus and user-given keywords to learn joint topic representation in the word embedding space
- Step 2: Embedding-based prediction on unlabeled data are then leveraged by neural models for pre-training and self-training

Jiaxin Huang, Yu Meng, Fang Guo, Heng Ji and Jiawei Han, "Aspect-Based Sentiment Analysis by Aspect-Sentiment Joint Embedding", EMNLP'20
Joint-Topic Representation Learning

Regularizing Pure Aspect/Sentiment Topics. We regularize the aspect topic embeddings $t_a$ and sentiment topic embeddings $t_s$ so that different topics are pushed apart.

- **Marginal topic regularization:**
  \[
  \mathcal{L}_{reg}^A = - \sum_{a \in A} \sum_{w_i \in l_a} \log P(t_a | w_i) \quad \mathcal{L}_{reg}^S = - \sum_{s \in S} \sum_{w_i \in l_s} \log P(t_s | w_i). \quad P(t | w_i) \propto \exp(u_i^T t)
  \]

- Words can be “classified” into topics based on embedding similarity.

- User-provided keywords are used for initialization, and more keywords are expanded based on cosine similarity in each embedding training epoch.
Joint-Topic Representation Learning

Topic Distribution

\[ P(t|w_i) \]  

Examples:

- **food**
  - good, food + bad, food
- **good**
  - good, food + good, ambience + good, service

- **d:** Mermaid Inn is an overall **good** restaurant with really **good** seafood.

- **P(d|w_i)**
- **P(w_j|w_i)**

- **global context**
- **center word**
- **local context**
- **topic**

- **Regularizing Joint <Sentiment, Aspect> Topics**
- **We connect the learning of joint topic embeddings with pure aspect/sentiment topics by exploring the relationship between marginal distribution and joint distribution.**

\[
P(t_a|w_i) = \sum_{s \in S} P(t_{s,a}|w_i) \quad P(t_s|w_i) = \sum_{a \in A} P(t_{s,a}|w_i)
\]

- **To form the joint topic regularization objective, we can replace the probability term in the pure aspect/sentiment regularization objective with the sum of joint probability**
To evaluate the quality of the joint topic representation, we retrieve their representative terms by ranking the embedding cosine similarity between words and each joint topic vector.

Representative terms are not restricted to be adjectives, such as “vomit” in (bad, food) and “commitment” in (good, support).

“Cramped” appears in both (bad, ambience) in restaurant domain and (bad, keyboard) in laptop domain.
# JASen: Quantitative Evaluation

## Aspect Extraction

<table>
<thead>
<tr>
<th>Methods</th>
<th>Restaurant</th>
<th></th>
<th></th>
<th></th>
<th>Laptop</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
<td>macro-F1</td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
<td>macro-F1</td>
</tr>
<tr>
<td>CosSim</td>
<td>61.43</td>
<td>50.12</td>
<td>50.26</td>
<td>42.31</td>
<td>53.84</td>
<td>58.79</td>
<td>54.64</td>
<td>52.18</td>
</tr>
<tr>
<td>ABAE(He et al., 2017)</td>
<td>67.34</td>
<td>46.63</td>
<td>50.79</td>
<td>45.31</td>
<td>59.84</td>
<td>59.96</td>
<td>59.60</td>
<td>56.21</td>
</tr>
<tr>
<td>CAAt(Tulkens and van Cranenburgh, 2020)</td>
<td>66.30</td>
<td>49.20</td>
<td>50.61</td>
<td>46.18</td>
<td>57.95</td>
<td>65.23</td>
<td>59.91</td>
<td>58.64</td>
</tr>
<tr>
<td>W2VLDA(García-Pablos et al., 2018)</td>
<td>70.75</td>
<td>58.82</td>
<td>57.44</td>
<td>51.40</td>
<td>64.94</td>
<td>67.78</td>
<td>65.79</td>
<td>63.44</td>
</tr>
<tr>
<td>BERT(Devlin et al., 2019)</td>
<td>72.98</td>
<td>58.20</td>
<td>74.63</td>
<td>55.72</td>
<td>67.52</td>
<td>68.26</td>
<td>67.29</td>
<td>65.45</td>
</tr>
<tr>
<td>JASen w/o joint</td>
<td>81.03</td>
<td>61.66</td>
<td>65.91</td>
<td>61.43</td>
<td>69.71</td>
<td>69.13</td>
<td>70.65</td>
<td>67.49</td>
</tr>
<tr>
<td>JASen w/o self train</td>
<td>82.90</td>
<td>63.15</td>
<td>72.51</td>
<td>64.94</td>
<td>70.36</td>
<td>68.77</td>
<td>70.91</td>
<td>68.79</td>
</tr>
<tr>
<td>JASen</td>
<td><strong>83.83</strong></td>
<td><strong>64.73</strong></td>
<td><strong>72.95</strong></td>
<td><strong>66.28</strong></td>
<td><strong>71.01</strong></td>
<td><strong>69.55</strong></td>
<td><strong>71.31</strong></td>
<td><strong>69.69</strong></td>
</tr>
</tbody>
</table>

## Sentiment Polarity Classification

<table>
<thead>
<tr>
<th>Methods</th>
<th>Restaurant</th>
<th></th>
<th></th>
<th></th>
<th>Laptop</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
<td>macro-F1</td>
<td>Accuracy</td>
<td>Precision</td>
<td>Recall</td>
<td>macro-F1</td>
</tr>
<tr>
<td>CosSim</td>
<td>70.14</td>
<td>74.72</td>
<td>61.26</td>
<td>59.89</td>
<td>68.73</td>
<td>69.91</td>
<td>68.95</td>
<td>68.41</td>
</tr>
<tr>
<td>W2VLDA</td>
<td>74.32</td>
<td>75.66</td>
<td>70.52</td>
<td>67.23</td>
<td>71.06</td>
<td>71.62</td>
<td>71.37</td>
<td>71.22</td>
</tr>
<tr>
<td>BERT</td>
<td>77.48</td>
<td>77.62</td>
<td>73.95</td>
<td>73.82</td>
<td>69.71</td>
<td>70.10</td>
<td>70.26</td>
<td>70.08</td>
</tr>
<tr>
<td>JASen w/o joint</td>
<td>78.07</td>
<td>80.60</td>
<td>72.40</td>
<td>73.71</td>
<td>72.31</td>
<td>72.34</td>
<td>72.25</td>
<td>72.26</td>
</tr>
<tr>
<td>JASen w/o self train</td>
<td>79.16</td>
<td>81.31</td>
<td>73.94</td>
<td>75.34</td>
<td>73.29</td>
<td>73.69</td>
<td>73.42</td>
<td>73.24</td>
</tr>
<tr>
<td>JASen</td>
<td><strong>81.96</strong></td>
<td><strong>82.85</strong></td>
<td><strong>78.11</strong></td>
<td><strong>79.44</strong></td>
<td><strong>74.59</strong></td>
<td><strong>74.69</strong></td>
<td><strong>74.65</strong></td>
<td><strong>74.59</strong></td>
</tr>
</tbody>
</table>
Joint Topic Representation Visualization

- Visualization of joint topics (purple stars), aspect topics (red crosses) and sentiment topics (blue dots) in the embedding space

- An interesting observation: Some aspect topics (e.g., ambience) lie approximately in the middle of their joint topics ("good, ambience" and "bad, ambience"), showing that our embedding learning objective understands the joint topics as decomposition of their "marginal" topics
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]
  - Pre-trained LM: MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment Analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
Mining Text Outliers in Document Directories

- People organize docs in directory structures
- Outliers: A doc may be put in wrong folders
- A doc $d_i$ may have two kinds of errors
  - (O) Out-of-distribution: $d_i$ does not belong to any existing folder in the directory
  - (M) Misclassification: $d_i$ belongs to another folder
- Mining text outliers from massive doc directories
- Considering both error types
- No human supervision
- New proximity-based algorithm: $kj$-Nearest Neighbours ($kj$-NN)
- Self-supervision, exploiting semantic similarities, estimating the relevance of the original labels

Messages are often put into wrong Email folders

Edouard Fouche, Yu Meng, Fang Guo, Honglei Zhuang, Klemens Boehm, and Jiawei Han, "Mining Text Outliers in Document Directories", ICDM'20
- Initial (imperfect) classification of docs into a finite set of classes C, denoted as $D \rightarrow C$
- Joint embedding: Segment the phrases $P$ from every doc using AutoPhrase and obtain the phrase and doc embeddings $V: O \rightarrow R^n$ in a joint spherical space via JoSE
- Self-supervision: Estimate the representativeness of each phrase $p \in P$ w.r.t. each class $c \in C$, denoted as a function $r: P \times C \rightarrow R^+$
- Mining class representativeness: based on integrity, popularity & distinctiveness
Infer the class of each element (doc) based not only on the class of its $k$ nearest docs but also on their relevance (i.e., average representativeness of its $j$ nearest sub-elements (phrases) for its class)

- Type M outlier: If a doc $d$ is closer to docs which are (i) relevant, and (ii) from another single class
- Type O outlier: if $d$ is similarly close to relevant documents of various classes
- New $K_j$-NN Algorithm evaluates $k$ nearest docs and $j$ nearest phrases
Comparison of Outlier Detection Methods: Type O, NYT

- **New:** Kj-NN Algorithm
- k nearest docs
- j nearest phrases
- Quantitative evaluation
  - LOF (local outlier factor)
  - RS-Hash (Randomised Subspace Hashing)
  - ANCS (Average Negative Cosine Similarity)
  - VMF-Q
  - TONMF
  - CVDD (A PLM-based )

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>AP</th>
<th>R1</th>
<th>R2</th>
<th>R5</th>
<th>AUC</th>
<th>AP</th>
<th>R1</th>
<th>R2</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>66.45</td>
<td>1.62</td>
<td>3.00</td>
<td>3.00</td>
<td>9.00</td>
<td>60.91</td>
<td>1.77</td>
<td>2.00</td>
<td>6.00</td>
<td>12.00</td>
</tr>
<tr>
<td>RS-Hash</td>
<td>46.62</td>
<td>0.87</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>46.14</td>
<td>0.90</td>
<td>0.00</td>
<td>0.00</td>
<td>2.00</td>
</tr>
<tr>
<td>ANCS</td>
<td>66.63</td>
<td>2.57</td>
<td>6.00</td>
<td>10.00</td>
<td>21.00</td>
<td>63.94</td>
<td>4.35</td>
<td>14.00</td>
<td>16.00</td>
<td>20.00</td>
</tr>
<tr>
<td>k-ANCS</td>
<td>83.89</td>
<td>3.65</td>
<td>3.00</td>
<td>7.00</td>
<td>19.00</td>
<td>81.08</td>
<td>4.43</td>
<td>12.00</td>
<td>14.00</td>
<td>20.00</td>
</tr>
<tr>
<td>TONMF</td>
<td>58.66</td>
<td>7.30</td>
<td>0.00</td>
<td>2.00</td>
<td>9.00</td>
<td>61.42</td>
<td>31.01</td>
<td>0.90</td>
<td>0.90</td>
<td>4.90</td>
</tr>
<tr>
<td>VMF-Q</td>
<td>76.34</td>
<td>2.21</td>
<td>2.00</td>
<td>3.00</td>
<td>11.00</td>
<td>85.76</td>
<td>4.00</td>
<td>6.00</td>
<td>8.00</td>
<td>22.00</td>
</tr>
<tr>
<td>CVDD</td>
<td>78.47</td>
<td>7.75</td>
<td>11.7</td>
<td>18.09</td>
<td>22.34</td>
<td>76.29</td>
<td>15.89</td>
<td>21.62</td>
<td>27.03</td>
<td>29.73</td>
</tr>
<tr>
<td>kj-NN</td>
<td>92.51</td>
<td>17.57</td>
<td>25.00</td>
<td>39.20</td>
<td>61.40</td>
<td>93.33</td>
<td>27.76</td>
<td>37.20</td>
<td>46.80</td>
<td>62.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>AP</th>
<th>R1</th>
<th>R2</th>
<th>R5</th>
<th>AUC</th>
<th>AP</th>
<th>R1</th>
<th>R2</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>60.66</td>
<td>2.45</td>
<td>2.00</td>
<td>2.50</td>
<td>6.00</td>
<td>74.76</td>
<td>4.12</td>
<td>5.00</td>
<td>5.00</td>
<td>15.00</td>
</tr>
<tr>
<td>RS-Hash</td>
<td>48.05</td>
<td>1.87</td>
<td>0.50</td>
<td>1.00</td>
<td>5.50</td>
<td>42.91</td>
<td>0.89</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ANCS</td>
<td>67.59</td>
<td>5.19</td>
<td>8.00</td>
<td>12.00</td>
<td>18.00</td>
<td>78.52</td>
<td>5.60</td>
<td>10.00</td>
<td>15.00</td>
<td>40.00</td>
</tr>
<tr>
<td>k-ANCS</td>
<td>82.51</td>
<td>5.94</td>
<td>3.00</td>
<td>5.50</td>
<td>14.50</td>
<td>88.56</td>
<td>6.59</td>
<td>15.00</td>
<td>20.00</td>
<td>30.00</td>
</tr>
<tr>
<td>TONMF</td>
<td>54.61</td>
<td>1.78</td>
<td>2.50</td>
<td>3.00</td>
<td>9.00</td>
<td>64.94</td>
<td>6.35</td>
<td>0.00</td>
<td>0.00</td>
<td>15.00</td>
</tr>
<tr>
<td>VMF-Q</td>
<td>83.67</td>
<td>6.87</td>
<td>4.00</td>
<td>11.00</td>
<td>20.00</td>
<td>83.98</td>
<td>4.59</td>
<td>5.00</td>
<td>10.00</td>
<td>20.00</td>
</tr>
<tr>
<td>CVDD</td>
<td>73.10</td>
<td>10.00</td>
<td>11.58</td>
<td>14.74</td>
<td>22.11</td>
<td>88.83</td>
<td>24.00</td>
<td>25.00</td>
<td>25.00</td>
<td>25.00</td>
</tr>
<tr>
<td>kj-NN</td>
<td>94.51</td>
<td>42.64</td>
<td>30.40</td>
<td>44.50</td>
<td>64.50</td>
<td>91.42</td>
<td>6.57</td>
<td>3.00</td>
<td>11.00</td>
<td>38.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>AP</th>
<th>R1</th>
<th>R2</th>
<th>R5</th>
<th>AUC</th>
<th>AP</th>
<th>R1</th>
<th>R2</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOF</td>
<td>52.28</td>
<td>5.44</td>
<td>1.80</td>
<td>3.40</td>
<td>7.00</td>
<td>77.93</td>
<td>2.77</td>
<td>0.00</td>
<td>0.00</td>
<td>20.00</td>
</tr>
<tr>
<td>RS-Hash</td>
<td>48.76</td>
<td>4.58</td>
<td>0.80</td>
<td>1.40</td>
<td>2.80</td>
<td>56.94</td>
<td>1.76</td>
<td>0.00</td>
<td>0.00</td>
<td>10.00</td>
</tr>
<tr>
<td>ANCS</td>
<td>67.67</td>
<td>10.13</td>
<td>6.60</td>
<td>9.80</td>
<td>19.20</td>
<td>83.89</td>
<td>13.65</td>
<td>30.00</td>
<td>40.00</td>
<td>40.00</td>
</tr>
<tr>
<td>k-ANCS</td>
<td>75.56</td>
<td>10.45</td>
<td>3.40</td>
<td>6.40</td>
<td>11.40</td>
<td>91.37</td>
<td>10.93</td>
<td>30.00</td>
<td>30.00</td>
<td>30.00</td>
</tr>
<tr>
<td>TONMF</td>
<td>52.95</td>
<td>1.50</td>
<td>1.40</td>
<td>2.40</td>
<td>6.40</td>
<td>71.13</td>
<td>29.92</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>VMF-Q</td>
<td>77.11</td>
<td>12.92</td>
<td>4.60</td>
<td>7.40</td>
<td>15.60</td>
<td>63.39</td>
<td>2.55</td>
<td>10.00</td>
<td>20.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CVDD</td>
<td>72.81</td>
<td>18.69</td>
<td>9.04</td>
<td>13.05</td>
<td>21.49</td>
<td>85.13</td>
<td>44.99</td>
<td>42.86</td>
<td>42.86</td>
<td>42.86</td>
</tr>
<tr>
<td>kj-NN</td>
<td>97.04</td>
<td>71.69</td>
<td>19.12</td>
<td>36.96</td>
<td>68.28</td>
<td>91.52</td>
<td>8.45</td>
<td>10.00</td>
<td>16.00</td>
<td>38.00</td>
</tr>
</tbody>
</table>
Comparison of Outlier Detection Methods: Type M, NYT

- No approach explicitly handle Type M outlier detection, but we can adapt any supervised classifier to this task
- Word-level CNN (W-CNN)
- Very Deep CNN (VD-CNN)
- Attention-Based Hierarchical RNN (AT-RNN)
- RCNN (combines both bi-directional recurrent structures and max-pooling)

- Note: Most existing outlier analysis methods (e.g., LOF) consider only Type O outliers
- Existence of Type M outliers increases the difficulty of clustering/modeling
Case Study of Text Outlier Detection (I)

**Type O (Education):** NYC will build a new home for one of its premier high schools, [...] under a schedule that seeks to show that its public schools can be built fast and well, Mayor Koch and Governor Cuomo said yesterday. The new school, incorporating the latest in modern laboratory equipment, fiber optic systems and an Olympic size swimming pool will be built [...] in lower Manhattan, with work to begin at the end of next year...

<table>
<thead>
<tr>
<th>1st-NN (Business):</th>
<th>2nd-NN (Politics): Praising the work of young scientists and inventors [...] , President Obama on monday announced a broad plan to create [...] initiatives designed to encourage children to study science, technology, engineering and mathematics. [...] Obama said he was committed to giving students the resources they need to pursue education...</th>
</tr>
</thead>
</table>

**Top phrases:** City, state, program, buildings, education, office, schools, year, project, company...

**Type M (Sport → Business):** Bellevue, Washington. Set between an indoor tennis club and a home appliance showroom, dozens of engineers, physicists and nuclear experts are chasing a radical dream of Bill Gates. The quest is for a new kind of nuclear reactor that would be fueled by today’s nuclear waste, supply all the electricity in the United States for the next 800 years and, possibly, cut the risk of nuclear weapons proliferation around the world...

<table>
<thead>
<tr>
<th>1st-NN (Science):</th>
<th>2nd-NN (Business): Waste management companies and the energy industries have long experimented with converting methane [...] into transportation fuel. Those efforts have met with mixed success, and a renewable natural gas fuel has not been widely available in the United States. But now, one leading supplier [...] is taking a big step toward changing that.</th>
</tr>
</thead>
</table>

**Top phrases:** Natural gas, reactor, energy, fuel, plant, nuclear power, electricity, project, company, United States...

**3rd-NN (Education):** After [...] intense political pressure [...] , schools chancellor Rudy Crew [...] said he would accept the candidate. Dr. Crew had provoked harsh criticism last month when [...] he used his new veto power [...] to reject Claire McIntee, an elementary school principal who was unanimously selected [...] to be the district’s top administrator...

**3rd-NN (Business):** Hoping to give new meaning to the term natural light, a small group of biotechnology hobbyists and entrepreneurs has started a project to develop plants that glow, potentially leading the way for trees that can replace electric streetlamps [...] . Rather than being the work of a corporation [...] , it will be done by a small group of hobbyist scientists...

The Type O outlier relates in some way to its nearest documents: They all relate to building construction projects or political decisions in education (or both). The Type M outlier, which was wrongly classified into the folder ‘Sport’, the nearest documents all relate to science (in particular, energy generation technologies) and business.
Case Study of Text Outlier Detection (II)

On the ARXIV dataset: The Type O outliers are not paper abstracts, so one should remove them from the data set. The Type M outlier may arguably belong to the category ‘Computation and Language’, instead of ‘Artificial Intelligence’.

- Examples of Type O on Arxiv look more interesting than that on NYT
- Type O detection criteria may need some rethinking?
  - Type O outlier: if $d$ is similarly close to relevant documents of various classes
- Type M or O outliers should be removed once detected for subsequent re-evaluation
Summary: Weakly-Supervised Text Classification

- Flat Text Classification
  - WeSTClass [CIKM’18], ConWea [ACL’20], LOTClass [EMNLP’20], X-Class [NAACL’21]
- Text Classification with Taxonomy Information
  - WeSHClass [AAAI’19], TaxoClass [NAACL’21]
- Text Classification with Metadata Information
  - MetaCat [SIGIR’20], HIMECat [WSDM’21], MICoL [WWW’22]
- Text Classification: Innovative Applications
  - Aspect-based Sentiment analysis [EMNLP’20]
  - Mining Text Outlier in Document Directories [ICDM’20]
- Weakly supervised text classification
  - Further study to make it competitive with supervised methods
  - Exploring its broad applications
References

- Edouard Fouche, Yu Meng, Fang Guo, Honglei Zhuang, Klemens Boehm, and Jiawei Han, "Mining Text Outliers in Document Directories", ICDM’20
- Jiaxin Huang, Yu Meng, Fang Guo, Heng Ji and Jiawei Han, "Aspect-Based Sentiment Analysis by Aspect-Sentiment Joint Embedding", EMNLP’20
- 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
- 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