Efficient Machine Teaching Frameworks for Natural Language Processing

Giannis Karamanolakis

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2022

Abstract

Efficient Machine Teaching Frameworks for Natural Language Processing

Giannis Karamanolakis

The past decade has seen tremendous growth in potential applications of language technologies in our daily lives due to increasing data, computational resources, and user interfaces. An important step to support emerging applications is the development of algorithms for processing the rich variety of human-generated text and extracting relevant information.

Machine learning, especially deep learning, has seen increasing success on various text benchmarks. However, while standard benchmarks have static tasks with expensive human-labeled data, real-world applications are characterized by dynamic task specifications and limited resources for data labeling, thus making it challenging to transfer the success of supervised machine learning to the real world. To deploy language technologies at scale, it is crucial to develop alternative techniques for teaching machines beyond data labeling.

In this dissertation, we address this data labeling bottleneck by studying and presenting resource-efficient frameworks for teaching machine learning models to solve language tasks across diverse domains and languages. Our goal is to (i) support emerging real-world problems without the expensive requirement of large-scale manual data labeling; and (ii) assist humans in teaching machines via more flexible types of interaction. Towards this goal, we describe our collaborations with experts across domains (including public health, earth sciences, news, and e-commerce) to integrate weakly-supervised neural networks into opera-

tional systems, and we present efficient machine teaching frameworks that leverage flexible forms of declarative knowledge as supervision: coarse labels, large hierarchical taxonomies, seed words, bilingual word translations, and general labeling rules.

First, we present two neural network architectures that we designed to leverage weak supervision in the form of coarse labels and hierarchical taxonomies, respectively, and highlight their successful integration into operational systems. Our Hierarchical Sigmoid Attention Network (HSAN) learns to highlight important sentences of potentially long documents without sentence-level supervision by, instead, using *coarse-grained supervision* at the document level. HSAN improves over previous weakly-supervised learning approaches across sentiment classification benchmarks and has been deployed to help inspections in health departments for the discovery of foodborne illness outbreaks. We also present TXtract, a neural network that extracts attributes for e-commerce products from thousands of diverse categories without using manually labeled data for each category, by instead considering category relationships in a *hierarchical taxonomy*. TXtract is a core component of Amazon's AutoKnow, a system that collects knowledge facts for over 10K product categories, and serves such information to Amazon search and product detail pages.

Second, we present architecture-agnostic machine teaching frameworks that we applied across domains, languages, and tasks. Our weakly-supervised co-training framework can train any type of text classifier using just a small number of class-indicative *seed words* and unlabeled data. In contrast to previous work that use seed words to initialize embedding layers, our iterative seed word distillation (ISWD) method leverages the predictive power of seed words as supervision signals and shows strong performance improvements for aspect detection in reviews across domains and languages. We further demonstrate the cross-lingual transfer abilities of our co-training approach via cross-lingual teacher-student (CLTS), a method for training document classifiers across diverse languages using labeled documents only in English and a limited budget for *bilingual translations*. Not all classification tasks, however, can be effectively addressed using human supervision in the form of seed words.

To capture a broader variety of tasks, we present weakly-supervised self-training (ASTRA), a weakly-supervised learning framework for training a classifier using more general *labeling rules* in addition to labeled and unlabeled data. As a complete set of accurate rules may be hard to obtain all in one shot, we further present an *interactive framework* that assists human annotators by automatically suggesting candidate labeling rules.

In conclusion, this thesis demonstrates the benefits of teaching machines with different types of interaction than the standard data labeling paradigm and shows promising results for new applications across domains and languages. To facilitate future research, we publish our code implementations and design new challenging benchmarks with various types of supervision. We believe that our proposed frameworks and experimental findings will influence research and will enable new applications of language technologies without the costly requirement of large manually labeled datasets.

Table of Contents

List of Figures

List of Tables

Acknowledgements

My experience during the past five years at Columbia was rich and rewarding, and for that I owe a great debt to many people who contributed to this journey.

First of all, I am deeply grateful to my advisors, Luis Gravano and Daniel Hsu, for their constant support and exceptional guidance. Luis and Daniel have influenced every aspect of my experience at Columbia and have taught me valuable lessons that have impacted my research career and life. Luis has taught me to strive to find the big picture and at the same time pay attention to detail in my research. There has not been a single word in this dissertation left unread by Luis, which is just one among the many examples proving Luis's respect to me as his student. Daniel has taught me to think clearly and express my thoughts precisely. Every word by Daniel can reveal new insights and perspectives, and his feedback has been crucial to shape core ideas in this dissertation. I have been extremely lucky to be jointly advised by Luis and Daniel and truly appreciate that they gave me the freedom to explore my own research interests and the opportunity to mentor talented students.

The completion of my Ph.D. would have never been possible without the support of other mentors. I would like to thank Luna Dong, Kathy McKeown, and Smaranda Muresan for serving on my dissertation committee and offering honest feedback and insightful advice. I would also like to thank Ahmed Hassan Awadallah, Luna Dong, Jun Ma, Subhabrata Mukherjee, and Guoqing Zheng for their support and guidance during my internships at Amazon (in 2019) and Microsoft Research (in 2020).

I would also like to thank my wonderful collaborators and co-authors: Ahmed Hassan

Awadallah, Alina Beygelzimer, Ivy Cao, Haw-Shiuan Chang, Lydia Chilton, Luna Dong, Tom Effland, Daniel Hsu, Christos Faloutsos, Lampros Flokas, Becca Funke, Luis Gravano, Katy Gero, Tony Jebara, Andrey Kan, Daniel Khashabi, Ziyi Liu, Zizhou Liu, Jun Ma, Yuning Mao, Max Mauerman, Andrew McCallum, Swaroop Mishra, Tejit Pabari, Subhabrata Mukherjee, Samuel Raab, Jakwanul Safin, Kai Shu, Da Tang, Elizabeth Tellman, Yaqing Wang, Yizhong Wang, Keyang Xu, Tong Zhao, Guoqing Zheng. They have introduced me to fascinating research problems and have tremendously influenced my research.

I am also grateful to many talented colleagues and professors at Columbia: Tariq Alhindi, Sakhar Alkhereyf, Emily Allaway, Jaan Altosaar, David Blei, Nishi Cestero, Tuhin Chakrabarty, Yiru Chen, Samuel Deng, Tom Effland, Noura Farra, Lampros Flokas, Davide Giri, Kira Goldner, Christopher Hidey, Zachary Huang, John Hui, Junyoung Kim, Faisal Ladhak, Mathias Lécuyer, Fei-Tzin Lee, Dan Mitropolsky, Haneen Mohammed, John Paparrizos, Savvas Petridis, Orestis Polychroniou, Fotis Psallidas, Mohammad Sadegh Rasooli, Kenneth Ross, Maja Rudolph, Charlie Summer, Chris Tosh, Elsbeth Turcan, Clayton Sanford, Kevin Shi, Sandip Sinha, Victor Soto, Ana Stoica, Emmanouil Vlatakis, Keyon Vafa, Kiran Vodrahalli, Olivia Winn, Eugene Wu, Keyang Xu, Ji Xu, and Carolina Zheng. I thank them all for sharing their knowledge and perspective during stimulating discussions.

My time in New York would not have been such enjoyable without the warm company of Vaggelis Atlidakis, Sofia Bakogianni, Vaggos Chatziafratis, Marios Georgiou, Theano Dimitraki, Lampros Flokas, Gregory Karageorge, Marilena Karakatsani, Ioannis Manousakis, Matt Morse, Ioannis Petromichelakis, Orestis Plevrakis, Marios Pomonis, Apostolis Psarros, Petros Sousouris, Ioanna Tzialla, Konstantina Tzialla, and Dimos Vogdanos. I am grateful to all of them for their friendship and support. New York gave me exceptional friends who were the most important factor in my decision to stay in the city after my Ph.D. studies.

Finally, I would like to thank my partner Katerina, my friends back home and elsewhere, and my family for their love, care, and support throughout this exciting journey. Words could never express my heartfelt gratitude to all of them.

This work was supported by the National Science Foundation under Grant No. IIS-15-63785. I am also grateful to the Gerondelis Foundation and Leventis Foundation for their scholarships supporting my research.

Dedication

For my father, mother, and sister...

Chapter 1: Introduction

In this dissertation, we study and present resource-efficient machine teaching frameworks with the purpose to (i) support emerging real-world problems without the expensive requirement of large-scale manual data labeling; and (ii) assist humans in teaching machines via more flexible types of interaction.

The past decade has seen tremendous growth in potential applications of language technologies in our daily lives due to the proliferation of online data (e.g., news articles, social media comments, and product reviews), the increasing availability of computational resources, and new user interfaces. For example, health departments nationwide have started to analyze social media content with the goal to detect (possibly rare) incidents related to public health. As another example, companies are investing in automatic tools for the analysis of positive and negative opinions mentioned in customer reviews about their products. As time progresses, having access to increasing amounts of unstructured data from diverse populations creates the opportunity for language technologies to have an impact in the real world.

An important step to support emerging problems in new areas of technology is the development of efficient Natural Language Processing (NLP) algorithms that can extract relevant information from the rich variety of text across domains and languages. Machine learning, especially supervised deep learning, has seen increasing success on various NLP benchmarks [Sang and De Meulder, 2003; Socher et al., 2013; Wang et al., 2018; Wang et al., 2019]. Recent progress in representation learning algorithms [Mikolov et al., 2013a; Peters et al., 2018; Devlin et al., 2019] in conjunction with the development of neural architectures [Kim, 2014; Wieting and Gimpel, 2017; Yang et al., 2016; Vaswani et al., 2017; Radford et al., 2018] have led to important performance gains compared to rule-based and

traditional learning techniques.

Given the success of supervised machine learning in standard NLP benchmarks, these techniques are promising to address emerging tasks. Training machine learning algorithms for a new task requires three main components: the model to train, the hardware to train on, and the data to train with. In the past years, there has been increasing availability of open-source frameworks for developing machine learning models [Paszke et al., 2017; Abadi et al., 2015; Wolf et al., 2019] and of pre-trained state-of-the-art models in online hubs.[1][2][3] At the same time, there has been tremendous progress in the availability of hardware, for example via cloud computing, which provides access to GPUs for training deep neural networks with a typical cost of less than a dollar per hour.[4][5][6] While model architectures and hardware are usually available, significant effort is often required to collect the data for training the models, which presents the main bottleneck in deploying supervised machine learning into the real world, as described next.

Supervised machine learning models require large, hand-labeled training datasets, which are both expensive and time-consuming to obtain for every new task. For example the SST benchmark [Socher et al., 2013] used for sentiment classification comes with 200,000 labeled sentences and the CoNLL benchmark [Sang and De Meulder, 2003] for named-entity-recognition comes with 3 million labeled words. While existing benchmarks include already-labeled data, it is prohibitively expensive to obtain large-scale labeled data for every new application, especially for applications that require domain expertise. Also, benchmark datasets are static, while emerging applications are characterized by dynamic task specifications. For example, changing the task definition from sentence- to phrase-level classification

[1] `https://www.tensorflow.org/hub`

[2] `https://pytorch.org/hub`

[3] `https://huggingface.co/models`

[4] `https://aws.amazon.com/pricing/`

[5] `https://azure.microsoft.com/en-us/pricing/`

[6] `https://cloud.google.com/pricing/`

in SST would require collecting new phrase-level labels from scratch. Dynamic task specification make it challenging to transfer the success of supervised machine learning to the real world.

Unsupervised learning approaches, such as clustering and topic modeling, aim to learn the structure of the dataset without expensive labeled data by instead using unlabeled data that is plentiful in most applications at no cost [Lloyd, 1982; MacQueen, 1967; Blei et al., 2003; Griffiths et al., 2003; He et al., 2017]. While there exist optimal algorithms for unsupervised learning, an important issue is that the structure discovered by these algorithms is not necessarily aligned with the user's needs. For example, the topics learned by unsupervised neural topic models are not perfectly aligned with the classes of interest for the target problem, so substantial human effort is required for interpreting and mapping the learned topics to meaningful aspects. At least minimum human supervision is required to guide the learning algorithm to address the target problem.

There have been several minimally-supervised learning approaches that attempt to reduce the amount of labeled training examples by considering unlabeled data, auxiliary domains, and tasks. Semi-supervised learning approaches leverage unlabeled data that are usually abundant (in contrast to labeled data) with additional statistical assumptions about how unlabeled data can be useful for the model [Blum and Mitchell, 1998; Joachims et al., 1999; Nigam et al., 2000; Nigam and Ghani, 2000; Zhu et al., 2003; Seeger, 2006; Zhou and Li, 2005; Raghavan et al., 2006; Raghavan and Allan, 2007; Small et al., 2011; Clark et al., 2018; Ruder and Plank, 2018; Berthelot et al., 2019]. Transfer learning approaches use labeled datasets from similar domains (domain adaptation) or tasks (multi-task learning) by assuming that such datasets can provide useful training signals for the target task [Caruana, 1997; Daumé III, 2007; Pan and Yang, 2009; Wan, 2009; Artetxe and Schwenk, 2019; Zhang and Yang, 2021]. Unsupervised pre-training approaches follow a sequential transfer learning paradigm by first pre-training models in vast amounts of already-available unlabeled texts (e.g., Wikipedia articles) with unsupervised training objectives (e.g., language modeling) and

then adapting the pre-trained models for the target task with the hope that the representations learned in the pre-training step are useful for the second step [Mikolov et al., 2013a; Pennington et al., 2014; Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018]. The above minimally-supervised learning approaches have achieved better performance across diverse NLP tasks with fewer labeled examples than supervised learning approaches [Wang et al., 2018; Wang et al., 2019; Hedderich et al., 2021a].

While there has been significant progress in addressing the labeled data bottleneck by reducing the amount of labeled data, another way to "expand the neck of the bottle" is to expand the types of interaction for humans to teach machines. The vast majority of the above minimally-supervised learning approaches support just a single type of interaction, that is to label individual instances, one at a time, with task labels. To understand why data labeling is not a scalable approach, consider the (binary) classification task of detecting rare diseases discussed in online documents. While health experts might already know specific symptoms of each disease, by following the dominant data labeling paradigm they would have to label many documents with a disease label to teach a model to associate the right symptoms with the right diseases. On the other hand, supporting richer types of supervision in a declarative form indicating the experts' intents about the model's behavior (e.g., a rule indicating how to address texts mentioning specific symptoms) could improve the efficiency of the teaching process. While a binary label covers a single text instance, such high-level predicates can cover multiple instances and as a result provide more powerful training signal.

Expanding the types of interaction for humans to teach machine learning models is a challenging and under-explored area with limited evidence of success. Existing approaches use specific types of declarative expert knowledge (e.g., keywords, regular expressions) in simple classes of models (e.g., probabilistic topic models) [Druck et al., 2008; Melville et al., 2009; Ganchev et al., 2010; Mann and McCallum, 2010; Lu et al., 2011; Settles, 2011; Jagarlamudi et al., 2012; Augenstein et al., 2016; Poulis and Dasgupta, 2017; Dasgupta et al., 2018]. However, most of these approaches cannot be directly combined with recent state-of-the-art

techniques for representation learning as it is not clear how to integrate declarative knowledge with the black-box neural network architectures, the backbone of modern representation learning. Additionally, most of these techniques are evaluated for just a small number of benchmarks in the English language, thus there is insufficient evidence of whether existing approaches can be applied at scale, across diverse domains, languages, and tasks. Therefore, to address the above limitations and transfer the success of deep neural networks from NLP benchmarks to the real world, it is important to develop new techniques for teaching machines with flexible types of interaction.

Motivated by the demand for resource-efficient frameworks for training accurate models, in this dissertation we investigate the design of frameworks for teaching machines with alternative types of human supervision. Our goal is to (i) support new applications across domains and languages without the expensive need of manually labeled data; and (ii) support more flexible types of interaction for humans to teach machines. Towards this goal, we summarize our collaborations with experts across domains (including public health and e-commerce) to integrate weakly-supervised neural networks into operational systems, and present efficient machine teaching frameworks that leverage flexible forms of declarative knowledge as supervision: coarse labels, large hierarchical taxonomies, seed words, bilingual word translations, and general labeling rules.

First, we present two neural network architectures that we designed to leverage weak supervision in the form of coarse labels and hierarchical taxonomies, respectively, and highlight their successful integration into operational systems. Our Hierarchical Sigmoid Attention Network (HSAN) learns to highlight important sentences of potentially long documents without sentence-level supervision by instead using coarse-grained supervision at the document level. HSAN improves over previous weakly-supervised learning approaches across sentiment classification benchmarks and has been deployed to help inspections in health departments for the discovery of foodborne illness outbreaks. We also developed TXtract, a neural network that extracts attributes for e-commerce products from thousands of diverse

product categories without using manually labeled data for each category, by instead considering category relationships in a hierarchical taxonomy. TXtract is a core component of Amazon's AutoKnow, a system that collects knowledge facts for over 10K product categories, and serves such information to Amazon search and product detail pages.

Second, we present architecture-agnostic machine teaching frameworks that we applied across domains, languages, and tasks. Our weakly-supervised co-training framework can train any type of text classifier using just a small number of class-indicative seed words and unlabeled data. In contrast to previous work that use seed words to initialize embedding layers, our iterative seed word distillation method, ISWD, leverages the predictive power of seed words as supervision signals and shows strong performance improvements for aspect detection in reviews across domains and languages. We further demonstrate the cross-lingual transfer abilities of our co-training approach via our cross-lingual teacher-student method, CLTS, which trains document classifiers across diverse languages using labeled documents only in English and a limited budget for bilingual translations. Not all classification tasks, however, can be effectively addressed using human supervision in the form of seed words. To capture a broader variety of tasks, we present weakly-supervised self-training, or ASTRA, a framework for training any type of classifier using general labeling rules, few labeled data, and unlabeled data. As a complete set of accurate rules may be hard to obtain at once, we further present an interactive framework that assists human annotators by automatically suggesting candidate labeling rules.

Specifically, this dissertation presents the following key contributions:

- **Fine-grained classification with coarse-grained labels:** In Chapter 3, we address the problem of phrase- and sentence-level classification using only coarse-grained supervision at the document level. We present a novel neural network for fine-grained classification using coarse-grained labels through a sigmoid attention mechanism, demonstrate its advantages across multiple benchmarks, and deploy it for daily inspections in health departments. Our Hierarchical Sigmoid Attention Network (HSAN) uses the

sigmoid attention mechanism as the aggregation function for Multiple Instance Learning (MIL), and improves over previous MIL-based approaches [Kotzias et al., 2015; Angelidis and Lapata, 2018a]; HSAN has been deployed to help inspections in health departments for the discovery of foodborne illness outbreaks.

- **Knowledge extraction with hierarchical taxonomies of product categories:** In Chapter 4, we address the problem of extraction of product attributes from online product descriptions from thousands of product categories. We present a novel neural network that jointly extracts attribute values across all product categories by leveraging their relationships in a hierarchical taxonomy, and demonstrate its advantages over 4,000 product categories at Amazon.com. While previous work focuses on a single product category [Zheng et al., 2018; Xu et al., 2019], our TXtract network leverages Amazon's taxonomy with thousands of diverse categories and effectively extracts attribute values without manually labeled data. We further demonstrate the integration of TXtract into Amazon's AutoKnow, a system that collects knowledge facts for over 10K product categories and serves such information to Amazon search and product detail pages.

- **Weakly-supervised text classification with seed words:** In Chapter 5, we address the problem of text classification using just a small number of class-indicative seed words. We present iterative seed word distillation, ISWD, a method for training any type of classifier using just seed words and unlabeled data. While previous work uses seed words to initialize neural networks [Lund et al., 2017; Angelidis and Lapata, 2018b], ISWD leverages the predictive power of seed words *during training* through a teacher-student co-training approach. We evaluate ISWD on fine-grained aspect detection in product and restaurant reviews and demonstrate its potential for more text classification applications.

- **Cross-lingual transfer of weak supervision with minimal resources:** In Chap-

ter 6, we address the problem of training document classifiers across diverse languages using labeled data only in English. We present a novel method that transfers weak supervision across languages using minimal cross-lingual resources, evaluate its performance on four tasks and eighteen languages, and suggest further improvements using richer resources. Our cross-lingual teacher-student (CLTS) approach extracts and transfers seed words across languages. With as few as twenty word translations, CLTS outperforms approaches with similar and sometimes more expensive cross-lingual resources, such as parallel corpora, machine translation, or pre-trained multilingual models [Prettenhofer and Stein, 2010; Rasooli et al., 2018; Eisenschlos et al., 2019].

- **Self-training with labeling rules:** In Chapter 7, we address the problem of training text classifiers using more general labeling rules, few labeled data, and many unlabeled data. In contrast to previous work on weak supervision that ignores data that are not captured by existing labeling rules [Ratner et al., 2017], we propose weakly-supervised self-training, ASTRA, a framework that leverages all unlabeled data through a self-training mechanism that integrates human rules and a deep neural network with contextualized representations [Devlin et al., 2019]. We evaluate ASTRA across six text classification benchmarks and demonstrate its effectiveness over settings with high rule sparsity.

- **Interactive rule suggestion:** In Chapter 8, we present an interactive framework that can assist humans in teaching machines by suggesting labeling rules for weak supervision. We perform an extensive analysis of existing datasets with human-provided rules and identify prevalent patterns across datasets that could inform guidelines for rule creation. We also describe a human-in-the-loop machine teaching framework that queries a human on both instances and rules with high-level predicates that are automatically extracted without the need for large labeled datasets. We evaluate our approach across six text classification benchmarks and show that by soliciting feedback on both

instances and candidate rules, it performs better than both non-interactive methods and active learning methods with instance-level queries. To facilitate future research, we further present new benchmarks for machine teaching with with different types of interaction.

The remainder of this dissertation is organized as follows. In Chapter 2, we start with basic definitions and notation used across chapters, as well as the necessary background on supervised machine learning for NLP. Then, Chapters 3 through 8 describe our work on teaching machines with coarse labels, large taxonomies, seed words, bilingual word translations, and labeling rules. Finally, we present our conclusions in Chapter 9.

Chapter 2: Preliminaries

In this chapter, we provide the necessary definitions and notation (Section 2.1) and provide background on supervised machine learning for NLP (Section 2.2).

2.1 Mathematical Notation and Definitions

Sets: We denote with $\mathbb{N} = \{1, 2, \dots\}$ the set of natural numbers and with $\mathbb{R}$ the set of real numbers. We denote the empty set as $\emptyset$.

Sequences: We denote as $\alpha = (\alpha_1, \dots, \alpha_N)$ a sequence with N elements. The length of a sequence α is denoted as $|\alpha|$. We denote a subsequence $(\alpha_i, \dots, \alpha_j)$ as $\alpha_{i:j}$.

Tensors: We denote scalars in lowercase italics (e.g., v), vectors in lowercase boldface (e.g., $\mathbf{v}$), and tensors in uppercase boldface (e.g., $\mathbf{V}$). For $x \in \mathbb{N}$, we denote with $\mathbb{R}^x$ the set of all x-dimensional vectors of real numbers. We denote the i-th element of a vector $\mathbf{v}$ as v_i and the concatenation of two vectors $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{u} \in \mathbb{R}^n$ as $[\mathbf{v}; \mathbf{u}] \in \mathbb{R}^{m+n}$

Variable	Description
$s = (x_1, ..., x_T)$	text segment
x_j	j-th token
$\mathbf{h}$	segment embedding
$\mathbf{h}_j$	j-th token embedding
$y \in \mathcal{Y} = \{1, 2, \dots, K\}$	hard label
$\mathbf{h}_i = \text{ENC}(s_i)$	segment encoder
$\mathbf{p}_i = (p_i^1, \dots, p_i^K) = \text{CLF}(\mathbf{h}_i)$	segment classifier

Table 2.1: Notation.

Text definitions: Text is a sequence of characters. A token is a sequence of contiguous characters in a specific text segment (e.g., document, sentence, phrase). Tokenization is the process of splitting a sequence of characters into one or more tokens. A token type is the class of all tokens that have the same sequence of characters. A vocabulary is a list of all token types.

Table 2.1 summarizes the notation used throughout this dissertation. We denote a text segment s as a sequence of T ordered tokens $s = (x_1, ..., x_T)$, where x_j is the index of the j-th token type in a vocabulary V. Text segments may have a variable number of tokens.

2.2 Supervised Machine Learning for NLP

Throughout this section, we consider a simplified view of supervised machine learning that is tailored to the NLP application scenarios considered in this thesis. For a thorough background on machine learning for NLP, see [Goldberg, 2016]; for an introduction on machine learning and deep learning in general, we refer to [Murphy, 2012; LeCun et al., 2015; Goodfellow et al., 2016; Murphy, 2022].

Supervised machine learning methods for text classification use embedding techniques followed by a classification model.

Segment encoding. During segment encoding, a segment $s_i = (x_{i1}, x_{i2}, \ldots, x_{iN_i})$ composed of N_i tokens is encoded as a fixed-size real vector $\mathbf{h}_i \in \mathbb{R}^d$. We refer to the whole segment encoding procedure as:

$$\mathbf{h}_i = \text{ENC}(s_i). \tag{2.1}$$

Throughout this thesis, we use the terms "segment encoding" and "segment embedding" interchangeably to describe the procedure of segment encoding.

Defining the encoder ENC is the main goal of representation learning in NLP. There are various types of transformations used in the literature, such as bag-of-words representations [Harris, 1954; Ko, 2012], word embeddings such as word2vec [Mikolov et al., 2013b]

and GloVe [Pennington et al., 2014] (when each segment is a single word), the average of word embeddings [Wieting et al., 2015; Arora et al., 2016], Recurrent Neural Networks (RNNs) [Wieting and Gimpel, 2017; Yang et al., 2016; Bahdanau et al., 2015], or Convolutional Neural Networks (CNNs) [Kim, 2014]. For a detailed description of these architectures, we refer to [Goldberg, 2016] and [Goodfellow et al., 2016].

The parameters of ENC can optionally be initialized with pre-trained parameters that are learned using general-domain unlabeled data and self-supervised learning objectives [Mikolov et al., 2013a; Pennington et al., 2014; Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018], which we refer to as "pre-training." In this case, we refer to the encoder ENC as a pre-trained model.

One pre-trained model used commonly in this work for English documents is BERT [Devlin et al., 2019]. BERT uses a neural network architecture called transformers; see [Vaswani et al., 2017]. BERT is pre-trained on a Wikipedia dump and the Books Corpus [Zhu et al., 2015] using two objectives called "Masked Language Modeling" and "Next Sentence Prediction." We refer to [Devlin et al., 2019] for a description of the BERT architecture and training details. There are two variants of BERT, namely, the base model, with 110M parameters, and the large model, with 336M parameters. Across this thesis, we explicitly state which BERT variant we use. Another pre-trained model used in our work for multilingual settings is multilingual BERT (mBERT or MultiBERT)[1], which is a BERT variant that was pre-trained on concatenated Wikipedia data from 104 languages.

Segment classification. During segment classification, the segment s_i is assigned to one of K predefined classes $\mathcal{Y} = \{1, 2, \ldots, K\}$. To provide a probability distribution $\mathbf{p}_i = (p_i^1, \ldots, p_i^K)$ over the K classes, the segment encoding $\mathbf{h}_i$ is fed to a classification model:

$$\mathbf{p}_i = \text{CLF}(\mathbf{h}_i) \tag{2.2}$$

[1] https://github.com/google-research/bert/tree/a9ba4b8d7704c1ae18d1b28c56c0430d41407eb1

Throughout this thesis, we use the terms "segment classification," "softmax classifier," and "classification layer" interchangeably to describe the procedure of segment classification. Usually in deep learning, and in this work unless otherwise stated, the classification layer is a hidden layer followed by the softmax function: $\mathbf{p}_i = \text{softmax}(\mathbf{W}\mathbf{h}_i + \mathbf{b})$, where $\mathbf{W} \in \mathbb{R}^{K \times d}$ is the weight matrix, $b \in \mathbb{R}^K$ is the weight bias vector of the classifier, and $\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ is the softmax function for multi-class classification.

We denote the machine learning model as p_θ, which consists of the segment embedding and classification layer and predicts probabilities for a segment s_i as:

$$\mathbf{p}_i = p_\theta(y \mid s_i) = \text{CLF}(\text{ENC}(s_i)), \tag{2.3}$$

where θ is the set of all trainable parameters corresponding to the embedding and classification layer.

The dominant machine teaching paradigm. To teach a machine learning model to solve a task, the dominant teaching paradigm requires the creation of a dataset with ground truth labeled segments: $D_L = (s_i, y_i)_{i=1}^N$.

Within this supervised learning setting, optimization techniques are employed to learn the parameters of the machine learning model using the labeled data D_L as supervision. The training objective is formulated as a loss function $\mathcal{L}$ to be minimized:

$$\mathcal{L} = \sum_{(s_i, y_i) \in D_L} \mathcal{L}(\theta; s_i, y_i). \tag{2.4}$$

A common loss function used in this scenario is the cross-entropy loss:

$$\mathcal{L}(\theta; s_i, y_i) = -\log p_\theta(y \mid s_i)_{y_i}. \tag{2.5}$$

In the case where the model p_θ is a neural network and we are given an initial set of mode

parameters θ, labeled data D_L, and a loss function $\mathcal{L}$, the most common approach to train the model is via gradient descent; see [Goodfellow et al., 2016]. In cases where ENC is a pre-trained model, we interchangeably use "training" and "fine-tuning" to denote the training process for the target classification task.

As its title suggests, the main focus of this thesis is to develop alternative, resource-efficient machine teaching frameworks to address the need of large ground-truth labeled datasets D_L via alternative types of supervision. In the next chapter, we present a method for training fine-grained segment classifiers with coarse-grained labels.

Chapter 3: Fine-Grained Classification with Coarse-Grained Supervision

In this chapter, we show how to train neural networks for segment classification using coarse labels, which is one type of supervision among all types discussed in Chapter 1. First, we provide an overview and motivation for fine-grained segment classification with coarse labels (Section 3.1). Second, we provide the necessary background and define our problem of focus (Section 3.2). Third, we define a class of non-hierarchical baselines to address our problem (Section 3.3) and present our Hierarchical Sigmoid Attention Network, or HSAN (Section 3.4). Then, we present our experimental evaluation across several benchmarks (Sections 3.5 and 3.6) and describe the deployment of HSAN for health departments (Section 3.7). Finally, we summarize the contributions of this chapter (Section 3.8).

3.1 Overview and Motivation

Many applications of text review classification, such as sentiment analysis, can benefit from a fine-grained understanding of the reviews. Consider the Yelp restaurant review in Figure 3.1. Some segments (here sentences or clauses) of the review express positive sentiment towards some of the items consumed, service, and ambience, but other segments express a negative sentiment towards the price and food. To capture the nuances expressed in such reviews, analyzing the reviews at the segment level is desirable.

We focus on segment classification when only review labels — but not segment labels — are available. The lack of segment labels prevents the use of supervised learning approaches. While review labels, such as user-provided ratings, are often available, they are not directly relevant for segment classification, thus presenting a challenge for supervised learning.

Figure 3.1: A Yelp review discussing both positive and negative aspects of a restaurant, as well as food poisoning.

Existing weakly supervised learning frameworks have been proposed for training models such as support vector machines [Andrews et al., 2003; Yessenalina et al., 2010; Gärtner et al., 2002], logistic regression [Kotzias et al., 2015], and hidden conditional random fields [Täckström and McDonald, 2011]. The most recent state-of-the-art approaches employ the Multiple Instance Learning (MIL) framework in hierarchical neural networks [Pappas and Popescu-Belis, 2014; Kotzias et al., 2015; Angelidis and Lapata, 2018a; Pappas and Popescu-Belis, 2017; Ilse et al., 2018]. MIL-based hierarchical networks combine the (unknown) segment labels through an aggregation function to form a single review label. This enables the use of ground-truth review labels as a weak form of supervision for training segment-level classifiers. However, it remains unanswered whether performance gains in current models stem from the hierarchical structure of the models or from the representational power of their deep learning components. Also, as we will see, the current modeling choices for the MIL aggregation function might be problematic for some applications and, in turn, might hurt the performance of the resulting classifiers.

Our work presents the following contributions:

1. We show that non-hierarchical, deep learning approaches for segment-level sentiment

classification — with only review-level labels — are strong, and they equal or exceed in performance hierarchical networks with various MIL aggregation functions.

2. We substantially improve previous hierarchical approaches for segment-level sentiment classification and propose the use of a new MIL aggregation function based on the sigmoid attention mechanism to jointly model the relative importance of each segment as a product of Bernoulli distributions. This modeling choice allows multiple segments to contribute with different weights to the review label, which is desirable in many applications, including segment-level sentiment classification.

3. We experiment beyond sentiment classification and apply our approach to the discovery of foodborne illness incidents in online restaurant reviews. We experimentally show that our MIL-based network effectively detects segments discussing food poisoning and has a higher chance than all previous models to identify unknown foodborne outbreaks. By identifying which review segments discuss food poisoning, epidemiologists can focus on the relevant portions of the review and safely ignore the rest.

We start with a review of the relevant background for multiple-instance learning and define our problem of focus (see Section 3.2). We continue as follows:

- We explore non-hierarchical baselines (Section 3.3).
- We develop HSAN, a neural network that uses the sigmoid attention mechanism to classify segments using review labels only as supervision (Section 3.4).
- We evaluate our ideas by conducting an experimental evaluation on sentiment classification (Sections 3.5 and 3.6).
- We evaluate our approach for foodborne illness detection and demonstrate its deployment for health departments (Section 3.7).

Finally, we discuss the implications of our work (Section 3.8). The material described in this chapter appears in [Karamanolakis et al., 2019c].

3.2 Background and Problem Definition

In this section, we summarize relevant work on weakly supervised models for segment classification (Section 3.2.1) and define our problem of focus (Section 3.2.2).

3.2.1 Multiple Instance Learning for Classification with Coarse Labels.

As discussed in Section 2.2, supervised approaches first use a segment encoder ENC to encode a segment s into a vector $\mathbf{h}_i = \text{ENC}(s_i)$ and then use a segment classifier CLF to classify $\mathbf{h}_i$ to one of C predefined classes $[\mathcal{Y}] := \{1, 2, \ldots, K\}$: $\mathbf{p}_i = \text{CLF}(\mathbf{h}_i)$. In contrast to traditional supervised learning, where *segment labels* are required to train segment classifiers, MIL-based models can be trained using *review labels* as a weak source of supervision, as we describe next.

State-of-the-art weakly supervised approaches for segment and review classification employ the Multiple Instance Learning (MIL) framework [Zhou et al., 2009; Pappas and Popescu-Belis, 2014; Kotzias et al., 2015; Pappas and Popescu-Belis, 2017; Angelidis and Lapata, 2018a]. MIL is employed for problems where data are arranged in groups (bags) of instances. In our setting, each review is a group of segments: $r = (s_1, s_2, \ldots, s_M)$. The key assumption followed by MIL is that the observed review label is an aggregation function of the unobserved segment labels: $p = \text{AGG}(\mathbf{p}_1, \ldots, \mathbf{p}_M)$. Hierarchical MIL-based models (Figure 3.2) work in three main steps: (1) encode the review segments into fixed-size vectors $\mathbf{h}_i = \text{ENC}(s_i)$, (2) provide segment predictions $\mathbf{p}_i = \text{CLF}(\mathbf{h}_i)$, and (3) aggregate the predictions to get a review-level probability estimate $\mathbf{p} = \text{AGG}(\mathbf{p}_1, \ldots, \mathbf{p}_M)$. Supervision during training is provided in the form of review labels.

Different modeling choices have been taken for each part of the MIL hierarchical architecture. [Kotzias et al., 2015] encoded sentences as the internal representations of a hierarchical CNN that was pre-trained for document-level sentiment classification [Denil et al., 2014], and used the uniform average for the aggregation function. [Pappas and Popescu-Belis, 2014;

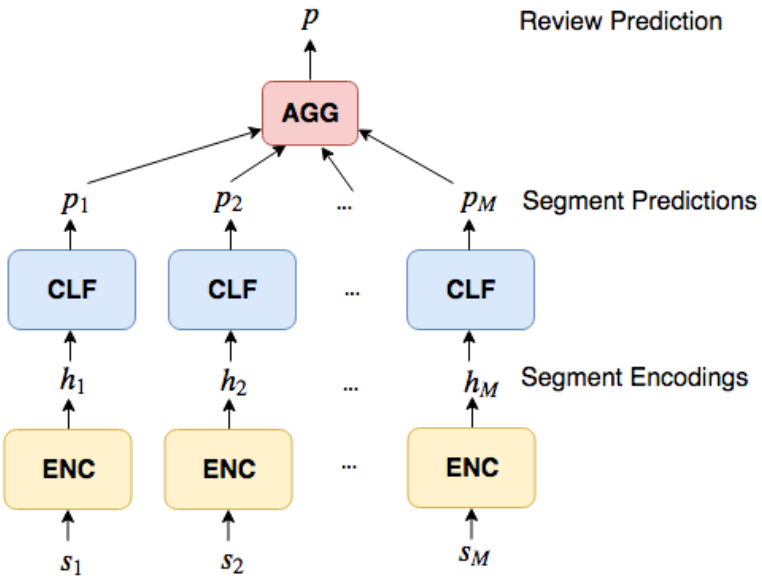


Figure 3.2: MIL-based hierarchical models.

Pappas and Popescu-Belis, 2017] employed Multiple Instance Regression, evaluated various models for segment encoding, including feed forward neural networks and Gated Recurrent Units (GRUs) [Bahdanau et al., 2015], and used the weighted average for the aggregation function, where the weights were computed by linear regression or a one-layer neural network. [Angelidis and Lapata, 2018a] proposed an end-to-end Multiple Instance Learning Network (MILNET), which outperformed previous models for sentiment classification using CNNs for segment encoding, a softmax layer for segment classification, and GRUs with attention [Bahdanau et al., 2015] to aggregate segment predictions as a weighted average. Our proposed model (Section 3.4) also follows the MIL hierarchical structure of Figure 3.2 for both sentiment classification and our important public health application (Section 3.7).

3.2.2 Problem Definition

Consider a text review for an entity, with M contiguous segments $r = (s_1, \ldots, s_M)$. Segments may have a variable number of words and different reviews may have a different number of segments. A discrete label $y_r \in [K]$ is provided for each review but the individual

segment labels are not provided. Our goal is to train a segment-level classifier that, given an unseen test review $r^t = (s_1^t, s_2^t, \ldots, s_{M_t}^t)$, predicts a label $\mathbf{p}_i$ for each segment and then aggregates the segment labels to infer the review label $y_r^t \in [K]$ for r^t.

3.3 Non-Hierarchical Baselines

We can address the problem described in Section 3.2.2 without using hierarchical approaches such as MIL. In fact, the hierarchical structure of Figure 3.2 for the MIL-based deep networks adds a level of complexity that has not been empirically justified, giving rise to the following question: do performance gains in current MIL-based models stem from their hierarchical structure or just from the representational power of their deep learning components?

We explore this question by evaluating a class of simpler non-hierarchical baselines: deep neural networks trained at the *review level* (without encoding and classifying individual segments) and applied at the *segment level* by treating each test segment as if it were a short "review." While the distribution of input length is different during training and testing, we will show that this class of non-hierarchical models is quite competitive and sometime outperforms MIL-based networks with inappropriate modeling choices.

3.4 Hierarchical Sigmoid Attention Network (HSAN)

We now describe the details of our MIL-based hierarchical approach, which we call Hierarchical Sigmoid Attention Network (HSAN). HSAN works in three steps to process a review, following the general architecture in Figure 3.2: (1) each segment s_i in the review is encoded as a fixed-size vector using word embeddings and CNNs [Kim, 2014]: $\mathbf{h}_i = \text{CNN}(s_i) \in \mathbb{R}^\ell$; (2) each segment encoding $\mathbf{h}_i$ is classified using a softmax classifier with parameters $W \in \mathbb{R}^\ell$ and $b \in \mathbb{R}$: $\mathbf{p}_i = \text{softmax}(W\mathbf{h}_i + b)$; and (3) a review prediction $\mathbf{p}$ is computed as an aggregation function of the segment predictions $\mathbf{p}_1, \ldots, \mathbf{p}_M$ from the previous step. A key contribution of our work is the motivation, definition, and evaluation of a suitable aggregation function

for HSAN, a critical design issue for MIL-based models.

The choice of aggregation function has a substantial impact on the performance of MIL-based models and should depend on the specific assumptions about the relationship between bags and instances [Carbonneau et al., 2018]. Importantly, the performance of MIL algorithms depends on the witness rate (WR), which is defined as the proportion of positive instances in positive bags. For example, when WR is very low (which is the case in our public health application of Section 3.7), using the uniform average as an aggregation function in MIL is not an appropriate modeling choice, because the contribution of the few positive instances to the bag label is outweighed by that of the negative instances.

The choice of the uniform average of segment predictions [Kotzias et al., 2015] is also problematic because particular segments of reviews might be more informative than other segments for the task at hand and thus should contribute with higher weights to the computation of the review label. For this reason, we opt for the weighted average [Pappas and Popescu-Belis, 2014; Angelidis and Lapata, 2018a]:

$$\mathbf{p} = \frac{\sum_{i=1}^{M} \alpha_i \cdot \mathbf{p}_i}{\sum_{i=1}^{M} \alpha_i}. \tag{3.1}$$

The weights $\alpha_i \in [0,1]$ define the relative contribution of the corresponding segments s_i to the review label. To estimate the segment weights, we adopt the attention mechanism [Bahdanau et al., 2015]. In contrast to MILNET [Angelidis and Lapata, 2018a], which uses the traditional softmax attention, we propose to use the sigmoid attention. Sigmoid attention is both functionally and semantically different from softmax attention and is more suitable for our problem, as we show next.

The probabilistic interpretation of softmax attention is that of a categorical latent variable $z \in \{1, \ldots, M\}$ that represents the index of the segment to be selected from the M

segments [Kim et al., 2017]. The attention probability distribution is:

$$p(z = i \mid (e_1, \ldots, e_M)) = \frac{\exp(e_i)}{\sum_{i=1}^{M} \exp(e_i)}, \tag{3.2}$$

where:

$$e_i = \mathbf{u}_a^T \tanh(\mathbf{W}_a \mathbf{h}'_i + \mathbf{b}_a), \tag{3.3}$$

where $\mathbf{h}'_i$ are context-dependent segment vectors computed using bi-directional GRUs (Bi-GRUs), $\mathbf{W}_a \in \mathbb{R}^{m \times n}$ and $\mathbf{b}_a \in \mathbb{R}^n$ are the attention model's weight and bias parameter, respectively, and $\mathbf{u}_a \in \mathbb{R}^m$ is the "attention query" vector parameter. The probabilistic interpretation of Equation 3.2 suggests that, when using the softmax attention, exactly one segment should be considered important under the constraint that the weights of all segments sum to one. This property of the softmax attention to prioritize one instance explains the successful application of the mechanism for problems such as machine translation [Bahdanau et al., 2015], where the role of attention is to align each target word to (usually) one of the M words from the source language. However, softmax attention is not well suited for estimating the aggregation function weights for our problem, where multiple segments usually affect the review-level prediction.

We hence propose using the sigmoid attention mechanism to compute the weights $\alpha_1, \ldots, \alpha_M$. In particular, we replace softmax in Equation (3.2) with the sigmoid (logistic) function:

$$\alpha_i = \sigma(e_i) = \frac{1}{1 + \exp(-e_i)}. \tag{3.4}$$

With sigmoid attention, the computation of the attention weight α_i does not depend on scores e_j for $j \neq i$. Indeed, the probabilistic interpretation of sigmoid attention is a vector z of discrete latent variables $z = (z_1, \ldots, z_M)$, where $z_i \in \{0, 1\}$ [Kim et al., 2017]. In other words, the relative importance of each segment is modeled as a Bernoulli distribution. The

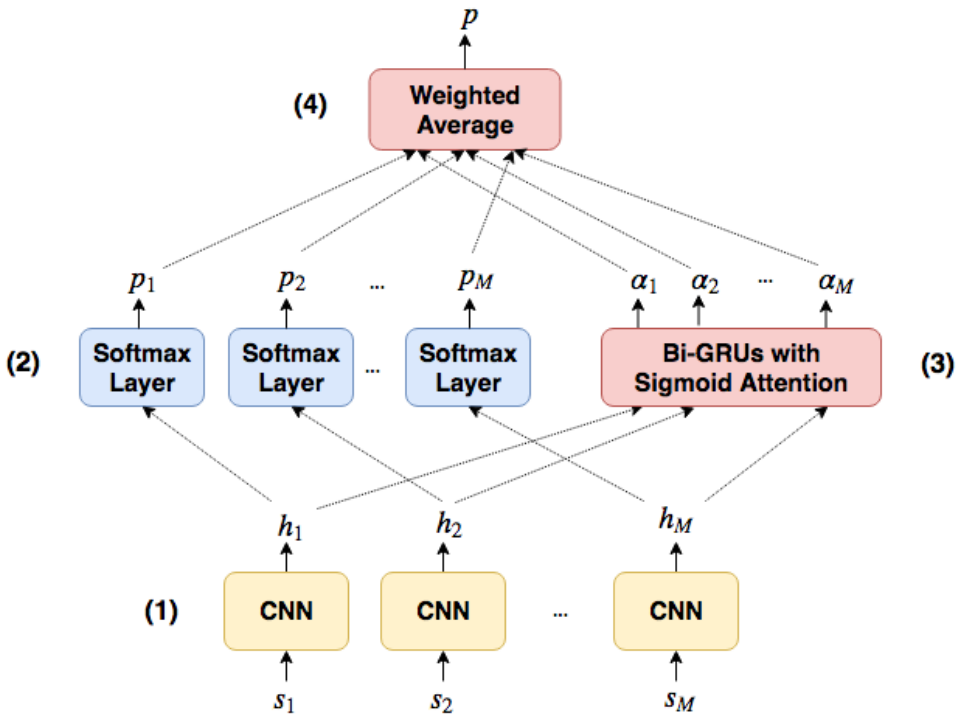


Figure 3.3: Our Hierarchical Sigmoid Attention Network.

sigmoid attention probability distribution is:

$$p(z_i = 1 \mid (e_1, \ldots, e_M)) = \sigma(e_i). \tag{3.5}$$

This probabilistic model indicates that $(z_1, \ldots, z_M)$ are conditionally independent given $(e_1, \ldots, e_M)$. Therefore, sigmoid attention allows multiple segments, or even no segments, to be selected. This property of sigmoid attention explains why it is more appropriate for our problem. Also, as we will see in the next sections, using the sigmoid attention is the key modeling change needed in MIL-based hierarchical networks to outperform non-hierarchical baselines for segment-level classification. Attention mechanisms using sigmoid activation have also been recently applied for tasks different than segment-level classification of reviews [Shen and Lee, 2016; Kim et al., 2017; Rei and Søgaard, 2018]. Our work differs from these approaches in that we use the sigmoid attention mechanism for the MIL aggregation

function of Equation 3.1, i.e., we aggregate segment labels $\mathbf{p}_i$ (instead of segment vectors $\mathbf{h}_i$) into a single review label $\mathbf{p}$ (instead of review vectors $\mathbf{h}$).

We summarize our HSAN architecture in Figure 4.2. HSAN follows the MIL framework and thus it does not require segment labels for training. Instead, we only use ground-truth review labels and jointly learn the model parameters by minimizing the negative log-likelihood of the model parameters. Even though a single label is available for each review, our model allows different segments of the review to receive different labels. Thus, we can appropriately handle reviews such as that in Figure 3.1 and assign a mix of positive and negative segment labels, even when the review as a whole has a negative (2-star) rating.

We now turn to another key contribution of our work, namely, the evaluation of critical aspects of hierarchical approaches and also our HSAN approach. For this, we focus on two important and fundamentally different, real-world applications: segment-level sentiment classification and the discovery of foodborne illness in restaurant reviews. First, we describe the experimental setting and results for sentiment classification (Sections 3.5 and 3.6, respectively), and then we discuss our foodborne illness detection results and deployment of HSAN for health departments (Section 3.7).

3.5 Experimental Settings

For segment-level sentiment classification, we use the Yelp'13 and IMDB corpora [Diao et al., 2014]. The Yelp'13 corpus [Tang et al., 2015] contains 335,018 user reviews of local businesses. Each review includes a 5-star rating ranging from 1 (negative) to 5 stars (positive). The IMDB corpus [Diao et al., 2014] contains 348,415 movie reviews with ratings ranging from 1 (negative) to 10 stars (positive). For both corpora, training (80%), validation (10%), and test (10%) sets are provided.

We do not use segment labels for training any models except the fully supervised Seg-* baselines (see below). For evaluating the segment-level classification performance on Yelp'13 and IMDB, we use the SPOT-Yelp and SPOT-IMDB datasets, respectively [Angelidis and

Statistic	SPOT-Yelp SENT	SPOT-Yelp EDU	SPOT-IMDB SENT	SPOT-IMDB EDU
# Segments	1,065	2,110	1,029	2,398
Positive segments (%)	39.9	32.9	37.9	25.6
Neutral segments (%)	21.7	34.3	29.2	47.7
Negative segments (%)	38.4	32.8	32.9	26.7
Witness positive (# segs)	7.9	12.1	6.0	8.5
Witness negative (# segs)	7.3	11.6	6.6	11.2
Witness salient (# segs)	8.5	14.0	7.6	12.6
WR positive	0.74	0.58	0.55	0.36
WR negative	0.68	0.53	0.63	0.43
WR salient	0.80	0.65	0.76	0.55

Table 3.1: Label statistics for the SPOT datasets. "WR (x)" is the witness rate, meaning the proportion of segments with label x in a review with label x. "Witness (x)" is the average number of segments with label x in a review with label x. "Salient" is the union of the "positive" and "negative" classes.

Lapata, 2018a]. Each dataset has been segmented both at sentences (SPOT-*-SENT) and EDUs (SPOT-*-EDU).[1]. The test sets have 3 labels (Table 1): "negative," "neutral," and "positive." These datasets contain 100 Yelp reviews and 97 IMDB reviews from the Yelp'13 and IMDB test sets, respectively.

For a robust evaluation of our approach (HSAN), we compare against state-of-the-art models and baselines:

- **Rev-*:** non-hierarchical models, trained at the review level and applied at the segment level (see Section 3.3); this family includes a logistic regression classifier trained on review embeddings, computed as the element-wise average of word embeddings ("Rev-LR-EMB"), a CNN ("Rev-CNN") [Kim, 2014], and a Bi-GRU with attention ("Rev-RNN") [Bahdanau et al., 2015]. Rev-LR-BoW encodes the review text as a bag-of-words vector including n-grams (for n=1, 2, and 3) and each term is weighted using the Term Frequency-Inverse Document Frequency (TF-IDF) statistic [Leskovec et al., 2014].

[1]The use of EDUs for sentiment classification is motivated in [Angelidis and Lapata, 2018a].

- **MIL-*:** MIL-based hierarchical deep learning models with different aggregation functions. "MIL-avg" computes the review label as the average of the segment-level predictions [Kotzias et al., 2015]. "MIL-softmax" uses the softmax attention mechanism –this is the best performing MILNET model reported in [Angelidis and Lapata, 2018a] ("MILNETgt"). "MIL-sigmoid" uses the sigmoid attention mechanism as we propose in Section 3.4 (HSAN model). All MIL-* models have the hierarchical structure of Figure 3.2 and for comparison reasons we use the same functions for segment encoding (ENC) and segment classification (CLF), namely, a CNN and a softmax classifier, respectively. For a fair comparison, all the MIL-* models have the same parameter configuration as MILNET (Section 5.3 in [Angelidis and Lapata, 2018a]).

For the evaluation of hierarchical non-MIL networks such as the hierarchical classifier of [Yang et al., 2016], see [Angelidis and Lapata, 2018a]. Here, we ignore this class of models as they have been outperformed by MILNET.

For all models using word embeddings (i.e., Seg-*, Rev-*, MIL-*), we initialize the word embeddings using 300-dimensional ($k = 300$) pre-trained word2vec embeddings [Mikolov et al., 2013b]. For the CNNs we use kernels of size 3, 4, and 5 words, 100 feature maps per kernel, stride of size 1, and max-over-time pooling to get fixed-size segment encodings (resulting in $\ell = 300$). For the forward and backward GRUs we use hidden vectors with 50 dimensions ($n = 2 \cdot 50 = 100$), while for the attention mechanism we use vectors of 100 dimensions ($m = 100$). We use dropout (with rate 0.5) on the word embeddings and the internal GRU states. We use L2 regularization for the softmax classifier.

The above models require only review-level labels for training, which is the scenario of focus of this work. For comparison purposes, we also evaluate a family of fully supervised baselines trained at the *segment* level:

- **Seg-*:** fully supervised baselines using SPOT segment labels for training. "Seg-LR" is a logistic regression classifier trained on segment embeddings, which are computed as the element-wise average of the corresponding word embeddings. We also report the

CNN baseline ("Seg-CNN"), which was evaluated in [Angelidis and Lapata, 2018a]. Seg-* baselines are evaluated using 10-fold cross-validation on the SPOT dataset.

We evaluate all approaches using the macro-averaged F1 score.

3.6 Experimental Results for Sentiment Classification

This section describes our experimental results for fine-grained sentiment classification. Table 3.2 reports the evaluation results on SPOT datasets for both sentence- and EDU-level classification.

The Seg-* baselines are not directly comparable with other models, as they are trained at the segment level on the (relatively small) SPOT datasets with segment labels. The more complex Seg-CNN model does not significantly improve over the simpler Seg-LR, perhaps due to the small training set available at the segment level.

Rev-CNN outperforms Seg-CNN in three out of the four datasets. Although Rev-CNN is trained at the review level (but is applied at the segment level), it is trained with 10 times as many examples as Seg-CNN. This suggests that, for the non-hierarchical CNN models, review-level training may be advantageous with more training examples. In addition, Rev-CNN outperforms Rev-LR-EMB, indicating that the fine-tuned features extracted by the CNN are an improvement over the pre-trained embeddings used by Rev-LR-EMB.

Rev-CNN outperforms MIL-avg and has comparable performance to MILNET: non-hierarchical deep learning models trained at the review level and applied at the segment level are strong baselines, because of their representational power. Thus, the Rev-* model class should be evaluated and compared with MIL-based hierarchical models for applications where segment labels are not available.

Interestingly, MIL-sigmoid (HSAN) consistently outperforms all models, including MIL-avg, MIL-softmax (MILNET), and the Rev-* baselines. This shows that:

1. the choice of aggregation function of MIL-based classifiers heavily impacts classification performance; and

Method	SPOT-Yelp SENT	SPOT-Yelp EDU	SPOT-IMDB SENT	SPOT-IMDB EDU
Seg-LR	55.6	59.2	60.5	62.8
Seg-CNN	56.2	60.0	58.3	63.0
Rev-LR-EMB	51.2	49.3	52.7	48.6
Rev-CNN	60.6	61.5	60.8	60.1
Rev-RNN	58.5	53.9	55.3	50.8
MIL-avg	51.8	46.8	45.7	38.4
MIL-softmax	63.4	59.9	64.0	59.9
MIL-sigmoid	**64.6**	**63.3**	**66.2**	**65.7**

Table 3.2: F1 score for segment-level sentiment classification.

2. MIL-based hierarchical networks can indeed outperform non-hierarchical networks when the appropriate aggregation function is used.

We emphasize that we use the same ENC and CLF functions across all MIL-based models to show that performance gains stem solely from the choice of aggregation function. Given that HSAN consistently outperforms MILNET in all datasets for segment-level sentiment classification, we conclude that the choice of sigmoid attention for aggregation is a better fit than softmax for this task.

The difference in performance between HSAN and MILNET is especially pronounced on the *-EDU datasets. We explain this behavior with the statistics of Table 3.1: "Witness (Salient)" is higher in *-EDU datasets compared to *-SENT datasets. In other words, *-EDU datasets contain more segments that should be considered important than *-SENT datasets. This implies that the attention model needs to "attend" to more segments in the case of *-EDU datasets: as we argued in Section 3.4, this is best modeled by sigmoid attention.

3.7 Deployment of HSAN for Health Departments

This section describes the application of HSAN for the discovery of foodborne illness in restaurant reviews, leading to the deployment of HSAN to help daily inspections by epidemiologists in health departments. First, we describe our public health application and

then we present our experimental setting and results.

Foodborne illness discovery in online restaurant reviews. Health departments nationwide have started to analyze social media content (e.g., Yelp reviews, Twitter messages) to identify foodborne illness outbreaks originating in restaurants. In Chicago [Harris et al., 2014], New York City [Effland et al., 2018], Nevada [Sadilek et al., 2016], and St. Louis [Harris et al., 2018], text classification systems have been successfully deployed for the detection of social media documents mentioning foodborne illness. (Figure 3.1 shows a Yelp review discussing a food poisoning incident.) After such social media documents are flagged by the classifiers, they are typically examined manually by epidemiologists, who decide if further investigation (e.g., interviewing the restaurant patrons who became ill, inspecting the restaurant) is warranted. This manual examination is time-consuming, and hence it is critically important to (1) produce accurate review-level classifiers, to identify foodborne illness cases while not showing epidemiologists large numbers of false-positive cases; and (2) annotate the flagged reviews to help the epidemiologists in their decision-making.

We propose to apply our segment classification approach to this important public health application. By identifying which review segments discuss food poisoning, epidemiologists can focus on the relevant portions of the review and safely ignore the rest. As we will see, our evaluation will focus on Yelp restaurant reviews. Discovering foodborne illness is fundamentally different from sentiment classification, because the mentions of food poisoning incidents in Yelp are rare. Furthermore, even reviews mentioning foodborne illness often include multiple sentences unrelated to foodborne illness (see Figure 3.1).

Experimental setting. For the discovery of foodborne illness, we use a dataset of Yelp restaurant reviews, manually labeled by epidemiologists in the New York City Department of Health and Mental Hygiene. This is the same training and test sets as in [Effland et al., 2018]. Each review is assigned a binary label ("Sick" vs. "Not Sick"). The review-level training set ("Silver" set in [Effland et al., 2018]) contains 21,551 (5,895 "Sick," 15,656 "Not

Sick") reviews posted before January 1, 2017. The review-level test set contains 2,975 (949 "Sick," 2,026 "Not Sick") reviews posted after January 1, 2017. Sample weights are also calculated to account for the selection bias in this dataset [Effland et al., 2018]. We split the review-level training set into training (90%) and validation (10%) sets, randomly stratified by label and sample weight. We do not use any sentence-level labels for training.

We fine-tune the model parameters on the validation set with respect to the F1 score. Given a test review, we predict a label for each sentence and aggregate the sentence predictions to get a single review prediction. For review-level classification, we use the review prediction, while for sentence-level evaluation we use the individual sentence predictions. The segment-level confidence scores are computed by multiplying the segment probability for the "Sick" class with its attention weight.

To test the models at the sentence level, epidemiologists have manually annotated each sentence for 437 out of the 949 "Sick" test reviews. Given a review for labeling, epidemiologists read the whole review text and decided on the label for each sentence. This led to 3,114 labeled sentences (630 "Sick," 2,484 "Not Sick"). In this sentence-level dataset, the WR of the "Sick" class is 0.25, which is significantly lower than the WR on sentiment classification datasets (Table 3.1). In other words, the proportion of "Sick" segments in "Sick" reviews is relatively low; in contrast, in sentiment classification the proportion of positive (or negative) segments is relatively high in positive (or negative) reviews.

We use the same baselines as for sentiment classification (Section 3.5) and additionally report a logistic regression classifier trained on bag-of-words review vectors ("Rev-LR-BoW"), because it is the best performing model in previous work [Effland et al., 2018].

For review-level foodborne classification, we account for the selection bias in the review-level test set by computing precision and recall using sample weights [Effland et al., 2018]. Because of the class imbalance at both the review and sentence levels, we report precision, recall, F1 score, and area under the precision-recall curve (AUPR). Also, we follow [Effland et al., 2018] and estimate 95% confidence intervals (95% CI) for the F1 and AUPR metrics

	Review-Level Evaluation				Sentence-Level Evaluation				
Model	**Prec**	**Rec**	**F1 (95% CI)**	**AUPR (95% CI)**	**Acc**	**Prec**	**Rec**	**F1**	**AUPR**
Rev-LR-BoW	85.3	88.2	86.7 (85.2, 88.2)	91.4 (90.0, 92.9)	89.1	**82.1**	58.8	68.5	80.9
Rev-LR-EMB	70.4	57.4	63.3 (51.3, 71.4)	69.6 (64.9, 75.5)	79.7	50.0	84.3	62.8	48.9
Rev-CNN	80.3	89.8	84.8 (83.2, 86.6)	93.5 (92.3, 94.6)	88.7	79.3	59.4	67.9	24.7
Rev-RNN	85.6	87.8	86.7 (84.9, 88.4)	92.9 (91.5, 94.2)	91.3	81.0	74.5	77.6	11.3
MIL-avg	67.4	53.7	59.8 (48.5, 68.2)	64.3 (59.6, 70.8)	90.3	75.0	78.0	76.5	73.6
MIL-softmax	82.9	92.8	87.6 (85.9, 89.0)	**94.1** (92.6, 99.4)	91.2	75.5	83.3	79.2	81.6
MIL-sigmoid	**86.5**	**92.9**	**89.6** (88.2, 91.0)	91.3 (88.7, 92.6)	**92.0**	76.4	**87.4**	**81.5**	**84.0**

Table 3.3: Review-level (left) and sentence-level (right) evaluation results for discovering foodborne illness in Yelp reviews.

using the percentile bootstrap method [Efron and Tibshirani, 1994] with sampled test sets of 1,000 reviews. For sentence-level foodborne classification, we also report the accuracy score.

Experimental results. Table 3.3 reports the evaluation results for both review- and sentence-level foodborne classification.[2] Rev-LR-EMB has significantly lower F1 score than Rev-CNN and Rev-RNN: representing a review as the uniform average of the word embeddings is not an appropriate modeling choice for this task, where only a few segments in each review are relevant to the positive class.

MIL-sigmoid (HSAN) achieves the highest F1 score among all models for review-level classification. MIL-avg has lower F1 score compared to other models: as discussed in Section 3.2.1, in applications where the value of WR is very low (here WR=0.25), the uniform average is not an appropriate aggregation function for MIL.

Applying the best classifier reported in [Effland et al., 2018] (Rev-LR-BoW) for sentence-level classification leads to high precision but very low recall. On the other hand, the MIL-* models outperform the Rev-* models in F1 score (with the exception of MIL-avg, which has lower F1 score than Rev-RNN): the MIL framework is appropriate for this task, especially when the weighted average is used for the aggregation function. The significant difference in recall and F1 score between different MIL-based models highlights once again

[2]We report review-level classification results because epidemiologists rely on the *review-level* predictions to decide whether to investigate restaurants; in turn, *segment-level* predictions help epidemiologists focus on the relevant portions of positively labeled reviews.

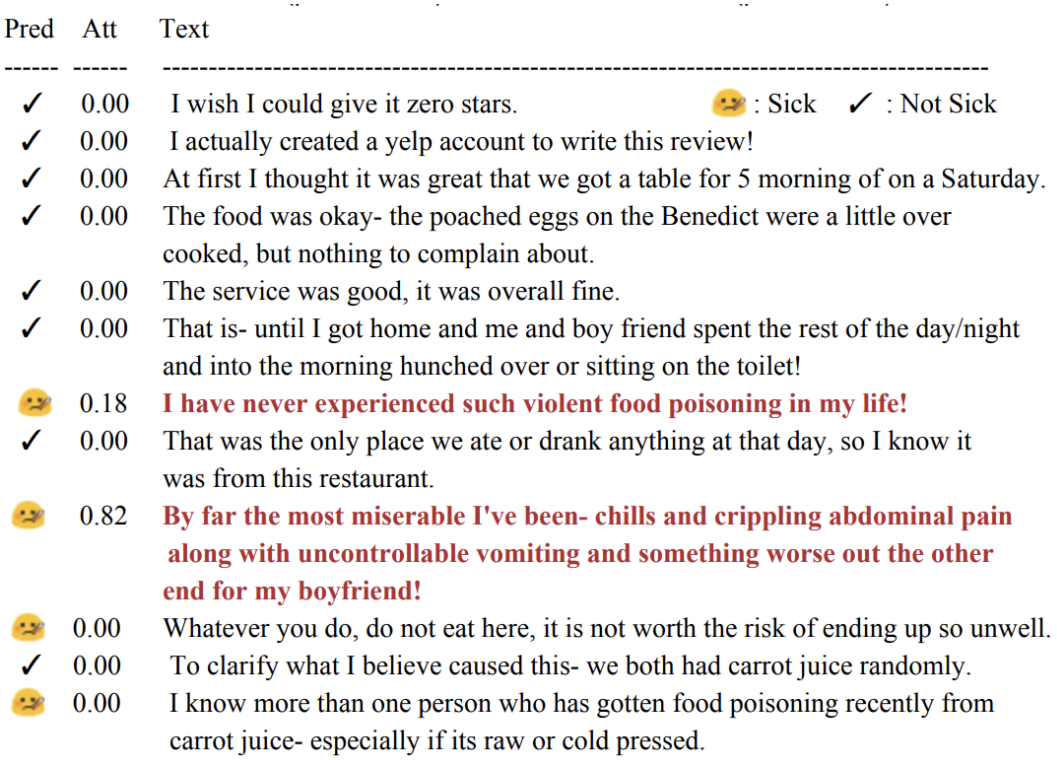


Figure 3.4: HSAN's fine-grained predictions for a Yelp review: for each sentence, HSAN provides one binary label (Pred) and one attention score (Att). A sentence is highlighted if its attention score is greater than 0.1.

the importance of choosing the appropriate aggregation function. MIL-sigmoid consistently outperforms MIL-softmax in all metrics, showing that the sigmoid attention properly encodes the hierarchical structure of reviews. MIL-sigmoid also outperforms all other models in all metrics. Also, MIL-sigmoid's recall is 48.6% higher than that of Rev-LR-BoW. In other words, MIL-sigmoid detects more sentences relevant to foodborne illness than Rev-LR-BoW, which is especially desirable for this application, as discussed next.

Fine-grained predictions could potentially help epidemiologists to quickly focus on the relevant portions of the reviews and safely ignore the rest. Figure 3.4 shows how the segment predictions and attention scores predicted by HSAN — with the highest recall and F1 score among all models that we evaluated — could be used to highlight important sentences of a review. We highlight sentences in red if the corresponding attention scores exceed a pre-defined threshold. In this example, high attention scores are assigned by HSAN to sentences

that mention food poisoning or symptoms related to food poisoning. This is particularly important because reviews on Yelp and other platforms can be long, with many irrelevant sentences surrounding the truly important ones for the task at hand.

To help epidemiologists, we have deployed HSAN for health departments in New York City and Los Angeles County. HSAN provides fine-grained predictions for Yelp restaurant reviews for daily inspection. We have also created a graphical user interface[3] for the inspection of candidate reviews in an interactive map of restaurants. In the main page that shows multiple reviews, we display only the sentences that are highlighted by HSAN and give the option of reading the text of the whole review. Such an interface allows epidemiologists to examine reviews more efficiently and, ultimately, more effectively.

3.8 Conclusions

In this chapter, we presented a Multiple Instance Learning-based model for fine-grained text classification that requires only review-level labels for training but produces both review- and segment-level labels. We summarize the contributions of this chapter as follows: (i) we explored non-hierarchical baselines trained at the review level and applied at the segment level by treating each test segment as if it were a short "review" (Section 3.3); (ii) we developed HSAN, a neural network with a new MIL aggregation function based on the sigmoid attention mechanism, which explicitly allows multiple segments to contribute to the review-level classification decision with different weights (Section 3.4); (iii) we evaluated our ideas by conducting an experimental evaluation on sentiment classification (Sections 3.5 and 3.6); and (iv) we applied our weakly supervised approach to the important public health application of foodborne illness discovery in online restaurant reviews and demonstrated its deployment for health departments (Section 3.7).

Our findings show that our non-hierarchical baselines are surprisingly strong and perform comparably or better than MIL-based hierarchical networks with a variety of aggregation

[3] `https://github.com/cu-publichealth/FoodborneML/tree/master/foodborne-viz`

functions. By fixing all components except the MIL aggregation function, we found that the sigmoid attention mechanism in HSAN is the key modeling change needed for MIL-based hierarchical networks to outperform the non-hierarchical baselines for segment-level sentiment classification. Consequently, we believe that HSAN emerges as a promising approach for MIL, especially when the witness rate (i.e., the percentage of positive instances within a bag) is low. Importantly, we showed that HSAN has a higher chance than all previous models to identify unknown foodborne outbreaks, and demonstrated how its fine-grained segment annotations can be used to highlight the segments that were considered important for the computation of the review-level label. By deploying HSAN for inspections in health departments, we provide epidemiologists a new tool to interact with machine learning models, first by using coarse labels to teach segment classifiers, and second to inspect reviews by reading the most important sentences as highlighted by HSAN.

Chapter 4: Knowledge Extraction with Hierarchical Taxonomies of Product Categories

In Chapter 3, we discussed a method for training neural networks with coarse labels and demonstrated the application of our weakly-supervised neural network, HSAN, in health departments. In this chapter, we address a different task, namely product knowledge extraction, and show how to train neural networks for thousands of product categories using a taxonomy with hierarchical category relations. First, we provide an overview and motivation for product knowledge extraction from thousands of product categories (Section 4.1). Second, we provide the necessary background and define our problem of focus (Section 4.2). Then, we present our taxonomy-aware network, or TXtract (Section 4.3). Then, we present our large-scale experimental evaluation across 4,000 product categories (Sections 4.4 and 4.5) and describe the integration of TXtract into Amazon's AutoKnow (Section 4.6). Finally, we conclude with the contributions of this chapter (Section 4.7).

4.1 Overview and Motivation

Real-world e-commerce platforms contain billions of products from thousands of different categories, organized in hierarchical taxonomies.

Consider for instance the "Ben & Jerry's" product assigned under "Ice Cream" in Figure 4.1. Knowledge about this product can be represented in structured form as a catalog of product attributes (e.g., *flavor*) and their values (e.g., "strawberry cheesecake"). Understanding precise values of product attributes is crucial for many applications including product search, recommendation, and question answering. However, structured attributes in product catalogs are often incomplete, leading to unsatisfactory search results.

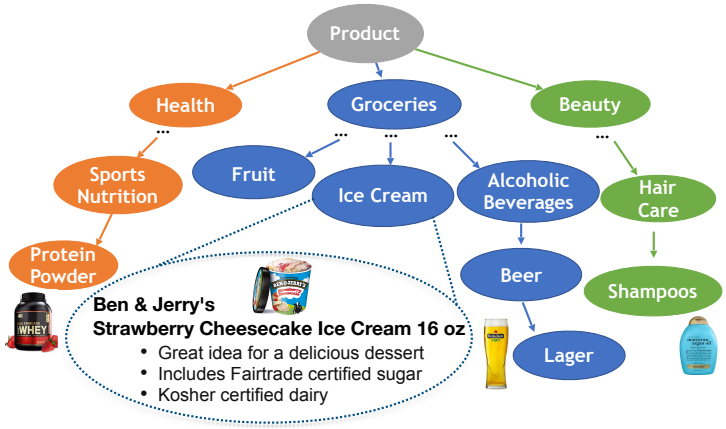


Figure 4.1: A hierarchical taxonomy with various product categories and the public webpage of a product assigned to "Ice Cream" category.

In this work, we extract such structured information from product profiles such as product titles and descriptions. In the previous example of an ice cream product, the corresponding title can potentially be used to extract values for attributes, such as "Ben & Jerry's" for *brand*, "Strawberry Cheesecake" for *flavor*, and "16 oz" for *capacity*.

State-of-the-art approaches for attribute value extraction [Zheng et al., 2018; Xu et al., 2019; Rezk et al., 2019] have employed deep learning to capture features of product attributes effectively for the extraction purpose. However, they are all designed without considering the product categories and thus cannot effectively capture the diversity of categories across the product taxonomy. Categories can be substantially different in terms of applicable attributes (e.g., a "Camera" product should not have *flavor*), attribute values (e.g., "Vitamin" products may have "fruit" *flavor* but "Banana" products should not) and more generally, text patterns used to describe the attribute values (e.g., the phrase "infused with" is commonly followed by a *scent* value such as "lavender" in "Hair Care" products but not in "Mattresses" products).

Here, we consider attribute value extraction for real-world hierarchical taxonomies with thousands of product categories, where directly applying previous approaches presents limitations. On the one extreme, ignoring the hierarchical structure of categories in the taxonomy and assuming a single "flat" space for all products does not capture category-specific characteristics and, as we show in our experiments, is not effective. On the other extreme, training

a separate deep neural network for each category in the product taxonomy is prohibitively expensive, and can suffer from lack of training data on small categories.

To address the limitations of previous approaches under this challenging setting, we present a framework for *category-specific* attribute value extraction that is both efficient and effective. Our deep neural network, TXtract, is *taxonomy-aware*: it leverages the hierarchical taxonomy of product categories and extracts attribute values for a product conditional to its category, such that TXtract automatically associates categories with specific attributes, valid attribute values, and category-specific text patterns. TXtract is trained on all categories in parallel and thus can be applied even on small categories with limited labels.

The key question we need to answer is *how to condition deep sequence models on product categories*. Our experiments suggest that following previous work to append category-specific artificial tokens to the input sequence, or concatenate category embeddings to hidden neural network layers, is not adequate. There are two key ideas behind our solution. First, we use the category information as context to generate category-specific token embeddings via conditional self-attention. Second, we conduct multi-task training by predicting product category from profile texts as an auxiliary task; sharing parameters across tasks allows us to get token embeddings that are discriminative of the product categories and further improve attribute extraction. Multi-task training also makes our extraction model more robust towards wrong category assignment, which occurs often in real e-commerce websites.[1]

To the best of our knowledge, TXtract is the first deep neural network that has been applied to attribute value extraction for hierarchical taxonomies with thousands of product categories. In particular, we make the following contributions:

1. We develop TXtract, a taxonomy-aware deep neural network for attribute value extraction from product profiles for multiple product categories. In TXtract, we capture the *hierarchical* relations between categories into *category embeddings*, which in turn

[1]As an example, an ethernet cable might be incorrectly assigned under "Hair Brushes"; see https://www.amazon.com/dp/B012AE5EP4.

we use as context to generate category-specific token embeddings via conditional self-attention.

2. We improve attribute value extraction through multi-task learning: TXtract jointly extracts attribute values and predicts the product categories by sharing representations across tasks.

3. We evaluate TXtract on a taxonomy of 4,000 product categories and show that it substantially outperforms state-of-the-art models by up to 10% in F1 and 15% in coverage across *all* product categories.

We start in Section 4.2 by providing relevant background and defining our problem of focus. We continue as follows:

- We develop TXtract, a neural network that leverages Amazon's taxonomy with thousands of diverse categories and effectively extracts attribute values without manually labeled data (Section 4.3).

- We evaluate our ideas by conducting an experimental evaluation on attribute value extraction from 4,000 product categories (Sections 4.4 and 4.5).

- We demonstrate the integration of TXtract into Amazon's AutoKnow, a system that collects knowledge facts for over 10K product categories and serves such information to Amazon search and product detail pages (Section 4.6).

Finally, we discuss the implications of our work (Section 4.7). The material described in this chapter appears in [Karamanolakis et al., 2020b; Dong et al., 2020].

4.2 Background and Problem Definition

Here, we discuss background on attribute value extraction and multi-task learning/meta-learning (Sections 4.2.1 and 4.2.2, respectively), and define our problem of focus (Section 4.2.3)

Input	Ben	&	Jerry's	black	cherry	cheesecake	ice	cream
Output	O	O	O	B	I	E	O	O

Table 4.1: Example of input/output tag sequences for the "flavor" attribute of an ice cream product.

4.2.1 Attribute Value Extraction from Product Profiles

Attribute value extraction was originally addressed with rule-based techniques [Nadeau and Sekine, 2007; Vandic et al., 2012; Gopalakrishnan et al., 2012] followed by supervised learning techniques [Ghani et al., 2006; Putthividhya and Hu, 2011; Ling and Weld, 2012; Petrovski and Bizer, 2017; Sheth et al., 2017].

Recent approaches for attribute value extraction rely on the open-world assumption to discover attribute values that have never been seen during training [Zheng et al., 2018]. Most techniques address open attribute value extraction by extracting emerging attributes via sequence tagging, similar to named entity recognition (NER) [Putthividhya and Hu, 2011; Chiu and Nichols, 2016; Lample et al., 2016; Yadav and Bethard, 2018]. Specifically, each token of the input sequence $s = (x_1, \ldots, x_T)$ is assigned a separate tag from {B, I, O, E}, where "B," "I," "O," and "E" represent the beginning, inside, outside, and end of an attribute, respectively. (Not extracting any values corresponds to a sequence of "O"-only tags.) Table 4.1 shows an input/output example of *flavor* value extraction from (part of) a product title. Given this output tag sequence, "black cherry cheesecake" is extracted as a *flavor* for the ice cream product.

State-of-the-art approaches address open attribute value extraction with deep sequence tagging models [Zheng et al., 2018; Xu et al., 2019; Rezk et al., 2019]. However, all previous methods can be adapted to a small number of categories and require many labeled data points per category.[2] Even the active learning method of [Zheng et al., 2018] requires humans to annotate at least hundreds of carefully selected examples per category. Our work differs

[2][Zheng et al., 2018] considered three categories: "Dog Food," "Cameras," and "Detergent." [Xu et al., 2019] consider one category: "Sports & Entertainment." [Rezk et al., 2019] considered 21 categories and trained a separate model for each category.

from previous approaches as we consider thousands of product categories organized in a hierarchical taxonomy.

4.2.2 Multi-Task and Meta Learning

Our framework is related to multi-task learning [Caruana, 1997] as we train a single model simultaneously on all categories (tasks). Traditional approaches consider a small number of different tasks, ranging from 2 to 20, and employ hard parameter sharing [Alonso and Plank, 2017; Yang et al., 2017; Ruder, 2019]: the first layers of the neural networks are shared across all tasks, while the separate layers (or "heads") are used for each individual task. In our setting, with thousands of different categories (tasks), our approach is efficient as we use a *single* head (rather than thousands) and effective as we distinguish between categories through low-dimensional category embeddings. Our work is also related to meta-learning approaches based on task embeddings [Finn et al., 2017; Achille et al., 2019; Lan et al., 2019]: the target tasks are represented in a low-dimensional space that captures task similarities. However, we generate category embeddings that reflect the *already available, hierarchical* structure of product categories in the taxonomy provided by experts.

4.2.3 Problem Definition

We represent the product taxonomy as a tree $\mathcal{C}$, where the root node is named "Product" and each taxonomy node corresponds to a distinct product category $c \in \mathcal{C}$. A directed edge between two nodes represents the category-to-subcategory relationship. A product is assigned to a category node in $\mathcal{C}$. In practice, there are often thousands of nodes in a taxonomy tree and the category assignment of a product may be incorrect. We now formally define our problem as follows.

Consider a product from a category c and the sequence of tokens $s = (x_1, \ldots, x_T)$ from its profile, where T is the sequence length. Let a be a target attribute for extraction. Attribute extraction identifies sub-sequences of tokens from s such that each sub-sequence represents a

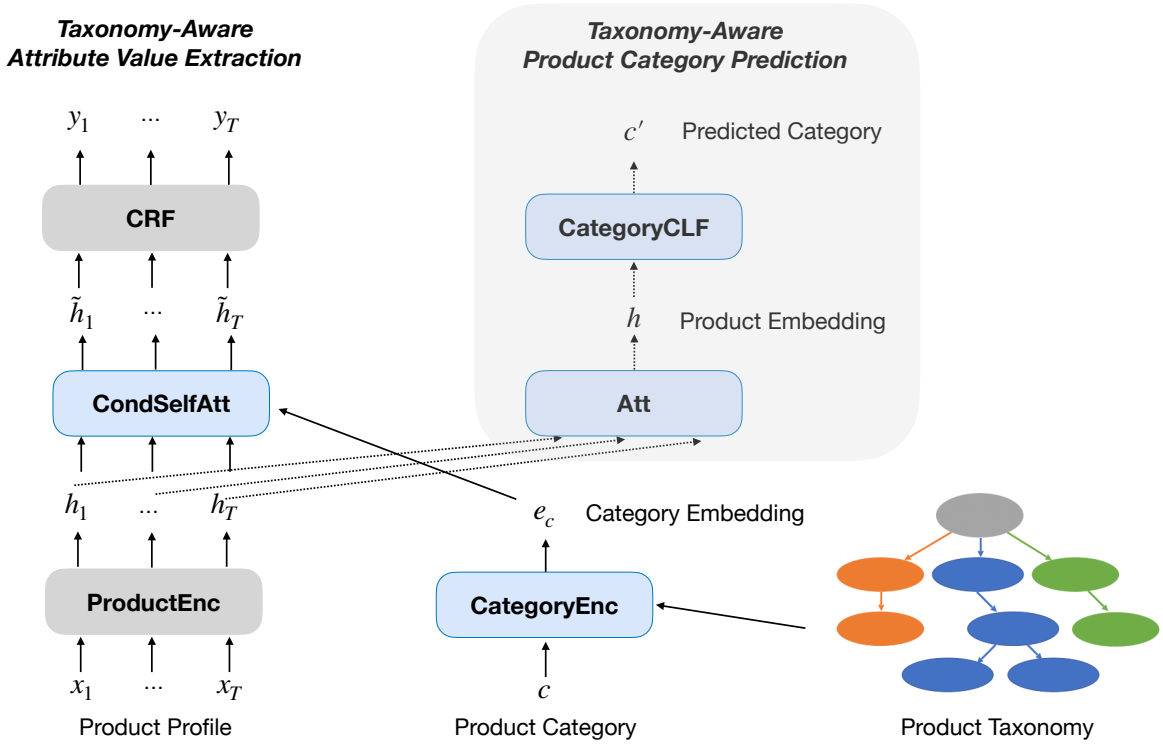


Figure 4.2: TXtract architecture: tokens $(x_1, \ldots, x_T)$ are classified to BIOE attribute tags $(y_1, \ldots, y_T)$ by conditioning to the product's category embedding e_c. TXtract is jointly trained to extract attribute values and assign a product to taxonomy nodes.

value for a. For instance, given (1) a product title $s =$"Ben & Jerry's Strawberry Cheesecake Ice Cream 16 oz," (2) a product category $c =$ "Ice Cream," and (3) a target attribute $\alpha =$ *flavor*, we would like to extract "Strawberry Cheesecake" as a *flavor* for this product. Note that *we may not see all valid attribute values during training.*

4.3 Taxonomy-Aware Network (TXtract)

In this work, we address open attribute value extraction using a taxonomy-aware deep sequence tagging model, TXtract. Figure 4.2 shows the model architecture, which contains two key components: attribute value extraction and product category prediction, accounting for the two tasks in multi-task training. Both components are taxonomy aware, as we describe next in detail.

4.3.1 Taxonomy-Aware Attribute Value Extraction

TXtract leverages the product taxonomy for attribute value extraction. The underlying intuition is that knowing the product category may help infer attribute applicability and associate the product with a certain range of valid attribute values. Our model uses the category embedding in conditional self-attention to guide the extraction of category-specific attribute values.

Product encoder. The product encoder ("ProductEnc") represents the text tokens of the product profile $(x_1, \ldots, x_T)$ as low-dimensional, real-valued vectors:

$$\mathbf{h}_1, \ldots \mathbf{h}_T = \text{ProductEnc}(\text{x}_1, \ldots, \text{x}_\text{T}) \in \mathbb{R}^d. \tag{4.1}$$

To effectively capture long-range dependencies between the input tokens, we use word embeddings followed by bidirectional LSTMs (BiLSTMs), similar to previous state-of-the-art approaches [Zheng et al., 2018; Xu et al., 2019].

Category encoder. Our category encoder ("CategoryEnc") encodes the hierarchical structure of product categories such that TXtract understands expert-defined relations across categories, such as "Lager" is a sub-category of "Beer". In particular, we embed each product category c (taxonomy node) into a low-dimensional latent space:

$$\mathbf{e}_c = \text{CategoryEnc}(c) \in \mathbb{R}^m. \tag{4.2}$$

To capture the hierarchical structure of the product taxonomy, we embed product categories into the m-dimensional Poincaré ball [Nickel and Kiela, 2017], because its underlying geometry has been shown to be appropriate for capturing both similarity and hierarchy.

Category conditional self-attention. The key component for taxonomy-aware value extraction is category conditional self-attention ("CondSelfAtt"). CondSelfAtt generates category-specific token embeddings ($\tilde{\mathbf{h}}_i \in \mathbb{R}^d$) by conditioning on the category embedding $\mathbf{e}_c$:

$$(\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_T) = \text{CondSelfAtt}((\mathbf{h}_1, \ldots, \mathbf{h}_T), \mathbf{e}_c). \tag{4.3}$$

To leverage the mutual interaction between all pairs of token embeddings $\mathbf{h}_t$, $\mathbf{h}_{t'}$ and the category embedding $\mathbf{e}_c$ we use self-attention and compute pairwise sigmoid attention weights:

$$\alpha_{t,t'} = \sigma(\mathbf{w}_\alpha^T \mathbf{g}_{t,t'} + b_\alpha), \qquad t, t' = 1, \ldots, T. \tag{4.4}$$

We compute $\mathbf{g}_{t,t'}$ using both the token embeddings $\mathbf{h}_t$, $\mathbf{h}_{t'}$ and the category embedding $\mathbf{e}_c$:

$$g_{t,t'} = \tanh(\mathbf{W}_1 \mathbf{h}_t + \mathbf{W}_2 \mathbf{h}_{t'} + \mathbf{W}_3 \mathbf{e}_c + \mathbf{b}_g), \tag{4.5}$$

where $\mathbf{W}_1 \in \mathbb{R}^{p \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{p \times d}$, $\mathbf{W}_3 \in \mathbb{R}^{p \times m}$, $\mathbf{w}_\alpha \in \mathbb{R}^p$ are trainable attention matrices and $\mathbf{b}_g \in \mathbb{R}^p$, $b_\alpha \in \mathbb{R}$, are trainable biases. The $T \times T$ attention matrix $\mathbf{A} = a_{t,t'}$ stores the pairwise attention weights. The contextualized token embeddings are computed as:

$$\tilde{\mathbf{h}}_t = \sum_{t'=1}^{T} \alpha_{t,t'} \cdot \mathbf{h}_{t'}. \tag{4.6}$$

CRF layer. We feed the contextualized token representations $\tilde{\mathbf{h}} = (\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_T)$ to CRFs to get the sequence of BIOE tags with the highest probability:

$$(y_1, \ldots, y_T) = \text{CRF}(\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_t). \tag{4.7}$$

We then extract attribute values as valid sub-sequences of the input tokens $(x_1, \ldots, x_T)$ with B/I/E tags (see Section 4.2.1).

Training for attribute value extraction. Our training objective for attribute value extraction is to minimize the negative conditional log-likelihood of the model parameters on N training products s_i with ground truth labels $(\hat{y}_{i1}, \ldots, \hat{y}_{iT})$:

$$L_a = -\sum_{i=1}^{N} \log Pr(\hat{y}_{i1}, \ldots, \hat{y}_{iT} \mid x_i, c_i) \tag{4.8}$$

We train our model on all categories in parallel, thus leveraging for a given category products from related categories. To generate training sequence labels from the corresponding attribute values, we use the distant supervision framework of [Mintz et al., 2009], similar to [Xu et al., 2019], by generating tagging labels according to existing (sparse) values in the catalog.

4.3.2 Taxonomy-Aware Product Category Prediction

We now describe how we train TXtract for the auxiliary task of product category prediction through multi-task learning. Our main idea is that by encouraging TXtract to predict the product categories using only the product profile, the model will learn token embeddings that are discriminative of the product categories. Thus, we introduce an inductive bias for more effective category-specific attribute value extraction.

Attention layer. Our attention component ("Att") represents the product profile $(x_1, \ldots, x_T)$ as a single vector $\mathbf{h} \in \mathbb{R}^n$ computed through the weighted combination of the ProductEnc's embeddings $(\mathbf{h}_1, \ldots, \mathbf{h}_T)$:

$$\mathbf{h} = \sum_{t=1}^{T} \beta_t \cdot \mathbf{h}_t. \tag{4.9}$$

This weighted combination allows tokens that are more informative for a product's category to get higher "attention weights" $\beta_t \in [0, 1]$. For example, we expect $x_t =$ "frozen" to receive a relatively high β_t for the classification of a product to the "Ice Cream" category. We

compute the attention weights as:

$$\beta_t = \text{softmax}(\mathbf{u}_c^T \tanh(\mathbf{W}_c \mathbf{h}_t + \mathbf{b}_c)), \tag{4.10}$$

where $\mathbf{W}_c \in \mathbb{R}^{q \times d}$, $\mathbf{b}_c \in \mathbb{R}^q$, $\mathbf{u}_c \in \mathbb{R}^q$ are trainable attention parameters.

Category classifier. Our category classifier ("CategoryCLF") classifies the product embedding $\mathbf{h}$ to the taxonomy nodes. In particular, we use a sigmoid classification layer to predict the probabilities of the taxonomy nodes:

$$(p_1, \ldots, p_{|\mathcal{C}|}) = \text{sigmoid}(\mathbf{W}_d \mathbf{h} + \mathbf{b}_d), \tag{4.11}$$

where $\mathbf{W}_d \in \mathbb{R}^{|\mathcal{C}| \times d}$ and $\mathbf{b}_d \in \mathbb{R}^{|\mathcal{C}|}$ are trainable parameters. We compute sigmoid (instead of softmax) node probabilities because we treat category prediction as *multi-label* classification, as we describe next.

Training for category prediction. Training for "flat" classification of products to thousands of categories is not effective because the model is fully penalized if it does not predict the exact true category $\hat{c}$ while at the same time ignores parent-children category relations. Here, we conduct "hierarchical" classification by incorporating the hierarchical structure of the product taxonomy into a *taxonomy-aware* loss function.

The insight behind our loss function is that a product assigned under $\hat{c}$ could also be assigned under any of the ancestors of $\hat{c}$. Thus, we consider hierarchical multi-label classification and encourage TXtract to assign a product to all nodes in the path from $\hat{c}$ to the root, denoted by $(\hat{c}_K, \hat{c}_{K-1}, \ldots, \hat{c}_1)$, where K is the level of the node $\hat{c}$ in the taxonomy tree. The model is thus encouraged to learn the hierarchical taxonomy relations and will be penalized less if it predicts high probabilities for ancestor nodes (e.g., "Beer" instead of "Lager" in Figure 4.1).

Our minimization objective is the *weighted* version of the binary cross-entropy (instead of *unweighted* categorical cross-entropy) loss:[3]

$$L_b = \sum_{c \in \mathcal{C}} w_c (y_c \cdot \log p_c + (1 - y_c) \cdot \log(1 - p_c)), \tag{4.12}$$

For the nodes in the path from $\hat{c}$ to the root $(\hat{c}_K, \hat{c}_{K-1}, \ldots, \hat{c}_1)$, we define positive labels $y_c = 1$ and weights w_c that are exponentially decreasing $(w^0, w^1, \ldots, w^{K-1})$, where $0 < w \leq 1$ is a tunable hyper-parameter. The remaining nodes in $\mathcal{C}$ receive negative labels $y_c = 0$ and fixed weight $w_c = w^{K-1}$.

4.3.3 Multi-Task Training

We jointly train TXtract for attribute value extraction and product category prediction by combining the loss functions of Eq. (4.8) and Eq. (4.12):

$$L = \gamma \cdot L_a + (1 - \gamma) \cdot L_b, \tag{4.13}$$

where $\gamma \in [0, 1]$ is a tunable hyper-parameter. Here, we employ multi-task learning, and share ProductEnc across both tasks.

We now turn into the empirical evaluation of TXtract and its comparison with state-of-the-art models and strong baselines for attribute value extraction on 4000 product categories. As we will show, TXtract leads to substantial improvement across all categories, showing the advantages of leveraging the product taxonomy.

4.4 Experimental Settings

In this section, we present our experimental setting.

[3]For simplicitly in notation, we define Eq. (4.12) for a single product. Defining for all training products is straightforward.

Dataset. We trained and evaluated TXtract on products from public web pages of Amazon.com. We randomly selected 2 million products from 4000 categories under 4 general domains (sub-trees) in the product taxonomy: Grocery, Baby product, Beauty product, and Health product.

Experimental setup. We split our dataset into training (60%), validation (20%), and test (20%) sets. We experimented with extraction of *flavor*, *scent*, and *brand* values from product titles, and with *ingredient* values from product titles and descriptions. For each attribute, we trained TXtract on the training set, we fine-tuned hyper-parameters on the validation set, and evaluated the model performance on the held-out test set.

Evaluation metrics. For a robust evaluation of attribute value extraction, we report several metrics. For a test product, we consider as true positive the case where the extracted values match at least one of the ground truth values (as some of the ground truth values may not exist in the text) and do not contain any wrong values.[4] We compute *Precision* (Prec) as the number of "matched" products divided by the number of products for which the model extracts at least one attribute value; *Recall* (Rec) is the number of "matched" products divided by the number of products associated with attribute values; finally, *F1* score is the harmonic mean of Prec and Rec. To obtain a global picture of the model's performance, we consider micro-average scores (Mi*), which first aggregates products across categories and computes Prec/Rec/F1 globally. To evaluate per-category performance we consider macro-average scores (Ma*), which first computes Prec/Rec/F1 for each category and then aggregates per-category scores. To evaluate the capability of our model to discover (potentially new) attribute values, we also report the *Value vocabulary* (Vocab) as the total number of unique attribute values extracted from the test set (higher number is often better); *Coverage* (Cov) is then the number of products for which the model extracted at least one

[4]For example, if the ground-truth is $[v_1]$ but the system extracts $[v_1, v_2, v_3]$, the extraction is considered as incorrect.

attribute value, divided by the total number of products.

For product category (multi-label) classification we report the area under Precision-Recall curve (AUPR), Prec, Rec, and F1 score.

Model configuration. We implemented our model in Tensorflow [Abadi et al., 2016] and Keras.[5] For a fair comparison, we consider the same configuration as OpenTag for the ProductEnc (BiLSTM)[6] and CRF components. We initialize the word embedding layer using 100-dimensional pre-trained Glove embeddings [Pennington et al., 2014]. We use masking to support variable-length input. Each of the LSTM layers has a hidden size of 100 dimensions, leading to a BiLSTM layer with $d = 200$ dimensional embeddings. We set the dropout rate to 0.4. For CategoryEnc, we train $m = 50$-dimensional Poincaré embeddings.[7] For CondSelfAtt, we use $p = 50$ dimensions. For Att, we use $q = 50$ dimensions. For multi-task training, we obtain satisfactory performance with default hyper-parameters $\gamma = 0.5$, $w = 1$, while we leave fine-tuning for future work. For parameter optimization, we use Adam [Kingma and Ba, 2014] with a batch size of 32. We train our model for up to 30 epochs and quit training if the validation loss does not decrease for more than three epochs.

Model Comparison. We compared our model with state-of-the-art models in the literature and introduced additional strong baselines:

1. "OpenTag": the model of [Zheng et al., 2018]. It is a special case of our system that consists of the ProductEnc and CRF components without leveraging the taxonomy.

2. "Title+*": a class of models for conditional attribute value extraction, where the taxonomy is introduced by artificially appending extra tokens ($(x'_1, \ldots, x'_{T'})$ and a special

[5] https://keras.io/

[6] We expect to see further performance improvement by considering pre-trained language models [Radford et al., 2018; Devlin et al., 2019] for ProductEnc, which we leave for future work.

[7] We use the public code provided by [Nickel and Kiela, 2017]: https://github.com/facebookresearch/poincare-embeddings.

separator token (<SEP>) to the beginning of a product's text, similar to [Johnson et al., 2017]:

$$x' = (x'_1, \ldots, x'_{T'}, \text{<SEP>}, x_1, \ldots, x_T)$$

Tokens $(x'_1, \ldots, x'_{T'})$ contain category information such as unique category id ("Title+id"), category name ("Title+name"), or the names of all categories in the path from the root to the category node, separated by an extra token <SEP2> ("Title+path").

3. "Concat-*": a class of models for taxonomy-aware attribute value extraction that concatenate the category embedding to the word embedding (-wemb) or hidden BiLSTM embedding layer (-LSTM) instead of using conditional self-attention. We evaluate Euclidean embeddings ("Concat-*-Euclidean") and Poincaré embeddings ("Concat-*-Poincaré").

4. "Gate": a model that leverages category embeddings $\mathbf{e}_c$ in a gating layer [Cho et al., 2014; Ma et al., 2019]: $\tilde{\mathbf{h}}_t = \mathbf{h}_t \otimes \sigma(\mathbf{W}_4\mathbf{h}_t + \mathbf{W}_5\mathbf{e}_c)$, where $\mathbf{W}_4 \in \mathbb{R}^{p\times d}$, $\mathbf{W}_5 \in \mathbb{R}^{p\times m}$ are trainable matrices, and $\otimes$ denotes element-wise multiplication. Our conditional self-attention is different as it leverages pairwise instead of single-token interactions with category embeddings.

5. "CondSelfAtt": the model with our conditional self-attention mechanism (Section 4.3.1). CondSelfAtt extracts attribute values but does not predict the product category.

6. "MT-*": a multi-task learning model that jointly performs (*not* taxonomy-aware) attribute value extraction and category prediction. "MT-flat" assumes "flat" categories, whereas "MT-hier" considers the hierarchical structure of the taxonomy (Section 4.3.2).

7. "TXtract": our model that jointly performs *taxonomy-aware* attribute value extraction (same as CondSelfAtt) and *hierarchical* category prediction (same as MT-hier).

Here, we do not report previous models (e.g., BiLSTM-CRF) for sequence tagging [Huang

				Micro-average			Macro-average		
Attr.	**Model**	**Vocab**	**Cov**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**
Flavor	OpenTag	6,756	73.2	57.5	70.3	49.6	54.6	68.0	47.3
	TXtract	**13,093**	**83.9**	**63.3**	**70.9**	**57.8**	**59.3**	**68.4**	**53.8**
Scent	OpenTag	10,525	75.8	70.6	**87.6**	60.2	59.3	**79.7**	50.8
	TXtract	**13,525**	**83.2**	**73.7**	86.1	**65.7**	**59.9**	78.3	**52.1**
Brand	OpenTag	48,943	73.1	63.4	81.6	51.9	51.7	75.1	41.5
	TXtract	**64,704**	**82.9**	**67.5**	**82.7**	**56.5**	**55.3**	**75.2**	**46.8**
Ingred.	OpenTag	9,910	70.0	35.7	46.6	29.1	20.9	34.6	16.7
	TXtract	**18,980**	**76.4**	**37.1**	**48.3**	**30.1**	**24.2**	**37.4**	**19.8**
Average relative increase			↑11.7%	↑6.2%	↑1.0%	↑9.3%	↑10.4%	↑6.8%	↑11.9%

Table 4.2: Extraction results for *flavor*, *scent*, *brand*, and *ingredients* across 4,000 categories. Across all attributes, TXtract improves OpenTag by 11.7% in coverage, 6.2% in micro-average F1, and 10.4% in macro-average F1.

et al., 2015; Kozareva et al., 2016; Lample et al., 2016], as OpenTag has been shown to outperform these models in [Zheng et al., 2018]. Moreover, when considering attributes separately, the model of [Xu et al., 2019] is the same as OpenTag, but with a different ProductEnc component; since we use the same ProductEnc for all alternatives, we expect the same trend and do not report its performance.

4.5 Experimental Results across 4,000 Product Categories

In this section, we report our experimental results and show the empirical benefits of TXtract across all 4,000 categories, highlighting the advantages of leveraging the product taxonomy.

Table 4.2 reports the results across all categories. Over all categories, our taxonomy-aware TXtract substantially improves over the state-of-the-art OpenTag by up to 10.1% in Micro F1, 14.6% in coverage, and 93.8% in vocabulary (for *flavor*).

Table 4.3 shows results for the four domains of our taxonomy under different training granularities: training on all domains versus training only on the target domain. Regardless of the configuration, TXtract substantially outperforms OpenTag, showing the general advantages of our approach. Interestingly, although training a single model on all of the

Domain			OpenTag/TXtract
Train	**Test**	**Attr.**	**Micro F1**
all	Grocery	*Flavor*	60.3 / **64.9** ↑7.6%
Grocery	Grocery		65.4 / **70.5** ↑7.8%
all	Baby	*Flavor*	54.4 / **63.0** ↑15.8%
Baby	Baby		69.2 / **71.8** ↑3.8%
all	Beauty	*Scent*	76.9 / **79.5** ↑3.4%
Beauty	Beauty		76.9 / **79.0** ↑2.7%
all	Health	*Scent*	63.0 / **69.1** ↑9.7%
Health	Health		60.9 / **63.5** ↑4.3%

Table 4.3: Evaluation results for each domain under training configurations of different granularity. TXtract outperforms OpenTag under all configurations.

four domains obtains lower F1 for *Flavor*, it obtains better results for *Scent*: training fewer models does not necessarily lead to lower quality and may actually improve extraction by learning from neighboring taxonomy trees.

Ablation study. Table 4.4 reports the performance of several alternative approaches for *flavor* value extraction across all categories. OpenTag does not leverage the product taxonomy, so it is outperformed by most approaches that we consider in this work.

"Title+*" baselines fail to leverage the taxonomy, thus leading to lower F1 score than OpenTag: implicitly leveraging categories as artificial tokens appended to the title is not effective in our setting.

Representing the taxonomy with category embeddings leads to significant improvement over OpenTag and "Title+*" baselines: even simpler approaches such as "Concat-*-Euclidean" outperform OpenTag across all metrics. However, "Concat-*" and "Gate-*" do not leverage category embeddings as effectively as "CondSelfAtt": conditioning on the category embedding for the computation of the pair-wise attention weights in the self-attention layer appears to be the most effective approach for leveraging the product taxonomy.

In Table 4.4, both MT-flat and MT-hier, which do not condition on the product taxonomy, outperform OpenTag on attribute value extraction: by learning to predict the product category, our model implicitly learns to condition on the product category for effective at-

Model	TX	MT	Micro F1
OpenTag	-	-	57.5
Title+id	✓	-	55.7 ↓3.1%
Title+name	✓	-	56.9 ↓1.0%
Title+path	✓	-	54.3 ↓5.6%
Concat-wemb-Euclidean	✓	-	60.1 ↑4.5%
Concat-wemb-Poincaré	✓	-	60.6 ↑5.4%
Concat-LSTM-Euclidean	✓	-	60.1 ↑4.5%
Concat-LSTM-Poincaré	✓	-	60.8 ↑5.7%
Gate-Poincaré	✓	-	60.6 ↑5.4%
CondSelfAtt-Poincaré	✓	-	61.9 ↑7.7
MT-flat	-	✓	60.9 ↑5.9%
MT-hier	-	✓	61.5 ↑7.0%
Concat & MT-hier	✓	✓	62.3 ↑8.3%
Gate & MT-hier	✓	✓	61.1 ↑6.3%
CondSelfAtt & MT-hier	✓	✓	**63.3** ↑10.1%

Table 4.4: Ablation study for *flavor* extraction across 4,000 categories. "TX" column indicates whether the taxonomy is leveraged for attribute value extraction (Section 4.3.1). "MT" column indicates whether multi-task learning is used (Section 4.3.2).

Category Prediction	AUPR	F1	Prec	Rec
Flat	0.61	53.9	74.2	48.0
Hierarchical	**0.68**	**62.7**	**80.4**	**56.9**

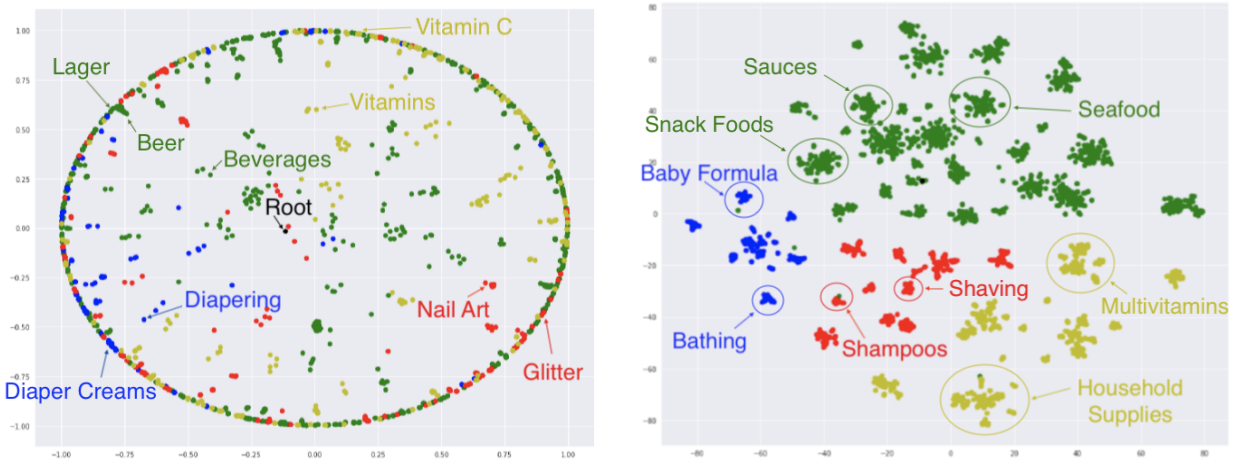
Table 4.5: Performance of product classification to the 4,000 nodes in the taxonomy using flat versus hierarchical multi-task learning.

tribute value extraction. MT-hier outperforms MT-flat: leveraging the hierarchical structure of the taxonomy is more effective than assuming flat categories.

Table 4.5 shows that category prediction is more effective when considering the hierarchical structure of the categories into our taxonomy-aware loss function than assuming flat categories.

Visualization of Poincaré embeddings Poincaré embeddings effectively capture the hierarchical structure of the product taxonomy: Figure 4.3a plots the embeddings of product categories in the 2-dimensional Poincaré disk.[8] Figure 8.3b plots the embeddings trained in

[8]We train 2-dimensional Poincaré embeddings only for visualization. In our experiments we use $d = 50$ dimensions.



(a) Taxonomy embeddings in the 2-dimensional Poincaré disk, where the distance of points grows exponentially to the radius. Leaf nodes are placed close to the boundary of the disk.

(b) Taxonomy embeddings projected from the 50-dimensional Poincaré ball to the 2-dimensional Euclidean space using t-SNE. Small clusters correspond to taxonomy sub-trees.

Figure 4.3: Poincaré embeddings of taxonomy nodes (product categories). Each point is a product category. Categories are colored based on the first-level taxonomy where they belong (green: Grocery products, blue: Baby products, red: Beauty products, yellow: Health products). Related categories in the taxonomy (e.g., categories belonging to the same sub-tree) have similar embeddings.

Figure 4.4: Examples of extracted attribute values from OpenTag and TXtract.

the 50-dimensional Poincaré ball and projected to the 2-dimensional Euclidean space through t-SNE [Maaten and Hinton, 2008].

Examples of extracted attribute values Figure 4.4 shows examples of product titles and attribute values extracted by OpenTag and TXtract. TXtract is able to detect category-specific values: in Figure 4.4a, "Purple Lemonade" is a valid *flavor* for "Vitamin Pills" but not for most of the other categories. OpenTag, which ignores product categories, fails to detect this value while TXtract successfully extracts it as a *flavor*. TXtract also learns attribute applicability: in Figure 4.4d, OpenTag erroneously extracts "palette" as *scent* for an "Eyeshadow" product, while this product should not have *scent*; on the other hand, TXtract, which considers category embeddings, does not extract any *scent* values for this product.

4.6 Integration of TXtract into Amazon's Product Knowledge Graph

We presented our method for large-scale attribute value extraction for products from a taxonomy with thousands of product categories. TXtract is both efficient and effective: it leverages the taxonomy into a deep neural network to improve extraction quality and can extract attribute values on all categories in parallel. TXtract significantly outperforms state-of-the-art approaches under a taxonomy with thousands of product categories.

TXtract has been a core component of Amazon's automatic knowledge graph of products, or AutoKnow [Dong et al., 2020]. In [Dong et al., 2020], we discuss the challenges associated with the organization of product knowledge in structured form to help downstream applications such as product search and question answering. Using TXtract as one of its main components, AutoKnow scales across tens of thousands of diverse categories and imputes missing values in the product Catalog without extra manual annotation efforts. AutoKnow collects knowledge facts for over 10K product categories, and the collected knowledge has been used for Amazon search and product detail pages.

4.7 Conclusions

In this chapter, we presented a novel method for large-scale attribute value extraction for products from a taxonomy with thousands of product categories. We summarize the contributions of this chapter as follows: (i) we developed TXtract, a taxonomy-aware deep neural network that extracts attribute values on all product categories in parallel. TXtract captures the hierarchical relations between categories into category embeddings, which in turn are used as context to generate category-specific token embeddings via conditional self-attention (Section 4.3.1); (ii) we developed a multi-task learning framework to jointly extract attribute values and predict product categories by sharing representations across the two tasks (Section 4.3.2); (iii) we performed a large-scale evaluation of TXtract across 4,000 product categories (Sections 4.4 and 4.5); and (iv) we discussed the integration of TXtract with Amazon's AutoKnow (Section 4.6).

Our findings show that TXtract is both effective and efficient: it leverages the taxonomy into a deep neural network to improve extraction quality and can extract attribute values on all categories in parallel. We also showed that TXtract substantially outperforms state-of-the-art models by up to 10% in F1 and 15% in coverage across all 4,000 product categories. We further demonstrated how TXtract plays an important role in knowledge fact collection for tens of thousands of product categories at Amazon. Although this work focuses on e-commerce, our approach to leverage taxonomies can be applied to broader domains such as finance, education, and biomedical research. We leave experiments on these domains for future work and now turn into a new machine teaching framework that can be applied across a broad set of tasks where taxonomies might not be available.

Chapter 5: Weakly-Supervised Text Classification with Seed Words

In Chapters 3 and 4, we presented two neural network architectures that we designed to leverage coarse-grained supervision and hierarchical category relationships, respectively, and highlighted their successful integration into operational systems. While these two approaches effectively address specific applications, it is hard to apply them for more applications at scale. First, while coarse-grained supervision and category relationships are readily available for tasks such as review sentiment classification (e.g., in the form of user-provided star ratings) and product value extraction (e.g., in the form of hierarchical taxonomies), there are many applications for which such types of supervision may not be available and might be expensive to obtain, especially for languages and settings associated with limited resources for teaching machines. Second, designing application-specific neural architectures involves the cost of having to re-design new architectures to effectively support new language domains, new types of supervision, and new representation learning approaches. These two limitations of our approaches in Chapters 3 and 4 raise the need for more scalable types of supervision and more general teaching frameworks.

In this chapter, we present an architecture-agnostic framework that can be used for training classifiers using a different type of supervision, namely, class-indicative seed words. First, we focus on the problem of fine-grained aspect detection in product and restaurant reviews and provide motivation for the use of seed words as a scalable type of supervision (Section 5.1). Second, we discuss related work and define our problem of focus (Section 5.2). Third, we present our weakly-supervised co-training framework, ISWD, which can train aspect detectors using just a small number of seed words (Section 5.3). Then, we present our

experimental evaluation for fine-grained aspect detection in product and restaurant reviews (Sections 5.4 and 5.5) and describe the application of ISWD for additional classification problems (Section 5.6). Finally, we summarize the contributions of this chapter (Section 5.7).

5.1 Overview and Motivation

In Chapter 4, we addressed the problem of extracting the values of product attributes from online product profiles with the goal to store product knowledge in structured format. In this section, our goal is to analyze *online user reviews* about products and restaurants. In contrast to attributes (e.g., flavor, ingredients), we focus on aspects of the entity that users care about (e.g., price, quality). Consider for example the Amazon product review in Figure 5.1. The text discusses various aspects of the TV such as price, ease of use, and sound

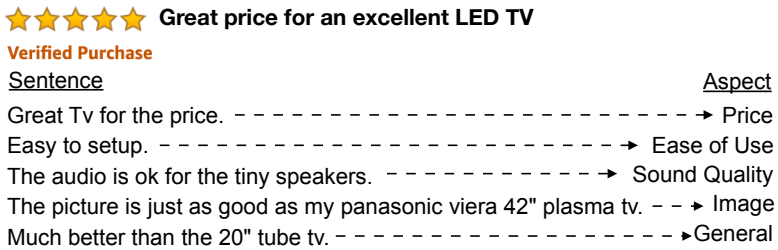


Figure 5.1: Example of product review with aspect annotations: each individual sentence of the review discusses a different aspect (e.g., price) of the TV.

quality. Individual review segments (e.g., sentences, clauses) may discuss different aspects, thus our goal is to train classifiers that classify each individual segment to the aspect that it discusses. Fine-grained aspect detection is a key task in downstream applications such as aspect-based sentiment analysis and multi-document summarization [Hu and Liu, 2004; Liu, 2012; Pontiki et al., 2016; Angelidis and Lapata, 2018b].

We focus on the problem of training segment classifiers when ground truth aspect labels are not available at any granularity. Indeed, reviews are often entered as unstructured, free-form text and do not come with aspect labels. It is infeasible to manually obtain segment annotations for retail stores like Amazon with millions of different products and as a result,

Aspect	Seed Words
Price (EN)	price, value, money, worth, paid
Image (EN)	picture, color, quality, black, bright
Food (EN)	food, delicious, pizza, cheese, sushi
Drinks (FR)	vin, bière, verre, bouteille, cocktail
Ambience (SP)	ambiente, mesas, terraza, acogedor, ruido

Table 5.1: Examples of aspects and five of their corresponding seed words in various domains (electronic products, restaurants) and languages ("EN" for English, "FR" for French, "SP" for Spanish).

neither supervised nor MIL approaches (Chapter 3) can be applied without aspect labels. Also, product knowledge bases used in Chapter 4 for distant supervision are not directly applicable in this setting, where we focus on the user's aspect of interest. Moreover, the topics learned by unsupervised neural topic models are not perfectly aligned with the users' aspects, so substantial human effort is required for interpreting and mapping the learned topics to meaningful aspects.

Here, we investigate whether neural networks can be effectively trained under this challenging setting when only a small number of descriptive keywords, or *seed words*, are available for each aspect class. Figure 5.1 shows examples of aspects and five of their corresponding seed words from our experimental datasets. In contrast to a classification label, which is only relevant for a single segment, a seed word can implicitly provide aspect supervision to potentially many segments. We assume that the seed words have already been collected either manually or automatically. Indeed, collecting a small[1] set of seed words per aspect is typically faster than manually annotating thousands of segments for training neural networks. As we will see, even noisy seed words that are only weakly predictive of the aspect will be useful for aspect detection.

Training neural networks for segment-level aspect detection using just a few seed words is a challenging task. Indeed, as a contribution of this work, we observe that current weakly supervised networks do not effectively leverage the predictive power of the available seed

[1]In our experiments, we only consider around 30 seed words per aspect. For comparison, the vocabulary of the datasets has more than 10,000 terms.

words. To address the shortcomings of previous seed word-based approaches, we propose a novel *weakly supervised* approach, which uses the available seed words in an effective way. In particular, we consider a *student-teacher* framework, according to which a bag-of-seed-words classifier (teacher) is applied on unlabeled segments to supervise a second model (student), which can be any supervised model, including neural networks.

Our approach introduces several important contributions:

1. Our teacher model considers each individual seed word as a (noisy) aspect indicator, which, as we will show, is more effective than previously proposed weakly supervised approaches.

2. By using only the teacher's aspect probabilities, our student generalizes better than the teacher and, as a result, the student outperforms both the teacher and previously proposed weakly supervised models.

3. We show how iterative co-training can be used to cope with noisy seed words: the teacher effectively estimates the predictive quality of the noisy seed words in an unsupervised manner using the associated predictions by the student.

4. Iterative co-training then leads to both improved teacher and student models. Overall, our ISWD approach consistently outperforms existing weakly supervised approaches, as we show with an experimental evaluation over six domains of product reviews and six multilingual datasets of restaurant reviews.

5. Our student-teacher approach could be applied for any classification task for which a small set of seed words describes each class. Towards our goal to support emerging applications, we apply ISWD for a new problem, the analysis of effects of COVID-19 on restaurant reviews.

We start by discussing related work and defining our problem of focus (Section 5.2). We continue as follows:

- We develop ISWD, a weakly-supervised co-training framework that leverages just seed words and unlabeled data and can be used with any type of classifier (Section 5.3).[2]
- We evaluate our ideas by conducting an experimental evaluation on fine-grained aspect detection of restaurant and product reviews (Sections 5.4 and 5.5).
- We demonstrate the application of ISWD for the analysis of what aspects of COVID-19 are discussed in restaurant reviews (Section 5.6).

Finally, we discuss the implications of our work (Section 5.7). The material described in this chapter appears in [Karamanolakis et al., 2019a; Karamanolakis et al., 2019b; Cao et al., 2021].

5.2 Related Work and Problem Definition

We now review relevant work on aspect detection (Section 5.2.1), co-training (Section 5.2.2), and knowledge distillation (Section 5.2.3). We also define our problem of focus (Section 5.2.4).

5.2.1 Segment-Level Aspect Detection

The goal of segment-level aspect detection is to classify a segment s to K aspects of interest.

Supervised approaches. Rule-based or traditional learning models for aspect detection have been outperformed by supervised neural networks [Liu et al., 2015; Poria et al., 2016; Zhang et al., 2018]. Following the notation from Section 2.2, supervised approaches first use a segment encoder ENC to encode a segment s into a vector $\mathbf{h}_i = \text{ENC}(s_i)$ and then use a segment classifier CLF to classify $\mathbf{h}_i$ to one of K predefined classes $[\mathcal{Y}] := \{1, 2, \ldots, K\}$: $\mathbf{p}_i = \text{CLF}(\mathbf{h}_i)$. For simplicity, here we write $\mathbf{p} = f(s)$. The parameters of the embedding

[2]Our Python implementation is publicly available at `https://github.com/gkaramanolakis/ISWD`.

function and the classification layer are learned using ground truth, segment-level aspect labels. However, aspect labels are not available in our setting, which hinders the application of supervised learning approaches.

Unsupervised approaches. Topic models have been used to train aspect detection with unannotated documents. Recently, neural topic models [Iyyer et al., 2016; Srivastava and Sutton, 2071; He et al., 2017] have been shown to produce more coherent topics than earlier models such as Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. In their Aspect Based Autoencoder (ABAE), [He et al., 2017] first use segment s to predict aspect probabilities $\mathbf{p} = f(s)$ and then use $\mathbf{p}$ to reconstruct an embedding $\mathbf{h}'$ for s as a convex combination of K aspect embeddings: $\mathbf{h}' = \sum_{k=1}^{K} p^k \mathbf{A}_k$, where $\mathbf{A}_k \in \mathbb{R}^d$ is the embedding of the k-th aspect. The aspect embeddings $\mathbf{A}_k$ are initialized by clustering the vocabulary embeddings using k-means with K clusters. ABAE is trained by minimizing the segment reconstruction error.[3]

Unfortunately, unsupervised topic models are not effective when used directly for aspect detection. In particular, in ABAE, the K topics learned to reconstruct the segments are not necessarily aligned with the K aspects of interest. A possible fix is to first learn $K' >> K$ topics and do a K'-to-K mapping as a post-hoc step. However, this mapping requires either aspect labels or substantial human effort for interpreting topics and associating them with aspects. This mapping is nevertheless not possible if the learned topics are not aligned with the aspects.

Weakly supervised approaches. Weakly supervised approaches use minimal domain knowledge (instead of ground truth labels) to model meaningful aspects. In our setting, domain knowledge is given as a set of seed words for each aspect of interest [Lu et al., 2011; Lund et al., 2017; Angelidis and Lapata, 2018b]. [Lu et al., 2011] use seed words as asymmetric priors in probabilistic topic models (including LDA). [Lund et al., 2017] use

[3]The reconstruction error can be efficiently estimated using contrastive max-margin objectives [Weston et al., 2011; Pennington et al., 2014].

LDA with fixed topic-word distributions, which are learned using seed words as "anchors" for topic inference [Arora et al., 2013]. Neither of these two approaches can be directly applied into more recent neural networks for aspect detection. [Angelidis and Lapata, 2018b] recently proposed a weakly supervised extension of the unsupervised ABAE. Their model, named Multi-seed Aspect Extractor, or MATE, initializes the aspect embedding $\mathbf{A}_k$ using the weighted average of the corresponding seed word embeddings (instead of the k-means centroids). To guarantee that the aspect embeddings will still be aligned with the K aspects of interest after training, [Angelidis and Lapata, 2018b] keep the aspect and word embeddings fixed throughout training. In this work, we will show that the predictive power of seed words can be leveraged more effectively by considering each individual seed word as a more direct source of supervision during training.

5.2.2 Co-training

Co-training [Blum and Mitchell, 1998] is a classic multi-view learning method for semi-supervised learning. In co-training, classifiers over different feature spaces are encouraged to agree in their predictions on a large pool of unlabeled examples. [Blum and Mitchell, 1998] justify co-training in a setting where the different views are conditionally independent given the label. Several subsequent works have relaxed this assumption and shown co-training to be effective in much more general settings [Balcan et al., 2005; Chen et al., 2011; Collins and Singer, 1999; Clark et al., 2018]. Co-training is also related to self-training (or bootstrapping) [Yarowsky, 1995], which trains a classifier using its own predictions and has been successfully applied for various NLP tasks [Collins and Singer, 1999; McClosky et al., 2006].

Recent research has successfully revisited these general ideas to solve NLP problems with modern deep learning methods. [Clark et al., 2018] propose "cross-view training" for sequence modeling tasks by modifying Bi-LSTMs for *semi-supervised* learning. [Ruder and Plank, 2018] show that classic bootstrapping approaches such as tri-training [Zhou

and Li, 2005] can be effectively integrated in neural networks for semi-supervised learning under domain shift. Our work provides further evidence that co-training can be effectively integrated into neural networks and combined with recent transfer learning approaches for NLP [Dai and Le, 2015; Howard and Ruder, 2018; Devlin et al., 2019; Radford et al., 2018], in a substantially different, *weakly supervised* setting where no ground-truth labels but only a few seed words are available for training.

5.2.3 Knowledge Distillation

Our approach is also related to the "knowledge distillation" framework [Buciluǎ et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015], which has received considerable attention recently [Lopez-Paz et al., 2016; Kim and Rush, 2016; Furlanello et al., 2018; Wang, 2019]. Traditional knowledge distillation aims at compressing a cumbersome model (teacher) to a simpler model (student) by training the student using both ground truth labels and the soft predictions of the teacher in a distillation objective. Our work also considers a student-teacher architecture and the distillation objective but under a considerably different, weakly supervised setting: (1) we do not use any labels for training and (2) we create conditions that allow the student to outperform the teacher; in turn, (3) we can use the student's predictions to learn a better teacher under co-training.

5.2.4 Problem Definition

Consider a corpus of text reviews from an entity domain (e.g., televisions, restaurants). Each review is split into segments (e.g., sentences, clauses). We also consider K pre-defined aspects of interest $(1, \ldots, K)$, including the "General" aspect, which we assume is the K-th aspect for simplicity. Different segments of the same review may be associated with different aspects but ground-truth aspect labels are *not* available for training. Instead, a small number of seed words G_k are provided for each aspect $k \in [K]$. Our goal is to use the corpus of training reviews and the available seed words $G = (G_1, \ldots, G_K)$ to train a classifier,

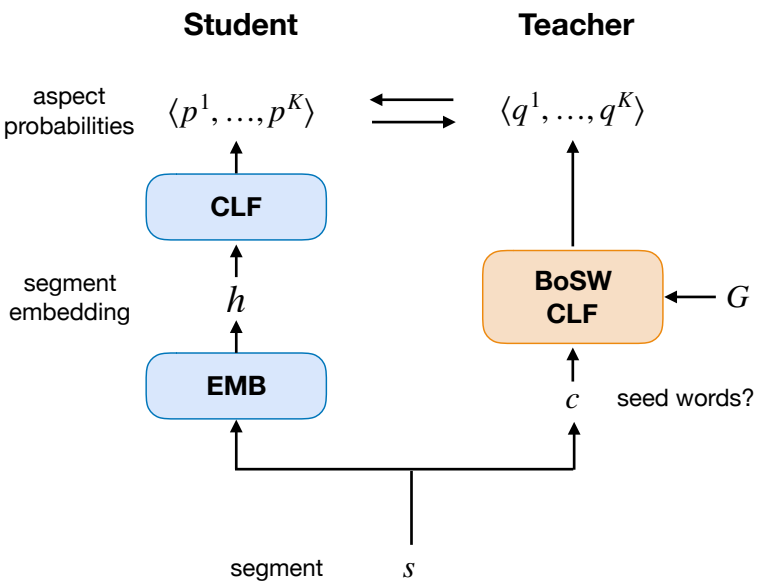


Figure 5.2: Our student-teacher approach for segment-level aspect detection using seed words.

which, given an unseen test segment s, predicts K aspect probabilities $\mathbf{p} = (p^1, \ldots, p^K)$.

5.3 Weakly-Supervised Co-Training with Seed Words (ISWD)

We now describe our weakly supervised framework for aspect detection. We consider a student-teacher architecture (Figure 5.2), where the teacher is a bag-of-words classifier based solely on the provided seed words (i.e., a "bag-of-seed-words" classifier), and the student is an embedding-based neural network trained on data "softly" labeled by the teacher (as in the distillation objective). In the rest of this section, we describe the individual components of our student-teacher architecture and our proposed algorithm for performing updates.

5.3.1 Teacher: A Bag-of-Seed-Words Classifier

Our teacher model leverages the available seed words G that are predictive of the K aspects. Let D denote the total number of seed words in G. We can represent a segment s_i using a bag-of-seed-words representation $c_i \in \mathbb{N}^D$, where c_i^j encodes the number of times

the j-th seed word occurs in s_i. (Note that c_i ignores the non-seed words.) The teacher's prediction for the k-th aspect is:

$$q_i^k = \frac{\exp(\sum_{j=1}^{D} \mathbb{1}\{j \in G_k\} \cdot c_i^j)}{\sum_{k'} \exp(\sum_{j=1}^{D} \mathbb{1}\{j \in G_{k'}\} \cdot c_i^j)}. \tag{5.1}$$

If no seed word appears in s, then the teacher predicts the "General" aspect by setting $q_i^K = 1$. Under this configuration the teacher uses seed words in a direct and intuitive way: it predicts aspect probabilities for the k-th aspect, which are proportional to the counts of the seed words under G_k, while if no seed word occurs in s, it predicts the "General" aspect. The classifier receives c_i as input and predicts $\mathbf{q}_i = (q_i^1, \ldots, q_i^K)$.

Although the teacher only uses seed words to predict the aspect of a segment, we also expect non-seed words to carry predictive power. Next, we describe the student network that learns to associate non-seed words with aspects.

5.3.2 Student: An Embedding-Based Network

Our student model is an embedding-based neural network (see Section 2.2 for details): a segment is first embedded ($\mathbf{h}_i = \text{EMB}(s_i) \in \mathbb{R}^d$) and then classified to the K aspects ($\mathbf{p}_i = \text{CLF}(\mathbf{h}_i)$) (see Section 5.2.1). The student does not use ground-truth aspect labels for training. Instead, it is trained by optimizing the distillation objective, i.e., the cross entropy between the teacher's (soft) predictions and the student's predictions:

$$H(\mathbf{q}_i, \mathbf{p}_i) = -\sum_k q_i^k \log p_i^k \tag{5.2}$$

While the teacher only uses the seed words in s_i to form its prediction $\mathbf{q}_i$, the student uses all the words in s_i. Thus, using the distillation loss for training, the student learns to use both seed words and non-seed words to predict aspects. As a result, the student is able to generalize better than the teacher and *predict aspects even in segments that do not contain any seed words*. To regularize the student model, we apply L2 regularization to the classifier's

weights and dropout regularization to the word embeddings [Srivastava et al., 2014]. As we will show in Section 4.5, our student with this configuration outperforms the teacher in aspect prediction.

5.3.3 Iterative Co-Training

In this section, we describe our iterative co-training algorithm to cope with noisy seed words. The teacher in Section 5.3.1 considers each seed word equally, which can be problematic because not all seed words are equally good for predicting an aspect. In this work, we propose to estimate the predictive quality of each seed word in an unsupervised way. Our approach is inspired in the Model Bootstrapped Expectation Maximization (MBEM) algorithm of [Khetan et al., 2018]. MBEM is guaranteed to converge (under mild conditions) when the number of training data is sufficiently large and the worker quality is sufficiently high. Here, we treat seed words as "noisy annotators" and adopt an iterative estimation procedure similar to MBEM, as we describe next.

We model the predictive quality of the j-th seed word as a weight vector $\mathbf{z}_j = (z_j^1, \ldots, z_j^K)$, where z_j^k measures the strength of the association with the k-th aspect. We thus change the teacher to consider seed word quality. In particular, we replace Equation (5.1) by:

$$q_i^k = \frac{\exp \sum_{j=1}^{D} \mathbb{1}\{j \in G_k\} \cdot \hat{z}_j^k \cdot c_i^j}{\sum_{k'} \exp \sum_{j=1}^{D} \mathbb{1}\{j \in G_{k'}\} \cdot \hat{z}_j^{k'} \cdot c_i^j}, \tag{5.3}$$

where $\hat{z}_j$ is the current estimate of z_j. As no ground-truth labels are available, we follow [Khetan et al., 2018] and estimate z_j via Maximum Likelihood Estimation using the student's predictions as the current estimate of the ground truth labels. In particular, we assume that the prediction of the student for a training segment s_i is $\hat{y}_i = \text{argmax}_k\, p_i^k$. Then, for each seed word we compute the quality estimate for the k-th aspect using the student's

Algorithm 1 Iterative Seed Word Distillation

Input: $\{s_i\}_{i\in[N]}$, D seed words grouped into K disjoint sets $G = (G_1, \ldots, G_K)$
Output: $\hat{f}$: predictor function for segment-level aspect detection

Predict $\{\mathbf{q}_i\}_{i\in[N]}$ (Eq. (5.1))	▷ *Apply teacher*
Repeat until convergence criterion	
Learn $\hat{f}$ (Eq. (5.2))	▷ *Train student*
Predict $\{\mathbf{p}_i = \hat{f}(s_i)\}_{i\in[N]}$	▷ *Apply student*
Update $\{z_j\}_{j\in[D]}$ (Eq. (5.4))	▷ *Update teacher*
Predict $\{\mathbf{q}_i\}_{i\in[N]}$ (Eq. (5.3))	▷ *Apply teacher*

predictions for N segments:

$$\hat{z}_j^k = \frac{\sum_{i=1}^{N} \mathbb{1}\{c_i^j > 0\}\mathbb{1}\{\hat{y}_i = k\}}{\sum_{k'}\sum_{i=1}^{N} \mathbb{1}\{c_i^j > 0\}\mathbb{1}\{\hat{y}_i = k'\}}. \tag{5.4}$$

According to Equation (5.4), the quality of the j-th seed word is estimated according to the student-teacher agreement on segments where the seed word appears.

Building upon the previous ideas, we present our Iterative Seed Word Distillation (ISWD) algorithm for effectively leveraging the seed words for fine-grained aspect detection. Each round of ISWD consists of the following steps (Algorithm 1): (1) we apply the teacher on unlabeled training segments to get predictions $\mathbf{q}_i$ (without considering seed word qualities); (2) we train the student using the teacher's predictions in the distillation objective of Equation (5.2);[4] (3) we apply the student in the training data to get predictions $\mathbf{p}_i$; and (4) we update the seed word quality parameters using the student's predictions in Equation (5.4).

In contrast to MATE, which uses the validation set (with aspect labels) to estimate seed weights in an initialization step, our proposed method is an unsupervised approach to modeling and adapting the seed word quality during training. We stop this iterative procedure after the disagreement between the student's and teacher's hard predictions in

[4]Note that the quality-aware loss function proposed in [Khetan et al., 2018], which is an alternative form of noise-aware loss functions [Natarajan et al., 2013], is equivalent to our distillation loss: using the log loss as $l(.)$ in Equation (4) of [Khetan et al., 2018] yields the cross entropy loss.

Bags	Keyboards	Boots	Headsets	TVs	Vacuums
Size/Fit	Feel/Comfort	Comfort	Sound	Image	Accessories
Quality	Layout	Size	Comfort	Sound	Ease of Use
Looks	Build Quality	Look	Ease of Use	Connectivity	Suction Power
Compartments	Extra Function.	Materials	Connectivity	Customer Serv.	Build Quality
Handles	Connectivity	Durability	Durability	Ease of Use	Noise
Protection	Price	Weather Resist.	Battery	Price	Weight
Price	Noise	Price	Price	Apps/Interface	Customer Serv.
Customer Serv.	Looks	Color	Look	Size/Look	Price
General	General	General	General	General	General

Table 4: The 9 aspect classes per domain of product reviews (OPOSUM).

the training data stops decreasing. We empirically observe that 2-3 rounds are sufficient to satisfy this criterion. This observation also agrees with [Khetan et al., 2018], who only run their algorithm for two rounds.

We now turn into the empirical evaluation of ISWD and its comparison with state-of-the-art models and strong baselines for fine-grained aspect detection (Sections 5.4 and 5.5). As we will show, ISWD leverages seed words more effectively than previous approaches across several benchmark datasets. Also, in Section 5.6, we present additional applications of ISWD.

5.4 Experimental Settings

We now present our experimental setting for aspect detection on several datasets of product and restaurant reviews.

Datasets. We train and evaluate our models on Amazon product reviews for six domains from the OPOSUM dataset and on restaurant reviews in six languages from the SemEval-2016 Aspect-based Sentiment Analysis task, as discussed next.

The OPOSUM dataset [Angelidis and Lapata, 2018b] is a subset of the Amazon Product Dataset [McAuley et al., 2015], which contains Amazon reviews from 6 domains: Laptop Bags, Keyboards, Boots, Bluetooth Headsets, Televisions, and Vacuums. The validation and test segments of each domain have been manually annotated with 9 aspects (Table 4).

The reviews of each domain are already segmented by [Angelidis and Lapata, 2018b] into elementary discourse units (EDUs) using a Rhetorical Structure Theory parser [Feng and Hirst, 2012]. The average number of training, validation, and test segments across domains is around 1 million, 700, and 700 segments, respectively.

The datasets used in the SemEval-2016 Aspect-based Sentiment Analysis task [Pontiki et al., 2016] contain reviews for multiple domains and languages. Here, we use the six corpora of multilingual (English, Spanish, French, Russian, Dutch, Turkish) restaurant reviews. The training, validation, and test segments have been manually annotated with 12 aspects, which are shared across languages: Restaurant General, Food Quality, Service General, Ambience General, Food Style_Options, Food Prices, Restaurant Miscellaneous, Restaurant Prices, Drinks Quality, Drinks Style_Options, Location General, and Drinks Prices. The reviews of each language are already segmented into sentences. The average number of training and test segments across languages is around 2500 and 800 segments respectively. The training segments of restaurant reviews are significantly fewer than the training segments of product reviews. Therefore, for non-English reviews we report results after a single co-training round. For our co-training experiments we augment the English reviews dataset with 50,000 English reviews randomly sampled from the Yelp Challenge corpus.[5]

For a fair comparison, we use exactly the same 30 seed words (per aspect and domain) used in [Angelidis and Lapata, 2018b] for the product reviews and use the same extraction method described in [Angelidis and Lapata, 2018b] to extract 30 seed words for the restaurant reviews.

Experimental procedure. For a fair comparison, we use exactly the same pre-processing (tokenization, stemming, and word embedding) and evaluation procedure as in [Angelidis and Lapata, 2018b]. For each domain, we train our model on the training set without using any aspect labels, and only use the seed words G via the teacher. For each model, we report

[5] `https://www.yelp.com/dataset/challenge`

the average test performance over 5 different runs with the parameter configuration that achieves best validation performance. As evaluation metric, we use the micro-averaged F1.

Model configuration. For the student network, we experiment with various modeling choices for segment representations: bag-of-words (BOW) classifiers, the unweighted average of word2vec embeddings (W2V), the weighted average of word2vec embeddings using bilinear attention [Luong et al., 2015] (same setting as [He et al., 2017; Angelidis and Lapata, 2018b]), and the average of contextualized word representations obtained from the second-to-last layer of the pre-trained (self-attention based) BERT model [Devlin et al., 2019] (see Section 2.2 for details on BERT). For the English product reviews, we use the base uncased BERT model. For the multilingual restaurant reviews, we use the multilingual cased BERT model.[6]

In iterative co-training, we train the student network to convergence in each iteration (which may require more than one epoch over the training data). The student's parameters are optimized using Adam [Kingma and Ba, 2014] with learning rate 0.005 and mini-batch size 50. Moreover, we observed that the iterative process is more stable when we interpolate between weights of the previous iteration and the estimated updates instead of directly applying the estimated seed weight updates (according to Equation (5.3)).

Model comparison. For a robust evaluation of our approach, we compare the following models and baselines:

- **LDA-Anchors:** The topic model of [Lund et al., 2017] using seed words as "anchors."
- **ABAE:** The unsupervised autoencoder of [He et al., 2017], where the learned topics were manually mapped to aspects.
- **MATE-*:** The MATE model of [Angelidis and Lapata, 2018b] with various configurations: initialization of the aspect embeddings $\mathbf{A}_k$ using the unweighted/weighted

[6]Both models can be found in https://github.com/google-research/bert/blob/master/multilingual.md. The multilingual cased BERT model is recommended by the authors instead of the multilingual uncased BERT model.

	Product Review Domain						
Method	**Bags**	**Keyboards**	**Boots**	**Headsets**	**TVs**	**Vacuums**	**AVG**
LDA-Anchors	33.5	34.7	31.7	38.4	29.8	30.1	33.0
ABAE	38.1	38.6	35.2	37.6	39.5	38.1	37.9
MATE	46.2	43.5	45.6	52.2	48.8	42.3	46.4
MATE-unweighted	41.6	41.3	41.2	48.5	45.7	40.6	43.2
MATE-MT (best)	48.6	45.3	46.4	54.5	51.8	47.7	49.1
Teacher	55.1	52.0	44.5	50.1	56.8	54.5	52.2
Student-BoW	57.3	56.2	48.8	59.8	59.6	55.8	56.3
Student-W2V	59.3	57.0	48.3	**66.8**	**64.0**	57.0	58.7
Student-W2V-RSW	51.3	57.2	46.6	63.0	62.1	57.1	56.2
Student-ATT	60.1	55.6	49.9	66.6	63.4	58.2	58.9
Student-BERT	**61.4**	**57.5**	**52.0**	66.5	63.0	**60.4**	**60.2**

Table 5.3: Micro-averaged F1 reported for 9-class EDU-level aspect detection in product reviews.

average of seed word embeddings and an extra multi-task training objective (MT).[7]

- **Teacher:** Our bag-of-seed-words teacher.
- **Student-*:** Our student network trained with various configurations for the EMB function.
- ***-Gold:** Supervised models trained using ground truth aspect labels, which are only available for restaurant reviews. These models are not directly comparable with the other models and baselines.

5.5 Experimental Results

Tables 5.3 and 5.4 show the results for aspect detection on product and restaurant reviews, respectively. The rightmost column of each table reports the average performance across the 6 domains/languages.

[7]The multi-task training objective in MATE requires datasets from different domains but same language, thus it cannot be applied in our datasets of restaurant reviews.

	Restaurant Review Language						
Method	**En**	**Sp**	**Fr**	**Ru**	**Du**	**Tur**	**AVG**
W2V-Gold	58.8	50.4	50.4	69.3	51.4	55.7	56.0
BERT-Gold	63.1	51.6	50.5	64.6	53.5	55.3	56.4
MATE	41.0	24.9	17.8	18.4	36.1	39.0	29.5
MATE-unweighted	40.3	18.3	19.2	21.8	31.5	25.2	26.1
Teacher	44.9	41.8	34.1	54.4	40.7	30.2	41.0
Student-W2V	47.2	40.9	32.4	**59.0**	42.1	42.3	44.0
Student-ATT	47.8	41.7	32.9	57.3	44.1	45.5	44.9
Student-BERT	**51.8**	**42.0**	**39.2**	58.0	**43.0**	**45.0**	**46.5**

Table 5.4: Micro-averaged F1 reported for 12-class sentence-level aspect detection in restaurant reviews. The fully supervised *-Gold models are not directly comparable with the weakly supervised models.

MATE-* models outperform ABAE. Using the seed words to initialize aspect embeddings leads to more accurate aspect predictions than mapping the learned (unsupervised) topics to aspects.

LDA-Anchors performs worse than MATE-* models. Although averages of seed words were used as "anchors" in the "Tandem Anchoring" algorithm, we observed that the learned topics did not correspond to our aspects of interest.

The teacher effectively leverages seed words. By leveraging the seed words in a more direct way, Teacher is able to outperform the MATE-* models. Thus, we can use Teacher's predictions as supervision for the student, as we describe next.

The student outperforms the teacher. Student-BoW outperforms Teacher: the two models have the same architecture but Teacher only considers seed words; regularizing Student's weights encourages Student to mimic the noisy aspect predictions of Teacher by also considering non-seed words for aspect detection. The benefits of our distillation approach are highlighted using neural networks with word embeddings. Student-W2V outperforms both Teacher and Student-BoW, showing that obtaining segment representations as the average of word embeddings is more effective than using bag-of-words representations for this task.

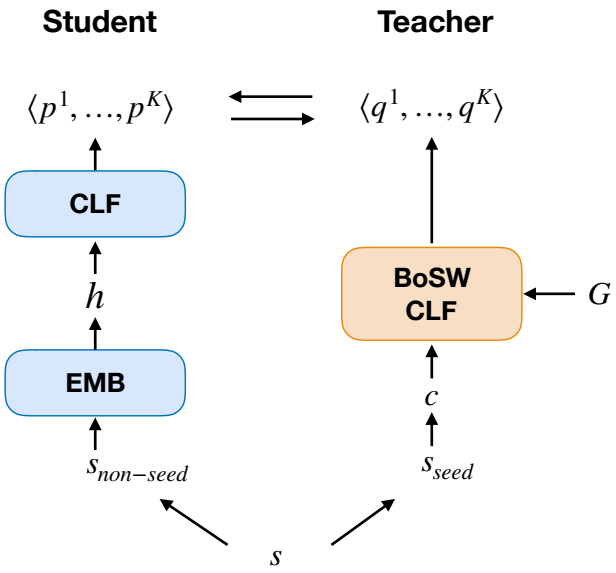


Figure 5.3: Our weakly supervised co-training approach when seed words are removed from the student's input (RSW baseline). Segment $s_{non-seed}$ is an edited version of s, where we replace each seed word in s by an "UNK" special token (like out-of-vocabulary words).

The student outperforms previous weakly supervised models even in one co–training round. Student-ATT outperforms MATE-unweighted (by 36.3% in product reviews and by 52.2% in restaurant reviews) even in a single co-training round: although the two models use exactly the same seed words (without weights), pre-trained word embeddings, EMB function, and CLF function, our student-teacher approach leverages the available seed words more effectively as noisy supervision than just for initialization. Also, using our approach, we can explore more powerful methods for segment embedding without the constraint of a fixed word embedding space. Indeed, using contextualized word representations in Student-BERT leads to the best performance over all models.

As expected, our weakly supervised approach does not outperform the fully supervised (*-Gold) models. However, our approach substantially reduces the performance gap between weakly supervised approaches and fully supervised approaches by 62%. The benefits of our student-teacher approach are consistent across all datasets, highlighting the predictive power of seed words across different domains and languages.

Method	Initial	Iterative
Product Reviews (AVG)		
MATE	46.4	-
Teacher / Student-W2V	52.2 / 58.7	**58.5** / **59.7**
Teacher / Student-BERT	52.2 / 60.2	**58.6** / **60.8**
Restaurant Reviews (En)		
MATE	29.5	-
Teacher / Student-W2V	44.9 / 47.2	**45.8** / **49.0**
Teacher / Student-BERT	44.9 / 51.8	**49.8** / **53.4**

Table 5.5: Micro-averaged F1 scores during the first round (middle column) and after iterative co-training (right column) in product reviews (top) and restaurant reviews (bottom).

The student leverages non-seed words. To better understand the extent to which non-seed words can predict the aspects of interest, we experiment with completely removing the seed words from Student-W2V's input during training (Student-W2V-RSW method; see Figure 5.3). Thus, in this setting, Student-W2V-RSW is forced to only use non-seed words to detect aspects. Note that the co-training assumption of conditionally independent views [Blum and Mitchell, 1998] is satisfied in this setting, where Teacher is only using seed words and Student-W2V is only using non-seed words. Student-W2V-RSW effectively learns to use non-seed words to predict aspects and performs better than Teacher (but worse than Student-W2V, which considers both seed and non-seed words).

Iterative co-training copes with noisy words. Further performance improvement in Teacher and Student-* can be observed with the iterative co-training procedure of Section 5.3.3. Table 5.5 reports the performance of Teacher and Student-* after co-training for both product reviews (top) and English restaurant reviews (bottom). Compared to the initial version of Teacher that does not model the quality of the seed words, iterative co-training leads to estimates of seed word quality that improve Teacher's performance up to 12.3% (in product reviews using Student-BERT).

A better teacher leads to a better student. Co-training leads to improved student performance in both datasets (Table 5.5). Compared to MATE, which uses the validation

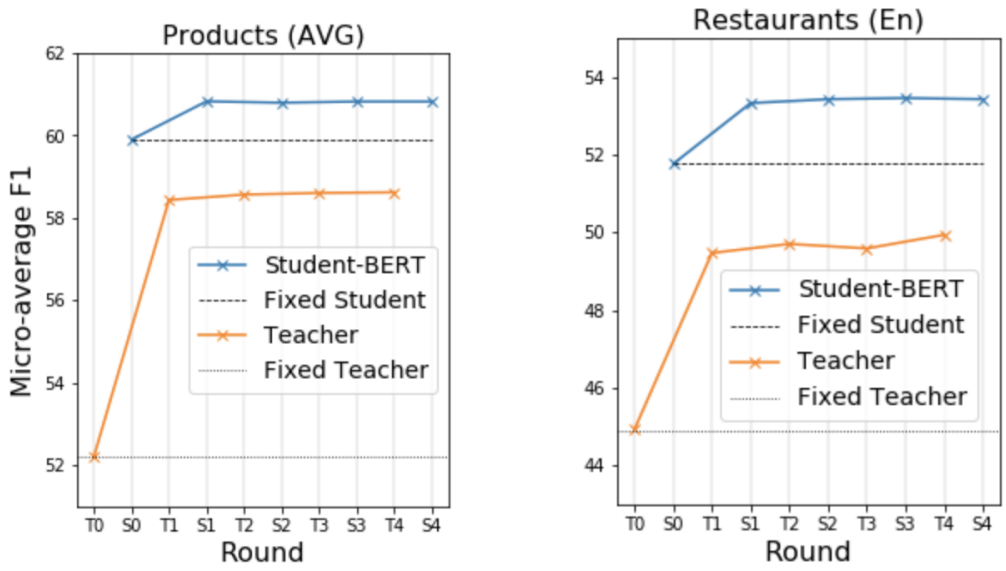


Figure 5.4: Co-training performance for each round reported for product reviews (left) and restaurant reviews (right). T<i>and S<i>correspond to the teacher's and student's performance, respectively, at the i-th round.

set to estimate the seed weights as a pre-processing step, we estimate and iteratively adapt the seed weights using the student-teacher disagreement, which substantially improves performance. Across the 12 datasets, Student-BERT leads to an average absolute increase of 14.1 F1 points.

Figure 5.4 plots Teacher's and Student-BERT's performance after each round of co-training. Most of the improvement for both Teacher and Student-BERT is gained in the first two rounds of co-training: "T0" (in Figure 5.4) is the initial teacher, while "T1" is the teacher with estimates of seed word qualities, which leads to more accurate predictions, e.g., in segments with multiple seed words from different aspects.

5.6 Using ISWD to Analyze COVID-19 Aspects of Restaurant Reviews

In this section, we present our analysis of the effects of COVID-19 on restaurant reviews, which led to revealing trends, such as increased mentions of hygienic practices of restaurants. As an important step for this analysis, we use our ISWD method to extract fine-grained

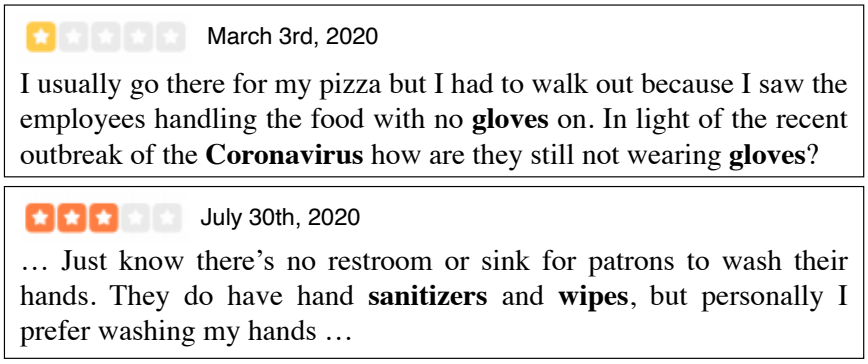


Figure 5.5: Examples of Yelp restaurant reviews discussing hygiene practices.

COVID-19 aspects related to restaurants (e.g., hygiene practices, sympathy and support, social distancing, etc.) from 3 million reviews. We further analyze the number and evolution of COVID-related aspects over time and show that the resulting time series have substantial correlation with critical statistics related to the COVID-19 pandemic, including the number of new COVID-19 cases.

The outbreak of the SARS-CoV-2 virus in December of 2019 and its evolution to the COVID-19 pandemic have had many devastating consequences in society. Restaurants have been among the hardest-hit businesses during the pandemic.[8] Yelp data (as of September 2020) shows that out of the 32,109 restaurant closures in the U.S., 61% have been permanent, and a greater impact is observed in local businesses in larger metropolitan areas, such as New York City (NYC) and Los Angeles County (LA), on which we focus in this work.

Restaurants operate under great uncertainty during this ongoing situation and, therefore, it is critical to understand how the pandemic has affected public attitude towards restaurants. The disruption in daily routines as well as fear and anxiety due to the pandemic have been shown to affect eating habits [Naja and Hamadeh, 2020; Di Renzo et al., 2020]. The pandemic may have also affected customers' preferences, such as changes in cuisine types, or higher expectations of hygiene and social distancing practices followed by restaurants.

In this work, we present our efforts to understand the effects of COVID-19 on restaurant

[8] `https://www.yelpeconomicaverage.com/business-closures-update-sep-2020.html`

	NYC	LA County
# Restaurants	55K	65K
# Users	344K	710K
# Reviews	1.0M	2.1M

Table 5.6: Statistics for our Yelp dataset of 3.1 million restaurant reviews collected during January 1, 2019 - December 31, 2020.

reviews. Reviewers provide ratings and free-form text to express their opinions and experiences about restaurants and we argue that the pandemic has affected such reviews. As an example, Figure 5.5 shows a Yelp review discussing the hygiene practices of a restaurant, including a mention of "coronavirus" and associated concerns.

To understand more broadly the effect of the pandemic on restaurant reviews, we analyze 3.1 million Yelp reviews published before and during the pandemic, for restaurants in two large metropolitan areas, namely, New York City and Los Angeles County. Table 5.6 shows more statistics for our dataset. In the rest of this section, we present our analysis of restaurant aspects related to COVID-19, their evolution through time, and their correlation with COVID-19 statistics. The material described in this section appears in [Cao et al., 2021].

Analysis of COVID-19 aspects in reviews. An important step to quantify changes in written text is to understand what aspects of restaurant operations are discussed in reviews referring to the pandemic. After reading 600 Yelp reviews posted after March 1, 2020, we identified the following main aspects of restaurants related to COVID-19:

1. Hygiene: hygiene conditions of restaurants and protective equipment (e.g., *"Just know there's no restroom or sink for patrons to wash their hands. They do have hand sanitizers and wipes, but personally I prefer washing my hands."*).

2. Transmission: concern of virus transmission (e.g., *"All the whole coughing without covering his mouth"*).

3. Social Distancing: social distancing measures (e.g., *"The tables are set far apart – a*

more than acceptable social distance").

4. Racism: racism experiences (e.g., *"She was the only one waiting at the register but no one came to ring her up. She waited for a while but decided to leave after realizing she was ignored because of her race."*).

5. Sympathy and Support: messages of solidarity, for example, towards local businesses (e.g., *"Help support your Chinatown restaurants who are deeply hurting from the stigma around corona virus."*).

6. Service: service changes during the pandemic (e.g., *"Not sure if the restaurant was empty because of the coronavirus scare but the food came out suuuuper fast..."*).

7. Other: aspects that are related to COVID but that do not fall under any of the above categories (e.g., *"Shame on management for taking advantage of people trying to keep safe from coronavirus during a NY state of emergency."*).

To automatically detect the presence of the above COVID-19 aspects across all 3 million reviews, we consider our ISWD method from Section 5.3. First, we manually define a small number of keywords or key phrases for each COVID-19 aspect and then, we use the keywords in a teacher to train a BERT-based student classifier on unlabeled reviews. We experimentally demonstrate that the student achieves better classification accuracy than the teacher on a set of reviews manually-labeled by us. Evaluation details are discussed in [Cao et al., 2021], while we now describe how COVID-19 aspect detection can help quantify the effects of COVID-19 on restaurant reviews.

Analysis of the evolution of restaurant review aspects over time. As one finding of our work, we show how the above analysis of COVID-19 aspects can help understand how reviews have changed during the pandemic. For a given aspect (e.g., Hygiene), we extract time series from the text of the reviews as the percentage of the reviews at each point in time that contain at least one aspect-specific keyword. Figure 5.6 shows the evolution of aspects

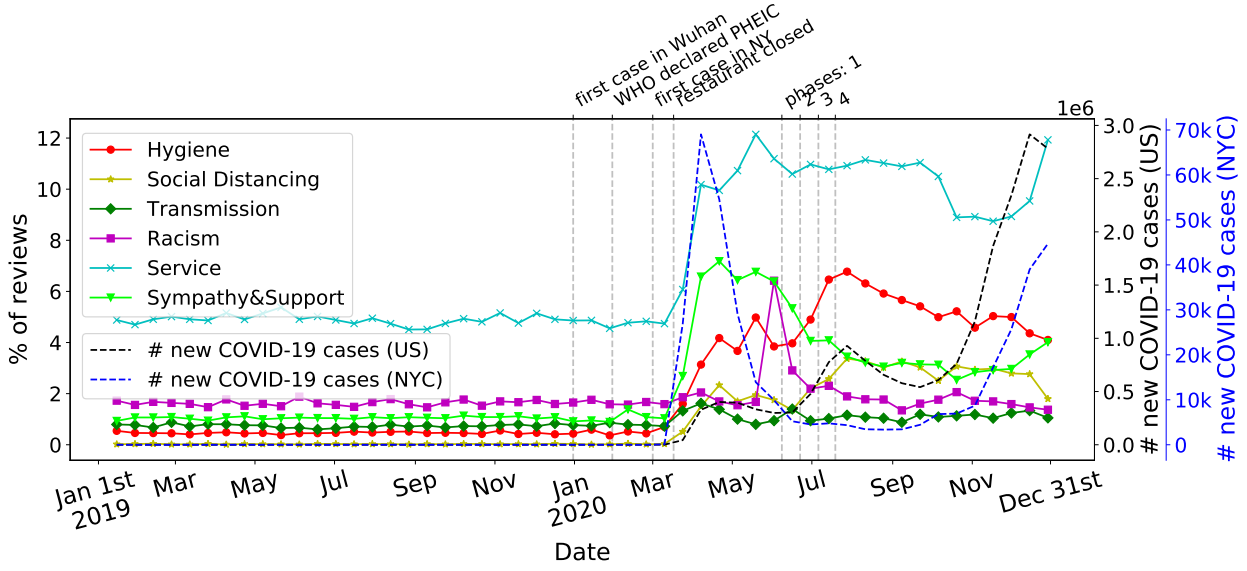


Figure 5.6: COVID aspects for NYC restaurants over January 1, 2019 - December 31, 2020.

over time for NYC. Aspects such as "Hygiene" and "Social Distancing" have been discussed more frequently after March 2020, covering up to 8% of the restaurant reviews: reviewers discuss such aspects during the pandemic more than before the pandemic. Interestingly, while "Hygiene" peaked during July 2020 (during restaurant re-opening) for both cities and since then keeps decreasing, "Sympathy & Support" peaked during Spring 2020, then decreased, and follows an increasing trend after November 2020.

Analysis of the correlation between restaurant aspects and critical COVID-19 statistics. As another finding of our work, we show that such time series extracted from the text of the reviews correlate with COVID-19 statistics. Table 5.7 reports the Spearman correlation between time series constructed for each COVID-19 aspect and the number of new COVID-19 cases. For both NYC and LA, there is significant correlation between restaurant review aspects and new cases of COVID-19, reaching up to Spearman's ρ=0.84 for the Hygiene aspect. For LA, COVID aspects have higher absolute correlation to the number of US cases compared to the number of LA cases. For NYC, most aspects present higher correlation with the number of NYC cases compared to the number of US cases. Even though

Time Series (NYC)	**NYC Cases**	**US Cases**
Social Distancing	0.768***	0.836***
Hygiene	0.765***	0.822***
Transmission	0.816***	0.804***
Sympathy & Support	0.822***	0.755***
Service	0.772***	0.736***
Racism	0.293**	0.237*
Time Series (LA)	**LA Cases**	**US Cases**
Service	0.536***	0.644***
Sympathy & Support	0.490***	0.551***
Hygiene	0.395***	0.538***
Transmission	0.409***	0.522***
Social Distancing	0.347**	0.513***
Racism	-0.006	-0.019

Table 5.7: Spearman correlation results from comparing COVID aspects and the number of COVID cases in NYC (top) and LA (bottom), sorted in decreasing order by correlation compared with the number of new US cases. Results are marked as statistically significant at the $p<0.1$*, $p<0.05$**, and $p<0.01$*** levels.

we cannot draw causal conclusions from these correlations, our results highlight interesting trends of Yelp reviews during the pandemic.

In addition to the above findings of restaurant review changes during the pandemic, in [Cao et al., 2021] we present additional findings such as increased interest in fast food restaurants compared to traditional American-food restaurants (including brunch restaurants). Our findings may provide useful insights for restaurant owners, customers, public health officials, and the broad research community.

5.7 Conclusions

In this chapter, we presented a weakly supervised approach for leveraging a small number of seed words (instead of ground truth aspect labels) for segment classification. We summarize the contributions of this chapter as follows: (i) we showed how to leverage the predictive power of seed words as weak supervision through our teacher model that considers each individual seed word as a (noisy) aspect indicator (Section 5.3.1); (ii) we presented a

technique that uses the seed-word based teacher to train an architecture-agnostic student classifier that leverages both seed words and their rich context in unlabeled segments (Section 5.3.2); (iii) we showed how iterative co-training can be used to cope with noisy seed words: the teacher effectively estimates the predictive quality of the noisy seed words in an unsupervised manner using the associated predictions by the student (Section 5.3.3); (iv) we showed the advantages of our ideas by performing an extensive experimental evaluation on fine-grained aspect detection of restaurant and product reviews (Sections 5.4 and 5.5); and (iv) we applied our teacher-student method for a new application, the analysis of the effects of COVID-19 on restaurant reviews.

Our findings show that our student-teacher approach leverages seed words more directly and effectively than previous weakly supervised approaches. The teacher model provides weak supervision to a student model, which we showed generalizes better than the teacher by also considering non-seed words and by using pre-trained word embeddings. We further showed that iterative co-training leads to a better teacher and, in turn, a better student. Our proposed method consistently outperforms previous weakly supervised methods across al 12 datasets, allowing for seed words from various domains and languages to be leveraged for aspect detection. Our student-teacher approach could be applied for any classification task for which a small set of seed words describe each class. By applying ISWD for the analysis of COVID-19 aspects, we showed revealing trends, such as increased mentions of hygienic practices of restaurants, which could potentially inform policies by public health departments, for example, to cover resource utilization. In the next chapter, we will present a new method for transferring supervision across languages, in which ISWD is one important component to reduce the cross-lingual resources needed for effective cross-lingual transfer.

Chapter 6: Cross-Lingual Transfer of Weak Supervision with Minimal Resources

In Chapter 5, we presented a weakly-supervised co-training framework for training classifiers using seed words and demonstrated its successful application for several domains and tasks. In this chapter, we show how this weakly-supervised co-training approach can help applications of document classification beyond English without additional supervision in non-English languages. Instead, we present a method for transferring supervision across languages using minimal cross-lingual resources in the form of bilingual word translations. First, we provide an overview and motivation of cross-lingual transfer with limited resources (Section 6.1). Second, we discuss related work and define our problem of focus (Section 6.2). Third, we present our cross-lingual teacher-student method framework, CLTS, which transfers seed words across languages (Section 6.3). Then, we present our experimental evaluation for document classification across 18 diverse languages (Sections 6.4 and 6.5) and describe the application of CLTS for additional classification problems (Section 6.6). Finally, we summarize the contributions of this chapter (Section 6.7).

6.1 Overview and Motivation

The main bottleneck in using supervised learning for multilingual document classification is the high cost of obtaining labeled documents for all of the target languages. To address this issue in a target language L_T, we consider a cross-lingual text classification approach that requires labeled documents only in a source language L_S and not in L_T.

Existing approaches for transferring supervision across languages rely on large parallel corpora or machine translation systems, which are expensive to obtain and are not available

for many languages.[1] To scale beyond high-resource languages, multilingual systems have to reduce the cross-lingual requirements and operate under a limited budget of cross-lingual resources. Such systems typically ignore target-language supervision, and rely on feature representations that bridge languages, such as cross-lingual word embeddings [Ruder et al., 2019] or multilingual transformer models [Wu and Dredze, 2019; Pires et al., 2019]. This general approach is less expensive but has a key limitation: by not considering labeled documents in L_T, it may fail to capture predictive patterns that are specific to L_T. Its performance is thus sensitive to the quality of pre-aligned features [Glavaš et al., 2019].

In this work, we show how to obtain weak supervision for training accurate classifiers in L_T without using manually labeled documents in L_T or expensive document translations. We propose a novel approach for cross-lingual text classification that transfers weak supervision from L_S to L_T using *minimal* cross-lingual resources: we only require a small number of task-specific keywords, or seed words, to be translated from L_S to L_T.

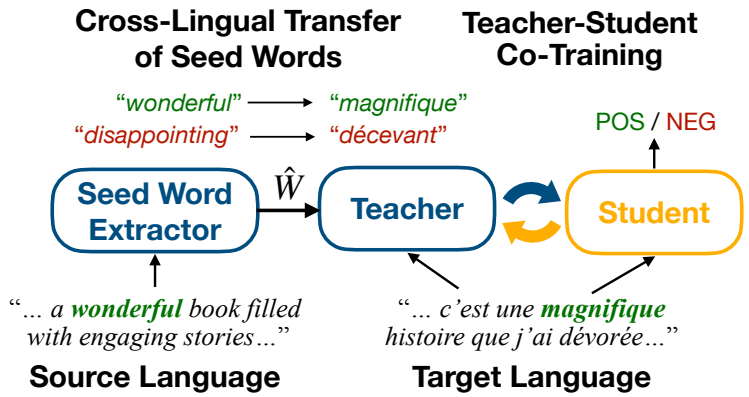


Figure 6.1: Our cross-lingual teacher-student (CLTS) method trains a student classifier in the target language by transferring weak supervision across languages.

Our core idea is that the most indicative seed words in L_S often translate to words that are also indicative in L_T. For instance, the word "wonderful" in English indicates positive sentiment, and so does its translation "magnifique" in French. Thus, given a limited budget for word translations (e.g., from a bilingual speaker), only the most important seed words

[1] As of July 2022, Google Translate (`https://translate.google.com/`) is available for 133 out of the about 4,000 written languages (`https://www.ethnologue.com/`).

should be prioritized to transfer task-specific information from L_S to L_T.

Having access only to limited cross-lingual resources creates important challenges, which we address with a novel cross-lingual teacher-student method, CLTS, which extends the monolingual seed word distillation method from Chapter 5 to effectively transfer seed words across languages. Our work presents the following contributions:

Efficient transfer of supervision across languages. As a first contribution, we present a method for cross-lingual transfer in low-resource settings with a limited word translation budget. CLTS extracts the most important seed words using the translation budget as a sparsity-inducing regularizer when training a classifier in L_S. Then, it transfers seed words and the classifier's weights across languages, and initializes a teacher classifier in L_T that uses the translated seed words.

Effective training of classifiers without using any labeled target documents. The teacher, as described above, predicts meaningful probabilities only for documents that contain translated seed words. As a second contribution, we effectively apply our weakly-supervised co-training approach from Chapter 5 to this cross-lingual setting. Because translations can induce errors and the translation budget is limited, the translated seed words may be noisy and not comprehensive for the task at hand. By extending the monolingual teacher-student approach from Chapter 5 to our setting, we train a student that outperforms the teacher across all languages by 59.6%.

Robust performance across languages and tasks. As a third contribution, we empirically show the benefits of generating weak supervision in 18 diverse languages and 4 document classification tasks. With as few as 20 seed-word translations and a bag-of-words logistic regression student, CLTS outperforms state-of-the-art methods relying on more complex multilingual models, such as multilingual BERT, across most languages. Using a monolingual BERT student leads to further improvements and outperforms even more expensive

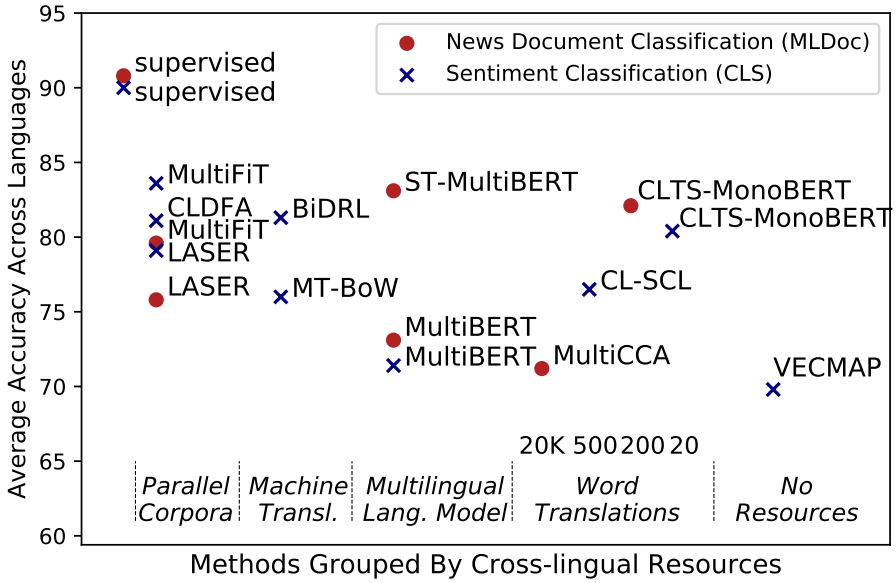


Figure 6.2: CLTS leverages a small number of word translations more effectively than previous methods and sometimes outperforms more expensive methods.

approaches (Figure 6.2). CLTS does not require cross-lingual resources such as parallel corpora, machine translation systems, or pre-trained multilingual language models, which makes it applicable in low-resource settings. As a preliminary exploration, we address medical emergency situation detection in Uyghur and Sinhalese with just 50 translated seed words per language, which could be easily obtained from bilingual speakers.

We start with a review of the relevant work on cross-lingual text classification (Section 6.2). We continue as follows:

- We develop CLTS, a method for training document classifiers across languages using labeled documents only in English and a limited budget for bilingual translations (Section 6.3).[2]
- We evaluate our ideas by conducting an experimental evaluation on document classification in 18 diverse languages, including low-resource languages (Sections 6.4 and 6.5).
- We present additional applications of cross-lingual learning (Section 6.6).

[2]Our Python implementation is publicly available at `https://github.com/gkaramanolakis/clts`.

Finally, we discuss the implications of our work (Section 6.7). The material described in this chapter appears in [Karamanolakis et al., 2020a; Liu et al., 2020].

6.2 Related Work and Problem Definition

We focus on a cross-lingual text classification scenario with labeled data in the source language L_S and unlabeled data in the target language L_T. We review the different types of required cross-lingual resources, starting with the most expensive types.

Annotation projection and machine translation. With parallel corpora (i.e., corpora where each document is written in both L_S and L_T), a classifier trained in L_S predicts labels for documents in L_S and its predictions are projected to documents in L_T to train a classifier in L_T [Mihalcea et al., 2007; Rasooli et al., 2018]. Unfortunately, parallel corpora are hard to find, especially in low-resource domains and languages.

Without parallel corpora, documents can be translated using machine translation (MT) systems [Wan, 2008; Wan, 2009; Salameh et al., 2015; Mohammad et al., 2016]. However, high-quality MT systems are limited to high-resource languages. Even when an MT system is available, translations may change document semantics and degrade classification accuracy [Duh et al., 2011; Salameh et al., 2015; Rasooli et al., 2018]. To avoid the "domain gap" introduced by MT, [Rasooli et al., 2018] use parallel or comparable data to create bilingual word dictionaries and translate just the words with available entries. Similarly to [Rasooli et al., 2018], our method does not require MT but we do not require parallel or comparable data. Instead, we require translations for just a small number of *task-specific* words that our method identifies automatically using source labeled data and a limited budget for word translations.

Cross-lingual representation learning. Other approaches rely on less expensive resources to align feature representations across languages, typically in a shared feature space

to enable cross-lingual model transfer.

Cross-lingual word embeddings, or CLWE, represent words from different languages in a joint embedding space, where words with similar meanings obtain similar vectors regardless of their language. (See [Ruder et al., 2019] for a survey.) Early CLWE approaches required expensive parallel data [Klementiev et al., 2012; Täckström et al., 2012]. In contrast, later approaches rely on high-coverage bilingual dictionaries [Gliozzo and Strapparava, 2006; Faruqui and Dyer, 2014; Gouws et al., 2015; Rasooli et al., 2018] or smaller "seed" dictionaries [Gouws and Søgaard, 2015; Artetxe et al., 2017]. Some recent CLWE approaches require no cross-lingual resources [Lample et al., 2018; Artetxe et al., 2018; Chen and Cardie, 2018; Søgaard et al., 2018] but perform substantially worse than approaches using seed dictionaries of 500-1,000 pairs [Vulić et al., 2019]. Our approach does not require CLWE and achieves competitive classification performance with substantially fewer translations of *task-specific* words.

Recently, multilingual transformer models were pre-trained in multiple languages in parallel using language modeling objectives [Devlin et al., 2019; Conneau and Lample, 2019]. Multilingual BERT, a version of BERT [Devlin et al., 2019] that was trained on 104 languages in parallel without using any cross-lingual resources, has received significant attention [Karthikeyan et al., 2019; Singh et al., 2019; Rogers et al., 2020]. Multilingual BERT performs well on zero-shot cross-lingual transfer [Wu and Dredze, 2019; Pires et al., 2019] and its performance can be further improved by considering target-language documents through self-training [Dong and de Melo, 2019]. In contrast, our approach does not require multilingual language models and sometimes outperforms multilingual BERT using a *monolingual* BERT student.

Knowledge distillation for cross-lingual classification. Our teacher-student approach is similar to other knowledge distillation approaches for cross-lingual classification. [Xu and Yang, 2017] apply knowledge distillation for cross-lingual text classification but require

expensive parallel corpora. MultiFiT [Eisenschlos et al., 2019] trains a classifier in L_T using the predictions of a cross-lingual model, namely, LASER [Artetxe and Schwenk, 2019], that also requires large parallel corpora. [Vyas and Carpuat, 2019] classify the semantic relation (e.g., synonymy) between two words from different languages by transferring *all* training examples across languages. Our approach addresses a different problem, where training examples are full documents (not words), and transferring source training documents would require MT. Related to distillation is the semi-supervised approach of [Shi et al., 2010] that trains a target classifier by transferring a source classifier using high-coverage dictionaries. Our approach is similar, but trains a classifier using sparsity regularization, and translates only the most important seed words.

Problem definition. Consider a source language L_S, a target language L_T, and a classification task with K predefined classes of interest $\mathcal{Y} = \{1, \ldots, K\}$ (e.g., sentiment categories). Labeled documents $D_S = \{(s_i^S, y_i)\}_{i=1}^N$ are available in L_S, where $y_i \in \mathcal{Y}$ and each source document s_i^S is a sequence of words from the source vocabulary V_S. Only unlabeled documents $D_T = \{s_i^T\}_{i=1}^M$ are available in L_T, where each target document s_i^T is a sequence of words from the target vocabulary V_T. We assume that there is no significant shift in the conditional distribution of labels given documents across languages. Furthermore, we assume a limited translation budget, so that up to B words can be translated from L_S to L_T.

Our goal is to use the labeled source documents D_S, the unlabeled target documents D_T, and the translations of no more than B source words to train a classifier that, given an unseen test document s_i^T in the target language L_T, predicts the corresponding label $y_i \in \mathcal{Y}$.

6.3 Cross-Lingual Teacher-Student (CLTS)

We now describe our cross-lingual teacher-student method, CLTS, for cross-lingual text classification. Given a limited budget of B translations, CLTS extracts only the B most important seed words in L_S (Section 6.3.1). Then, CLTS transfers the seed words and their

weights from L_S to L_T, to initialize a classifier in L_T (Section 6.3.2). Using this classifier as a teacher, CLTS trains a student that predicts labels using both seed words and their context in target documents (Section 6.3.3).

6.3.1 Seed-Word Extraction in L_S

CLTS starts by automatically extracting a set G_k^S of indicative seed words per class k in L_S. Previous extraction approaches, such as tf-idf variants [Angelidis and Lapata, 2018b], have been effective in monolingual settings with limited labeled data. In Chapter 5, we exploited these seed words to efficiently train monolingual classifiers. In our cross-lingual scenario, with many labeled *source* documents and a limited translation budget B, we propose a different approach based on a supervised classifier trained with sparsity regularization.

Specifically, CLTS extracts seed words from the weights $\mathbf{W} \in \mathbb{R}^{K \times |V_S|}$ of a classifier trained using D_S. Given a source document s_i^S with a bag-of-words encoding $\mathbf{h}_i^S \in \mathbb{R}^{|V_S|}$, the classifier predicts class probabilities $\mathbf{p}_i = (p_i^1, \ldots, p_i^K) = \text{softmax}(\mathbf{W}\mathbf{h}_i)$. CLTS includes the word $v_c \in V_S$ in G_k^S if the classifier considers it to increase the probability p_i^k through a positive weight $\mathbf{W}_{kc}$:

$$G_k^S = \{v_c^S \mid \mathbf{W}_{kc} > 0\}. \tag{6.1}$$

The set of all source seed words $G^S = G_1^S \cup \cdots \cup G_K^S$ may be much larger than the translation budget B. We encourage the classifier to capture only the most important seed words *during* training through sparsity regularization:

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_{i=1}^{N} \mathcal{L}(y_i, \mathbf{W}\mathbf{h}_i^S) + \lambda_B \mathcal{R}_{\text{sparse}}(\mathbf{W}) \tag{6.2}$$

where $\mathcal{L}$ is the training loss function (logistic loss), $\mathcal{R}_{\text{sparse}}(.)$ is a sparsity regularizer (L1 norm), and $\lambda_B \in \mathbb{R}$ is a hyperparameter controlling the relative power of $\mathcal{R}_{\text{sparse}}$. Higher λ_B values lead to sparser matrices $\hat{\mathbf{W}}$ and thus to fewer seed words. Therefore, we tune[3] λ_B to

[3]We efficiently tune λ_B by computing the "regularization path" with the "warm-start" technique [Koh

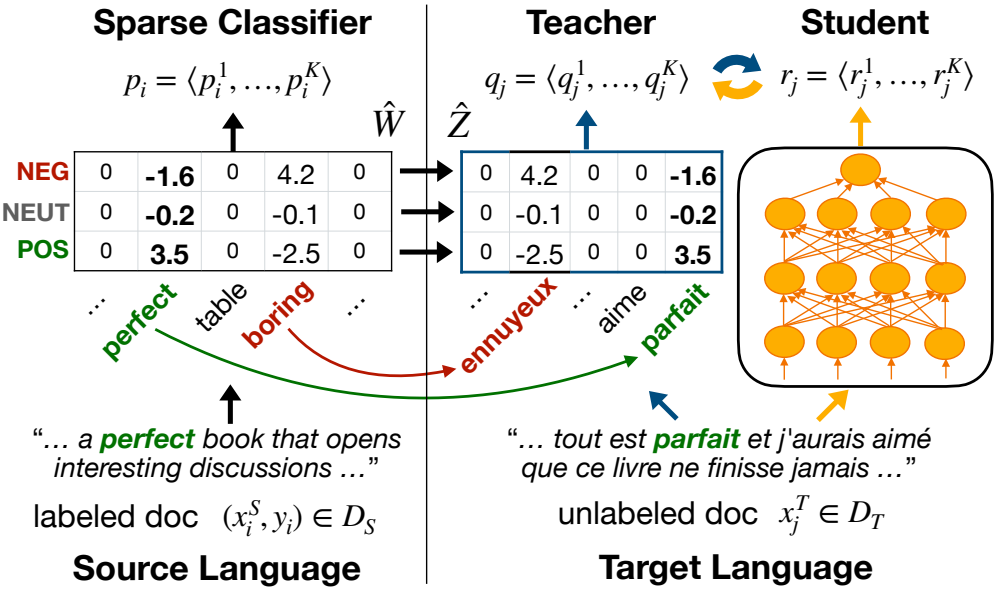


Figure 6.3: CLTS (1) learns a sparse weight matrix $\hat{\mathbf{W}}$ in L_S; (2) transfers the columns of $\hat{\mathbf{W}}$ for B seed words to initialize $\hat{\mathbf{Z}}$; and (3) uses $\hat{\mathbf{Z}}$ as a teacher to iteratively train a student on unlabeled documents D_T.

be as high as possible while at the same time leading to the extraction of at least B seed words. After training, G^S consists of the B seed words with highest weight.

6.3.2 Cross-Lingual Seed Weight Transfer

We now describe our cross-lingual transfer method. CLTS transfers both translated seed words and their learned weights to initialize a "weak" classifier in L_T that considers translated seed words and their relative importance for the target task.

Specifically, CLTS first translates the B seed words in G^S into a set G^T with seed words in L_T. Then, for each translation pair (v^S, v^T), CLTS transfers the column for v^S in $\hat{\mathbf{W}}$ to a corresponding column for v^T in a $K \times |V_T|$ matrix $\hat{\mathbf{Z}}$:

$$\hat{\mathbf{Z}}_{k,v^S} = \hat{\mathbf{W}}_{k,v^T} \quad \forall k \in [K] \tag{6.3}$$

Importantly, for each word, we transfer the weights for all classes (instead of just a single

et al., 2007].

weight $\hat{\mathbf{W}}_{kc}$) across languages. Therefore, *without using any labeled documents* in L_T, CLTS constructs a classifier that, given a test document s_j^T in L_T, predicts class probabilities $\mathbf{q}_j = (q_j^1, \ldots, q_j^K)$:

$$q_j^k = \frac{\exp(\hat{z}_k^\top \mathbf{h}_j^T)}{\sum_{k'} \exp(\hat{\mathbf{z}}_{k'}^\top \mathbf{h}_j^T)}, \tag{6.4}$$

where $\mathbf{h}_j^T \in \mathbb{R}^{|V_T|}$ is a bag-of-words encoding for s_j^T and $\hat{\mathbf{z}}_k$ is the k-th row of $\hat{\mathbf{Z}}$. Note that columns of $\hat{\mathbf{Z}}$ for non-seed words in V_T are all zeros and thus this classifier predicts meaningful probabilities only for documents with seed words in G^T.

6.3.3 Teacher-Student Co-Training in L_T

We now describe how CLTS trains a classifier in L_T that leverages indicative features, which may not be captured by the small set of translated seed words. As illustrated in Figure 6.3, translated seed words (e.g., "parfait") often co-occur with other words (e.g., "aime," meaning "love") that have zero weight in $\hat{\mathbf{Z}}$ but are also helpful for the task at hand. To exploit such words in the absence of labeled target documents, we extend our monolingual ISWD method from Section 5.3 to our cross-lingual setting, and use our classifier based on translated seed words as a teacher to train a student, as we describe next.

First, CLTS uses our classifier from Equation 6.5 as a teacher to predict labels $\mathbf{q}_j$ for *unlabeled* documents $s_j^T \in D_T$ that contain seed words: $D'_T = \{(s_j^T, \mathbf{q}_j)\}_{s_j^T | s_j^T \cap G^T \neq \varnothing} \subseteq D_T$. Note that our teacher with weights transferred across languages is different than that of ISWD, which simply "counts" seed words.

Next, CLTS trains a student f^T that also exploits the context of the seed words. Given a document s_j^T in L_T, the student predicts class probabilities:

$$r_j = (r_j^1, \ldots, r_j^K) = f^T(s_j^T; \theta), \tag{6.5}$$

where the predictor function f^T with weight parameters θ can be of any type, such as a pre-trained transformer-based classifier that captures language-specific word composition.

Algorithm 1 Cross-Lingual Teacher-Student

Input: Unlabeled documents $D_T = \{s_j^T\}_{j=1}^M$, labeled documents $D_S = \{(s_i^S, y_i)\}_{i=1}^N$, budget of up to B word translations (L_S to L_T)

Output: $\hat{f}^T$: predictor function in L_T

1: Learn λ_B-sparse $\hat{\mathbf{W}}$ using D_S, B (Eq. (6.2))
2: Extract B seed words G^S from $\hat{\mathbf{W}}$ (Eq. (6.1))
3: Translate G^S to target seed words G^T in L_T
4: Transfer $\hat{\mathbf{W}}$ to initialize teacher $\hat{\mathbf{Z}}$ (Eq. (6.3))
5: Get $D'_T = \{(s_j^T, \mathbf{q}_j)\}_{s_j^T | s_j^T \cap G^T \neq \varnothing}$ (Eq. (6.4))
6: **Repeat until convergence**
   a. Learn student $\hat{f}^T$ using D'_T (Eq. (6.6))
   b. Get $D'_T = \{(s_j^T, \hat{f}^T(s_j^T)\}_{j \in [M]}$ (Eq. (6.5))

The student is trained via the distillation objective (see Section 5.2.3):

$$\hat{\theta} = \arg\min_{\theta} \sum_{(s_j^T, \mathbf{q}_j) \in D'_T} H(\mathbf{q}_j, f^T(s_j^T)) + \lambda \mathcal{R}(\theta), \tag{6.6}$$

where $H(q, r) = -\sum_k \mathbf{q}^k \log r^k$ is the cross entropy between student's and teacher's predictions, $\mathcal{R}(.)$ is a regularizer (L2 norm), and $\lambda \in \mathbb{R}$ is a hyperparameter controlling the relative power of $\mathcal{R}$. Importantly, through extra regularization ($\mathcal{R}$, dropout) the student also associates non-seed words with target classes, and generalizes better than the teacher by making predictions even for documents that do not contain any seed words.

Then, CLTS uses the student in place of the teacher to annotate *all* M unlabeled examples in D_T and create $D'_T = \{(s_j^T, \hat{f}^T(s_j^T)\}_{j \in [M])}$. While in the first iteration D'_T contains only documents with seed words, in the second iteration CLTS adds in D'_T *all* unlabeled documents to create a larger training set for the student. This also differs from ISWD, which updates the weights of the initial seed words but does not provide pseudo-labels for documents with no seed words. This change is important in our cross-lingual setting with a limited translation budget, where the translated seed words G^T may only cover a very small subset D'_T of D_T.

Algorithm 1 summarizes the CLTS method for cross-lingual classification by translating B seed words. Iterative co-training converges when the disagreement between the student's and teacher's hard predictions on unlabeled data stops decreasing. In our experiments, just two rounds of co-training are generally sufficient for the student to outperform the teacher and achieve competitive performance even with a tight translation budget B.

We now turn into the empirical evaluation of CLTS and its comparison with state-of-the-art approaches for cross-lingual text classification tasks (Sections 6.4 and 6.5). As we will show, CLTS effectively transfers seed words across languages and outperforms approaches that use similar or even more expensive cross-lingual resources. Also, in Section 6.6, we present additional applications of cross-lingual transfer.

6.4 Experimental Settings

We now describe our experimental settings for several cross-lingual text classification tasks in various languages. We describe our four classification tasks, implementation details for each component in CLTS (seed word extraction in L_S, seed word transfer, and teacher-student co-training in L_T), and the comparison of CLTS with other models.

We use English (En) as a source language, and evaluate CLTS on 18 diverse target languages: Bulgarian (Bg), German (De), Spanish (Es), Persian (Fa), French (Fr), Croatian (Hr), Hungarian (Hu), Italian (It), Japanese (Ja), Polish (Pl), Portuguese (Pt), Russian (Ru), Sinhalese (Si), Slovak (Sk), Slovenian (Sl), Swedish (Sv), Uyghur (Ug), and Chinese (Zh). We focus on four classification tasks: **T1:** 4-class classification of news documents in the MLDoc corpus [Schwenk and Li, 2018]; **T2:** binary sentiment classification of product reviews in the CLS corpus [Prettenhofer and Stein, 2010]; **T3:** 3-class sentiment classification of tweets in the Twitter Sentiment corpus (TwitterSent; [Mozetič et al., 2016]), Persian reviews in the SentiPers corpus [Hosseini et al., 2018], and Uyghur documents in the LDC LORELEI corpus [Strassel and Tracey, 2016]; and **T4:** medical emergency situation detection in Uyghur and Sinhalese documents from the LDC LORELEI corpus.

Document classification in MLDoc. The Multilingual Document Classification Corpus (MLDoc[4]; [Schwenk and Li, 2018]) contains Reuters news documents in English, German, Spanish, French, Italian, Russian, Chinese, and Japanese. Each document is labeled with one of the four categories:

- CCAT (Corporate/Industrial)
- ECAT (Economics)
- GCAT (Government/Social)
- MCAT (Markets)

MLDoc was pre-processed and split by [Schwenk and Li, 2018] into 1,000 training, 1,000 validation, and 4,000 test documents for each language. We use labeled training documents only in English for training the source classifier. We treat training documents in German, Spanish, French, Italian, Russian, Chinese, and Japanese as unlabeled in CLTS by ignoring the labels.

Review sentiment classification in CLS. The Cross-Lingual Sentiment corpus (CLS[5]; [Prettenhofer and Stein, 2010]) contains Amazon product reviews in English, German, French, and Japanese. Each language includes product reviews from three domains: books, dvd, and music. Each labeled document includes a binary (positive, negative) sentiment label. Validation sets are not available for CLS. We use labeled training documents only in English for training the source classifier. We ignore training documents in German, French, and Japanese, and use unlabeled documents in CLTS.

Sentiment classification in TwitterSent, Sentipers, and LORELEI. The Twitter Sentiment corpus (TwitterSent; [Mozetič et al., 2016]) contains Twitter posts in Bulgarian

[4] `https://github.com/facebookresearch/MLDoc`

[5] `https://webis.de/data/webis-cls-10.html`

(Bg), German (De), English (En), Spanish (Es), Croatian (Hr), Hungarian (Hu), Polish (Pl), Portuguese (Pt), Slovak (Sk), Slovenian (Sl), and Swedish (Sv). We use the pre-processed and tokenized data provided by [Rasooli et al., 2018]. In addition to these tweets, [Rasooli et al., 2018] also use pre-processed and tokenized Persian (Fa) product reviews from the SentiPers corpus [Hosseini et al., 2018] and manually labeled Uyghur (Ug) documents from the LDC LORELEI corpus. On the above datasets, each document is labeled with a sentiment label: positive, neutral, or negative. We use labeled training documents only in English for training the source classifier. We treat training documents in the rest of the languages as unlabeled.

Experimental procedure. We use English as the source language, where we train a source classifier and extract B seed words using labeled documents (Section 6.3.1). Then, we obtain translations for $B \leq 500$ English seed words using the MUSE[6] bilingual dictionaries [Lample et al., 2018]. We do not use labeled documents in the target language for training (Section 5.2.4). We report both the teacher's and student's performance in L_T averaged over 5 different runs. We consider any test document that contains no seed words as a "mistake" for the teacher.

Configuration for source seed word extraction. The inputs to the classifier in L_S are tf-idf weighted unigram vectors[7]. For the classifier, we use scikit-learn's logistic regression[8] with the following parameters: penalty="l1", C=λ_B, solver="liblinear", multi_class="ovr". In other words, we address multi-class classification by training K binary "one-vs.-rest" logistic regression classifiers to minimize the $L1$-regularized logistic loss (LASSO). (We use scikit-learn version 0.22.1, which does not support a "multinomial" loss with L1-penalized classifiers.) We tune λ_B by computing the "regularization path" between 0.1 and 10^7, evenly

[6] `https://github.com/facebookresearch/MUSE#ground-truth-bilingual-dictionaries`

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`

[8] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

spaced on a log scale into 50 steps. To efficiently[9] compute the regularization path, we use the "warm-start" technique [Koh et al., 2007], where the solution of the previous optimization step is used to initialize the solution for the next one. This is supported in scikit-learn by setting the warm_start parameter of logistic regression to True.

Configuration for seed word transfer. We obtain seed-word translations using the MUSE[10] bilingual dictionaries [Lample et al., 2018], which contain up to 100,000 dictionary entries per language pair. Importantly, we use only the translations for $B \leq 500$ English seed words. To understand the impact of translation budget in performance, we experiment with the following values for $\frac{B}{K}$: [2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200]. We leave for future work the non-uniform distribution of seed words across classes, which might improve efficiency as "easier" classes may be modeled with fewer seed words.

If a source word has multiple translations in MUSE,[11] we use all translations as noisy target seed words with the same weight, while if a seed word has no translation in the target language, then we directly use it as a target seed word (this may be useful for named entities, emojis, etc.). Translations provided by a human annotator would possibly lead to better target seed words but, as we show here, even noisy automatic translations can be effectively used in CLTS.

Model comparison. For a robust evaluation of CLTS, we compare models with different types of cross-lingual resources:

- ***Project-**** uses the parallel LDC or EuroParl (EP) corpora for annotation projection [Rasooli et al., 2018].

[9] Using a 16-core CPU machine, we compute λ_B and train the source classifier in less than one minute.

[10] `https://github.com/facebookresearch/MUSE#ground-truth-bilingual-dictionaries`

[11] Various translations for a word in MUSE may correspond to different senses of the word. For example, the seed word "shares" for the "Corporate" topic translates to both "comparte" (share) and "acciones" (stocks) in Spanish.

- ***LASER*** uses millions of parallel corpora to obtain cross-lingual sentence embeddings [Artetxe and Schwenk, 2019].

- ***MultiFiT*** uses ***LASER*** to create pseudo-labels in L_T [Eisenschlos et al., 2019] and trains a classifier in L_T based on a pre-trained language model [Howard and Ruder, 2018].

- ***CLWE-par*** uses parallel corpora to train CLWE [Rasooli et al., 2018].

- ***MT-BOW*** uses Google Translate to translate test documents from L_T to L_S and applies a bag-of-words classifier in L_S [Prettenhofer and Stein, 2010].

- ***BiDRL*** uses Google Translate to translate documents from L_S to L_T and L_T to L_S [Zhou et al., 2016].

- ***CLDFA*** uses task-specific parallel corpora for cross-lingual distillation [Xu and Yang, 2017].

- ***SentiWordNet*** uses bilingual dictionaries with over 20K entries to transfer the SentiWordNet03 [Baccianella et al., 2010] to the target language and applies a rule-based heuristic [Rasooli et al., 2018].

- ***CLWE-Wikt*** uses bilingual dictionaries with over 20K entries extracted from Wiktionary[12] to create CLWE for training a bi-directional LSTM classifier [Rasooli et al., 2018].

- ***MultiCCA*** uses bilingual dictionaries with around 20K entries to train CLWE [Ammar et al., 2016], trains a convolutional neural network (CNN) in L_S and applies it in L_T [Schwenk and Li, 2018].

- ***CL-SCL*** obtains 450 word translations as "pivots" for cross-lingual domain adaptation [Prettenhofer and Stein, 2010].

[12] `https://www.wiktionary.org/`

Method	De	Es	Fr	It	Ru	Zh	Ja	AVG
Methods below use parallel corpora (MultiFiT requires LASER)								
LASER	87.7	**79.3**	84.0	71.2	67.3	76.7	64.6	75.8
MultiFiT	**91.6**	79.1	**89.4**	**76.0**	**67.8**	**82.5**	**69.6**	**79.4**
Methods below use pre-trained multi-lingual language models								
MultiBERT	79.8	72.1	73.5	63.7	73.7	76.0	72.8	73.1
ST-MultiBERT	**90.0**	**85.3**	**88.4**	**75.2**	**79.3**	**87.0**	**76.8**	**83.1**
Methods below use bilingual dictionaries (Student requires Teacher)								
MultiCCA (B=20K)	81.2	72.5	72.4	69.4	60.8	74.7	67.6	71.2
Teacher (B=160)	72.7	73.5	77.6	62.5	46.9	53.3	31.9	59.8
Student-LogReg	87.4	86.0	89.1	70.5	71.9	82.4	68.8	79.4
Student-MonoBERT	**90.4**	**86.3**	**91.2**	**74.7**	**75.6**	**84.0**	**72.6**	**82.1**

Table 6.1: Accuracy results on MLDoc.

- Our ***CLTS*** approach uses B word translations not for domain adaptation but to create weak supervision in L_T through the teacher (Teacher) for training the student (Student-LogReg or Student-MonoBERT).

- ***VECMAP*** uses identical strings across languages as a weak signal to train CLWE [Artetxe et al., 2017].

- ***MultiBERT*** uses multilingual BERT to train a classifier in L_S and applies it in L_T [Wu and Dredze, 2019] without considering labeled documents in L_T (zero-shot setting).

- ***ST-MultiBERT*** further considers labeled documents in L_T for fine-tuning multilingual BERT through self-training [Dong and de Melo, 2019].

6.5 Experimental Results Across 18 Languages

Tables 6.1, 6.2, and 6.3 show results for each classification task and language. The rightmost column of each table reports the average performance across all languages (and domains for CLS). For brevity, we report the average performance across the three review domains (Books, DVD, Music) for each language in the CLS corpus.

Model	**De**	**Fr**	**Ja**	**AVG**
Methods below use parallel corpora or MT				
MT-BOW	78.3	78.5	71.2	76.0
BiDRL	84.3	83.5	76.2	81.3
CLDFA	82.0	83.1	78.1	81.1
LASER	80.4	82.7	75.3	79.5
MultiFiT	**85.3**	**85.6**	**79.9**	**83.6**
Methods below use multi-lingual language models				
MultiBERT	72.0	75.4	66.9	71.4
Methods below use dictionaries or no resources				
VECMAP	75.3	78.2	55.9	69.8
CL-SCL (B=450)	78.1	78.4	73.1	76.5
Teacher (B=20)	38.1	48.6	22.7	36.5
Student-LogReg	78.7	79.6	**78.6**	79.0
Student-MonoBERT	**80.1**	**83.4**	77.6	**80.4**

Table 6.2: Accuracy results on CLS.

Method	**Ar**	**Bg**	**De**	**Es**	**Fa**	**Hr**	**Hu**	**Pl**	**Pt**	**Ru**	**Sk**	**Sl**	**Sv**	**Ug**	**AVG**
Methods below use parallel corpora															
Project-LDC	37.2	-	-	**42.7**	33.1	-	**47.0**	-	-	**48.0**	-	-	-	**38.6**	(41.1)
Project-EP	-	**38.7**	**47.3**	41.8	-	-	38.1	38.8	**39.3**	-	**30.0**	**44.6**	**44.6**	-	(40.4)
CLWE-Par	**37.3**	33.0	43.5	42.6	**40.1**	**30.8**	41.1	**41.7**	38.6	44.8	22.6	32.2	39.1	30.0	37.0
Methods below use comparable corpora or bilingual dictionaries															
CLWE-CP	21.1	28.6	37.7	27.7	20.7	13.9	22.4	30.2	22.2	25.3	24.6	25.3	31.1	25.7	25.5
SentiWordNet (B>20K)	25.6	30.6	32.0	25.3	25.3	19.8	29.2	26.0	22.9	29.5	19.2	28.1	22.7	36.7	26.6
CLWE-Wikt (B>20K)	31.0	45.3	51.0	37.7	31.7	-	40.8	32.9	35.4	**43.8**	36.6	32.1	40.4	28.0	(37.4)
Teacher (B=500)	22.7	42.8	45.5	42.7	30.9	36.4	39.4	40.7	34.4	29.8	40.4	29.5	38.7	20.3	35.3
Student-LogReg	**39.0**	**46.3**	**52.5**	**44.9**	**45.7**	**39.4**	**45.2**	**45.4**	**38.7**	43.2	**43.3**	**42.1**	**50.4**	**41.2**	**44.1**

Table 6.3: Macro-averaged F1 results on TwitterSent, SentiPers, and LORELEI.

Student outperforms Teacher. Teacher considers the noisy translated seed words for classification. Even the simple Student-LogReg technique leverages the context of the seed words and substantially outperforms Teacher. Leveraging pre-trained representations in Student-MonoBERT leads to further improvement. On average, across all languages and datasets, Student outperforms Teacher by 59.6%: CLTS effectively improves performance in L_T *without using labeled documents*.

Student outperforms previous approaches. Student-MonoBERT outperforms *Multi-BERT* by 12.5% on average across all languages and domains in MLDoc and CLS: CLTS effectively generates weak supervision in L_T for fine-tuning monolingual BERT. Importantly, CLTS is effective under minimal resources: with the translation of just $\frac{B}{K}$ seed words per class, Student-LogReg outperforms other approaches that rely on much larger dictionaries (*MultiCCA*, *CL-SCL*, *SentiWordNet*, *CLWE-Wiktionary*). Surprisingly, in several languages CLTS outperforms even more expensive approaches that rely on parallel corpora or machine translation systems (*LASER*, *MultiFiT*, *MT-BOW*, *BiDRL*, *CLDFA*, *CLWE-BW*, *Project-LDC*).

CLTS is effective under a minimal translation budget. Figure 6.4 shows CLTS's performance as a function of the number of seed words per class ($\frac{B}{K}$). Even with just 3 seed words per class, Student-MonoBERT performs remarkably well. Student's and Teacher's performance significantly increases with $\frac{B}{K}$ and most performance gains are obtained for lower values of $\frac{B}{K}$. This is explained by the fact that CLTS prioritizes the most indicative seed words for translation. Therefore, as $\frac{B}{K}$ increases, the additional seed words that are translated are less indicative than the already-translated seed words and as a result have lower chances of translating to important seed words in the target language. The gap between the Teacher and Student performance has a maximum value of 40 absolute accuracy points and decreases as Teacher considers more seed words but does not get lower than 10, highlighting that Student learns predictive patterns in L_T that may never be considered by Teacher.

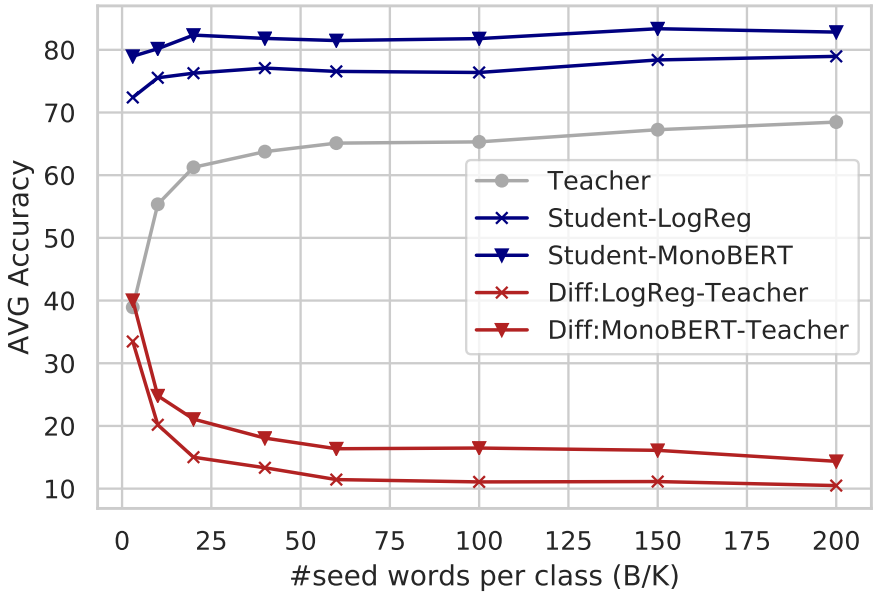


Figure 6.4: Validation accuracy across all MLDoc languages as a function of the translation budget $\frac{B}{K}$.

CLTS is robust to noisy translated seed words. In practice, an indicative seed word in L_S may not translate to an indicative word in L_T. Our results above show that Student in CLTS performs well even when seed words are automatically translated across languages. To further understand our method's behavior with noisy translated seed words, we introduce additional simulated noise of different types and severities. According to Figure 6.5, "unif" and "freq" noise, which replace translated seed words with random words, affect CLTS less than "adv" noise, which introduces many erroneous teacher-labels. Student is less sensitive than Teacher to noisy seed words: their performance gap (*-Diff) increases with the magnitude of translation noise (up to 0.7) for both "unif" and "freq" noise. Student's accuracy is relatively high for noise rates up to 0.3, even with "adv" noise: CLTS is effective even when 30% of the translated seed words are assumed indicative for the wrong class.

Examples of extracted seed words. Table 6.4 reports the 10 most important seed words extracted for each of the four news document classes in CLS. Table 6.5 reports the 10 most important seed words extracted for each binary class and domain in CLS. Figure 6.6

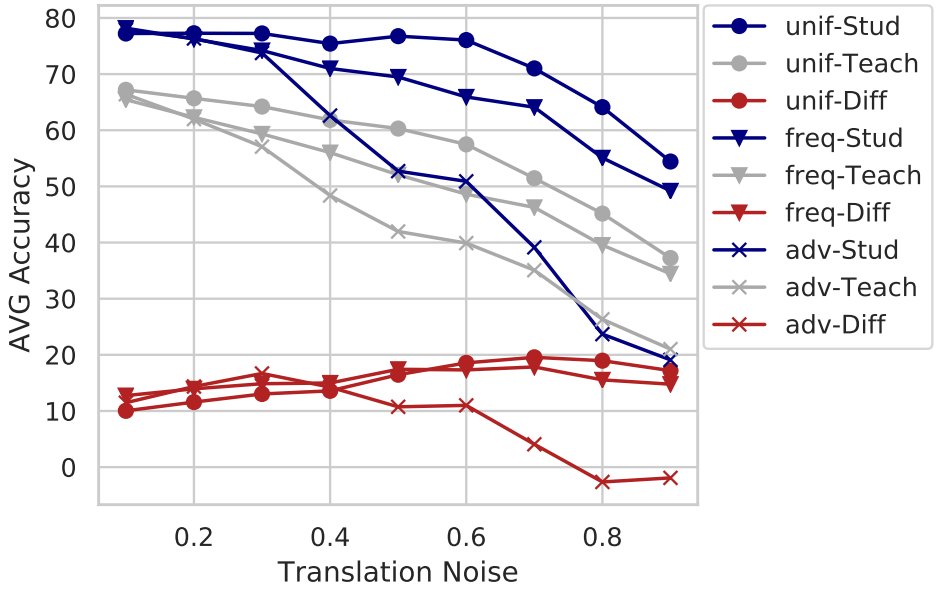


Figure 6.5: Average validation accuracy in MLDoc for Teacher (Teach), Student-LogReg (Stud), and their absolute difference in accuracy (Diff) under different scales of noise applied to the translated seed words: "unif" replaces a seed word with a different word sampled uniformly at random from V_T, "freq" replaces a seed word with a word randomly sampled from V_T with probability proportional to its frequency in D_T, "adv" assigns a seed word to a different random class $k' \neq k$ by swapping its class weights in $\hat{\mathbf{Z}}$.

CCAT	company, inc, ltd, corp, group, profit, executive, newsroom, rating, shares
ECAT	bonds, economic, deficit, inflation, growth, tax, economy, percent, foreign, budget
GCAT	president, police, stories, party, sunday, people, opposition, beat, win, team
MCAT	traders, futures, dealers, market, bids, points, trading, day, copper, prices

Table 6.4: MLDoc: Top 10 English seed words extracted per class (Section 6.3.1).

DVD-POS	best, great, excellent, love, highly, enjoy, wonderful, life, good, favorite
BOOK-POS	excellent, great, lives, wonderful, life, fascinating, fun, easy, love, best
MUSIC-POS	amazing, highly, great, favorites, best, favorite, awesome, classic, excellent, love
DVD-NEG	waste, boring, worst, bad, disappointing, disappointed, awful, poor, horrible, terrible
BOOKS-NEG	money, disappointed, disappointing, boring, disappointment, worst, waste, bad, finish, terrible
MUSIC-NEG	boring, worst, disappointment, poor, sorry, garbage, money, disappointing, bad, horrible

Table 6.5: CLS: Top 10 English seed words extracted per class and domain (Section 6.3.1).

POSITIVE

love, happy, thank, amazing, 😍 , great, cute, beautiful, excited, best, good, !, proud, thanks, nice, awesome, ❤️ , perfect, 😊 , birthday

NEUTRAL

follow, http, 0, new, via, what's, $, followed, co, pm, check,], pleas e, app, …, posted, #gameinsight, vote, https, free

NEGATIVE (sanitized)

hate, f**k, s**t, 😒 , b***h, 😩 , sad, worst, f*****g, stupid, tired, 😭 , 😔 , sucks, wtf, sick, wrong, can't, annoying, people

Figure 6.6: TwitterSent: Top 20 seed words extracted per class (Section 6.3.1). Interestingly, some of the seed words are actually not words but emojis used by Twitter users to indicate the corresponding sentiment class.

reports the 20 most important seed words extracted for each of the 3 sentiment classes in TwitterSent, SentiPers and LORELEI.

Ablation study. Table 6.6 reports results on MLDoc by changing parts of CLTS. The first row reports Student-Logreg without any changes. **Change (a):** using the clarity-scoring (similar to tf-idf weighting) method of [Angelidis and Lapata, 2018b] leads to 3% lower accuracy than extracting seed words from the weights of a classifier trained through sparsity regularization. **Change (b):** obtaining translations through Google Translate leads to 0.8% lower accuracy than using bilingual MUSE dictionary. We observed that Google Translate sometimes translates words to wrong translations without extra context, while MUSE dictionaries provide more accurate translations. **Change (c):** updating Teacher similar to ISWD (Chapter 5), where the Teacher updates seed word qualities but does not consider documents without seed words during training, leads to 1.3% lower accuracy than our approach, which replaces the teacher by the student and thus considers even documents without seed words. **Change (d):** removing seed words from Student's input leads to 2.8% lower accuracy than letting Student consider both seed words and non-seed words. This shows that even without using seed words, Student still performs accurately (77.2%

Change	AVG Acc
- (Original Student-LogReg)	**79.4**
(a) Extract seed words as in [Angelidis and Lapata, 2018b]	77.0 (↓ 3.0%)
(b) Replace MUSE translations by Google Translate	78.8 (↓ 0.8%)
(c) Update Teacher as in ISWD (Chapter 5.3)	78.4 (↓ 1.3%)
(d) Remove seed words from Student's input	77.2 (↓ 2.8%)

Table 6.6: Ablation experiments on MLDoc.

	Target Acc (MultiCCA / MultiBERT / Student-LogReg)			
Source Language	En	De	Es	Fr
En	-	81.2/80.2/**87.4**	72.5/76.9/**86.0**	72.4/72.6/**89.1**
De	56.0/59.7/**82.8**	-	73.2/54.0/**81.3**	71.6/60.0/**84.9**
Es	74.0/74.2/**80.8**	55.8/57.6/**83.3**	-	65.6/71.8/**89.0**
Fr	64.8/76.1/**84.1**	53.7/51.8/**84.5**	65.4/72.1/**85.5**	-

Table 6.7: MultiCCA (left) vs. MultiBERT (center) vs. Student-LogReg (right) for various train (rows) and test (columns) configurations on MLDoc. Student-LogReg substantially outperforms MultiCCA and MultiBERT across all train and test configurations: CLTS effectively transfers weak supervision also from non-English source languages.

accuracy across languages), indicating that Student successfully exploits indicative features in the context of the seed words.

Testing CLTS in non-English source languages. To evaluate whether our results generalize to non-English source languages, we run additional experiments using De, Es, and Fr as source languages in CLS. For those experiments, we also consider En as a target language. Table 6.7 reports the evaluation results. Across all configurations, there is no clear winner between MultiCCA and MultiBERT, but our Student-LogReg consistently outperforms both approaches, indicating that CLTS is also effective with non-English source languages.

6.6 More Cross-Lingual Transfer Applications

In Section 6.5, we evaluated our cross-lingual transfer ideas on several benchmarks for cross-lingual document classification. In this section, we apply cross-lingual transfer for more problems across languages, namely, medical emergency detection (Section 6.6.1) and

MEDICAL EMERGENCY (Uyghur, Sinhalese)

English	->	Uyghur	Sinhalese
1. injured	->	يارىلانغان	තුවාල ලැබුවා
2. attacks	->	ھۇجۇملار	ප්‍රහාර
3. medical	->	medical	වෛද්ය
4. crisis	->	كرىزىس	අර්බුදය
5. disease	->	كېسەل	රෝගය
6. malaria	->	بەزگەك كېسىلى	මැලේරියාව
7. health	->	ساغلاملىق	සෞඛ්‍යය
8. injuring	->	يارىلىنىش	තුවාල වීම
9. yemen	->	يەمەن	යේමනය
10. hospitals	->	دوختۇرخانىلار	රෝහල්
11. others	->	باشقىلار	අන් අය
12. violence	->	زوراۋانلىق	ප්‍රචණ්ඩත්වය
13. tortured	->	قىيىن-قىستاققا ئېلىنغان	වධ හිංසා කළා
14. imprisoned	->	تۇرمىگە تاشلاندى	සිරගත කළා
15. casualties	->	تالاپەتكە ئۇچرىغان	ජීවිත හානි
16. aid	->	ياردەم	ආධාර
17. outbreak	->	تارقىلىش	පැතිරීම
18. terrible	->	قورقۇنچلۇق	භයානකයි
19. hospital	->	دوختۇرخانا	රෝහල
20. victims	->	زىيانكەشلىككە ئۇچرىغۇچىلار	විත්දිනයින්

Figure 6.7: Top 20 extracted seed words for the "medical emergency" class and their translations to Uyghur and Sinhalese obtained through Google Translate. Google Translate erroneously returns "medical" as a Uyghur translation of the word "medical."

foodborne illness detection (Section 6.6.2).

6.6.1 Detecting Medical Emergencies in Low-Resource Languages

We now show a preliminary exploration of CLTS for detecting medical emergency situations in the low-resource Uyghur and Sinhalese languages by just translating a small number of English seed words across languages. For the purpose of this application, we use the LORELEI corpus, as discussed next.

The Low Resource Languages for Emergent Incidents (LORELEI) corpus [Strassel and Tracey, 2016] contains (among others) documents in Uyghur (Ug)[13] and Sinhalese (Si)[14].

[13]LDC2016E57_LORELEI_Uyghur

[14]LDC2018E57_LORELEI_Sinhalese

Each document is labeled with an emergency need. Similar to [Yuan et al., 2020b], we consider binary classification to medical versus non-medical emergency need. In English, we use 806 labeled documents for training the source classifier. In Uyghur, we use 5,000 unlabeled documents for training the student and 226 labeled documents for evaluation. In Sinhalese, we use 5,000 unlabeled documents for training the student and 36 labeled documents for evaluation. (Unfortunately, our number of labeled documents for each language is different than that reported in [Yuan et al., 2020b].) Given the limited number of labeled documents, we do not consider validation sets for our experiments.

To detect emergencies in Uyghur and Sinhalese without labeled data in these languages, we use CLTS and just translate $B = 50$ seed words. As Uyghur and Sinhalese have no entries in the MUSE dictionary (used in Section 4.4), we use Google Translate to get seed word translations.[15] For reproducibility, we cached the translations obtained from Google Translate.

Figure 6.7 shows the top 10 seed words transferred by CLTS for the medical emergency class. We train Student-LogReg because BERT is not available for Uyghur or Sinhalese. End-to-end training and evaluation of CLTS takes just 160 seconds for Uyghur and 174 seconds for Sinhalese. The accuracy in Uyghur is 23.9% for the teacher and 66.8% for the student. The accuracy in Sinhalese is 30.4% for the teacher and 73.2% for the student.

These preliminary results indicate that CLTS could be easily applied for emerging tasks in low-resource languages, for example by asking a bilingual speaker to translate a small number of seed words. We expect such correct translations to lead to further improvements over automatic translations.

[15] Google Translate started supporting Uyghur on February 26, 2020, and Sinhalese at an earlier (unknown) time.

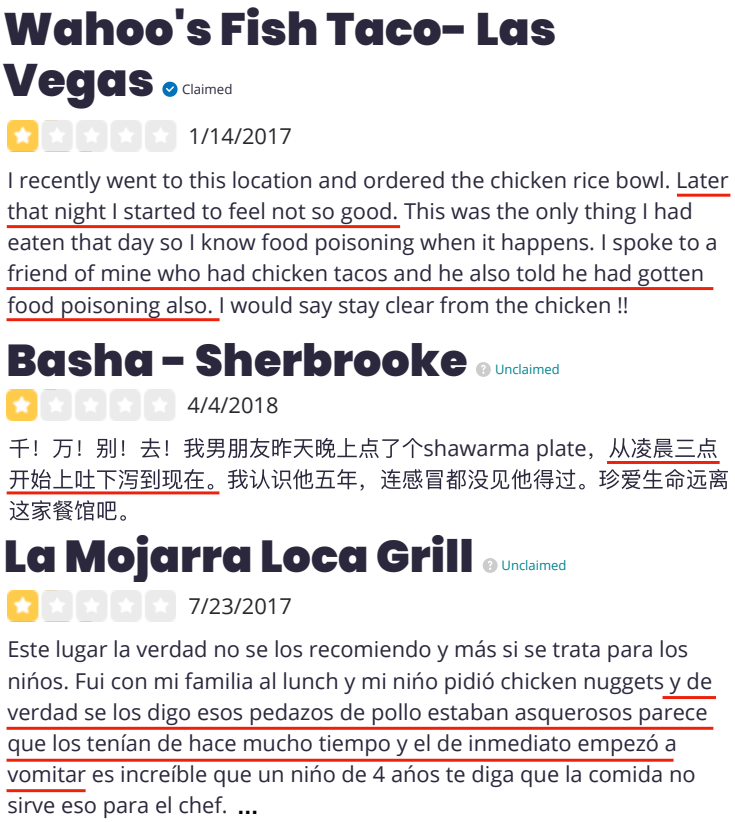


Figure 6.8: Examples of Yelp restaurant reviews discussing food poisoning in different languages.

6.6.2 Foodborne Illness Detection across Languages

We further apply cross-lingual transfer techniques to increase the coverage of our public health system (see Section 3.7). Our current system for foodborne illness detection has been applied for documents in English and, as a result, a promising direction is to increase coverage and recall by considering documents in additional languages, such as Spanish or Chinese. Figure 6.8 shows examples of Yelp restaurant reviews discussing food poisoning in English, Chinese, and Spanish.

To efficiently cover non-English languages without the need for non-English labeled data, in [Liu et al., 2020] we follow a cross-lingual learning approach and transfer English labeled data across languages. First, we collect unlabeled multilingual reviews from Yelp restaurants in New York City, Los Angeles, as well as other metropolitan areas in the Yelp Challenge

dataset.[16] Then, we show that even though recent zero-shot approaches based on pre-trained multi-lingual BERT (mBERT) can effectively align languages for aspects such as sentiment, those approaches are less effective for capturing the nuances of foodborne illness. To improve performance without extra annotations, we create artificial training documents in the target language through machine translation and train mBERT jointly for the source (English) and target language. We demonstrate the benefits of our approach through extensive experiments with Yelp restaurant reviews in seven languages. Our classifiers identify foodborne illness complaints in multilingual reviews from the Yelp Challenge dataset, which highlights the potential of our general approach for deployment in health departments.

6.7 Conclusions

In this chapter, we presented a cross-lingual text classification method, CLTS, that efficiently transfers weak supervision across languages using minimal cross-lingual resources. We summarize the contributions of this chapter as follows: (i) we presented an efficient method for transferring supervision across languages, which first transfers the most important seed words using the translation budget as a sparsity-inducing regularizer when training a classifier in the source language (Section 6.3.1), and then transfers seed words and the classifier's weights across languages, and initializes a teacher classifier in the target language that uses the translated seed words (Section 6.3.2); (ii) we effectively applied our weakly-supervised co-training approach from Chapter 5 to this cross-lingual setting for training accurate classifiers in the target language without any labeled target documents (Section 6.3.3); (iii) we evaluated our ideas by performing an extensive experimental evaluation on document classification benchmarks across 18 diverse languages (Sections 6.4 and 6.5); (iv) we applied CLTS for the detection of medical emergency situations in the low-resource Uyghur and Sinhalese languages by just translating a small number of English seed words across languages (Section 6.6.1); and (v) we presented a cross-lingual transfer method for extending our foodborne

[16] `https://www.kaggle.com/yelp-dataset/yelp-dataset`

illness detection across languages without extra labeling efforts (Section 6.6.2).

Our findings show that CLTS effectively transfers supervision from English to all 18 languages for training classifiers using unlabeled-only target documents. Even a simple student outperforms the teacher across all languages by 59.6%, thus proving the effectiveness of our co-training approach for tasks beyond aspect detection, which was our main focus in Chapter 5. CLTS outperforms previous state-of-the-art approaches that require more complex models and more expensive resources, highlighting the promise of generating weak supervision in the target language. We further showed that CLTS is robust to noisy translated seed words and therefore can be used even when there is no budget to hire a bilingual speaker by instead using automatically translated seed words, e.g., via machine translation. Due to the resource-efficiency of our approach, we were able to apply it to low-resource languages and trained accurate classifiers for emergency event detection. Also, by applying our cross-lingual transfer ideas for foodborne illness detection, we trained classifiers that successfully identified reviews discussing food poisoning across several languages, which highlights the potential of our approach for successful, real-world deployment in health departments. In the future, it would be interesting to extend CLTS for more tasks, such as cross-lingual named-entity recognition [Xie et al., 2018]. A first step towards this goal is to expand the teacher architecture to support more complex types of supervision beyond seed words. In the next chapter, we present (among other contributions) a new teacher architecture that supports more general labeling rules.

Chapter 7: Self-Training with Labeling Rules

In Chapters 5 and 6, we presented two architecture-agnostic frameworks for training text classifiers using seed words and their translations, respectively. Not all classification tasks, however, can be effectively addressed using human supervision in the form of seed words. To capture a broader variety of tasks, this chapter presents an architecture-agnostic method that leverages more general labeling rules, few labeled data, and unlabeled data. First, we motivate the problem of learning with labeling rules (Section 7.1). Second, we discuss related work and define our problem of focus (Section 7.2). Third, we present our ASTRA framework, which can train any classifier using labeling rules, few labeled data, and unlabeled data (Section 7.3). Then, we present our experimental evaluation for classification across six weak supervision benchmarks (Sections 7.4 and 7.5). Finally, we summarize the contributions of this chapter (Section 7.6).

7.1 Overview and Motivation

In order to mitigate labeled data scarcity, recent works have tapped into weak or noisy sources of supervision, such as regular expression patterns [Augenstein et al., 2016], class-indicative keywords [Ren et al., 2018b; Karamanolakis et al., 2019a], alignment rules over existing knowledge bases [Mintz et al., 2009; Xu et al., 2013], or heuristic labeling rules [Ratner et al., 2017; Bach et al., 2019; Badene et al., 2019; Awasthi et al., 2020]. These different types of sources can be used as weak rules for heuristically annotating large amounts of unlabeled data. For instance, consider the question type classification task from the TREC dataset with regular expression patterns such as: *label all questions containing the token "when" as* `numeric` (e.g., "When was Shakespeare born?"). Approaches relying on such

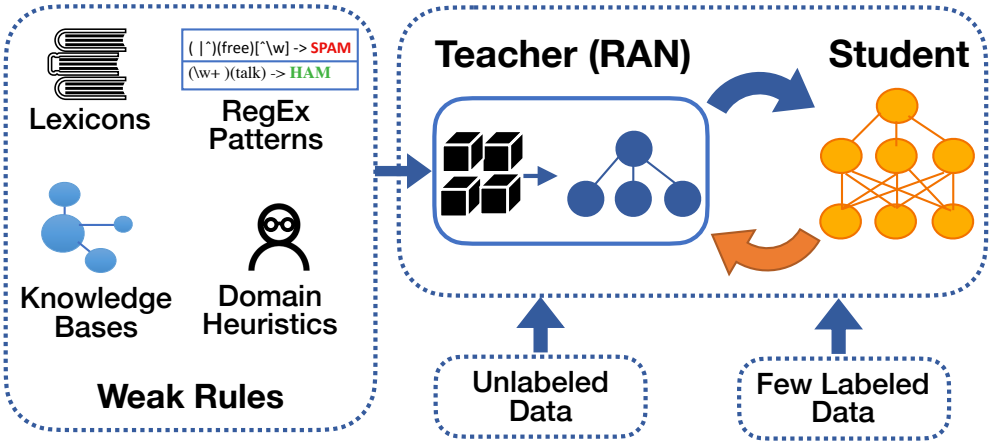


Figure 7.1: Our weak supervision framework, ASTRA, leverages domain-specific rules, a large amount of (task-specific) unlabeled data, and a small amount of labeled data via iterative self-training.

Label	Pattern
HUM	`( |^)(which|who|what)[^\w]*([^\s]+ )*(person| man|woman|human)[^\w]*( |$)`
ENTY	`( |^)(what)[^\w]*(\w+ ){0,1}(is)[^\w]* *([^\s]+ )*(surname|address|name)[^\w]*( |$)`
NUM	`( |^)(what)[^\w]* ([^\s]+ )*(percentage |share|number|population)[^\w]*( |$)`
DESC	`( |^)(how|what|what)[^\w]* * (\w+ ){0,1}(do|does|does)[^\w]*( |$)`

Table 7.1: Sample of REGEX rules from the TREC-6 dataset capturing the various question categories (HUM: Human, ENTY: Entity, NUM: Numeric Value, DESC: Description, ABBR: Abbreviation).

weak rules typically suffer from the following challenges: (i) *Noise.* Rules by their heuristic nature rely on shallow patterns and may predict wrong labels for many instances. For example, the question "When would such a rule be justified?" refers to circumstances rather than numeric expressions. (ii) *Coverage.* Rules generally have a low coverage as they assign labels to only specific subsets of instances. (iii) *Conflicts.* Different rules may generate conflicting predictions for the same instance, making it challenging to train a robust classifier.

To address the challenges with conflicting and noisy rules, existing approaches learn weights indicating how much to trust individual rules. In the absence of large-scale manual annotations, the rule weights are usually learned via mutual agreement and disagreement of rules over unlabeled data [Ratner et al., 2017; Platanios et al., 2017; Sachan et al., 2018; Bach et al., 2019; Ratner et al., 2019; Awasthi et al., 2020]. An important drawback of these approaches is low coverage, since rules assign weak labels to only a subset of the data, thus leading to low rule overlap to compute rule agreement. For instance, in our experiments on six real-world datasets, we observe that 66% of the instances are covered by fewer than 2 rules and 40% of the instances are not covered by any rule at all. Rule sparsity limits the effectiveness of previous approaches, thus leading to strong assumptions, such as that each rule has the same weight across all instances [Ratner et al., 2017; Bach et al., 2019; Ratner et al., 2019], or that additional supervision is available in the form of labeled "exemplars" used to create such rules in the first place [Awasthi et al., 2020]. Most importantly, all these works ignore (as a data pre-processing step) unlabeled instances that are not covered by any of the rules, thus leaving potentially valuable data behind.

Overview of our method. In this work, we present a weak supervision framework, which we call ASTRA, which considers all task-specific unlabeled instances and domain-specific rules without strong assumptions about the nature or source of the rules. ASTRA makes effective use of a small amount of labeled data, lots of task-specific unlabeled data, and domain-specific rules through iterative teacher-student co-training (see Figure 7.1). A stu-

dent model based on contextualized representations provides pseudo-labels for all instances, thereby allowing us to leverage all unlabeled data including instances that are not covered by any heuristic rules. To deal with the noisy nature of heuristic rules and pseudo-labels from the student, we develop a rule attention (teacher) network that learns to predict the fidelity of these rules and pseudo-labels conditioned on the context of the instances that they cover. We develop a semi-supervised learning objective based on minimum entropy regularization to learn all of the above tasks jointly without the requirement of additional rule-exemplar supervision.

Overall, we make the following contributions:

- We propose an iterative self-training mechanism for training deep neural networks with weak supervision by making effective use of task-specific unlabeled data and domain-specific heuristic rules. The self-trained student model predictions augment the weak supervision framework with instances that are not covered by rules.
- We propose a rule attention teacher network (RAN) for combining multiple rules and student model predictions with instance-specific weights conditioned on the corresponding contexts. Furthermore, we construct a semi-supervised learning objective for training RAN without strong assumptions about the structure or nature of the weak rules.
- We demonstrate the effectiveness of our approach on several benchmark datasets for text classification, where our method significantly outperforms state-of-the-art weak supervision methods.

We start with a review of the related work and define our problem of focus (Section 7.2). We continue as follows:

- We develop ASTRA, a weakly-supervised learning framework for training any type of classifier using labeling rules, few labeled data, and unlabeled data (Section 7.3).[1]

[1]Our Python implementation is publicly available at `https://github.com/microsoft/ASTRA`.

- We evaluate our ideas by conducting an experimental evaluation on sequence classification and sequence tagging datasets (Sections 7.4 and 7.5).

Finally, we discuss the implications of our work (Section 7.6). The material described in this chapter appears in [Karamanolakis et al., 2021].

7.2 Related Work and Problem Definition

In this section, we discuss related work on self-training and learning with noisy labels or rules, and define our problem of focus. Refer to [Hedderich et al., 2021b] for a thorough survey of approaches addressing low-resource scenarios.

Self-training. Self-training [Yarowsky, 1995; Nigam and Ghani, 2000; Lee, 2013], one of the earliest semi-supervised learning approaches [Chapelle et al., 2009], trains a base model (student) on a small amount of labeled data; applies it to pseudo-label (task-specific) unlabeled data; uses pseudo-labels to augment the labeled data; and re-trains the student in an iterative manner. Self-training has recently been shown to obtain state-of-the-art performance for tasks like image classification [Li et al., 2019; Xie et al., 2020; Zoph et al., 2020], few-shot text classification [Mukherjee and Awadallah, 2020; Wang et al., 2021], and neural machine translation [Zhang and Zong, 2016; He et al., 2019], and has shown complementary advantages to unsupervised pre-training [Zoph et al., 2020]. A typical issue in self-training is error propagation from noisy pseudo-labels. This is addressed in ASTRA via a rule attention network that computes the fidelity of pseudo-labels instead of directly using them to re-train the student.

Learning with noisy labels. Classification under label noise from a single source has been an active research topic [Frénay and Verleysen, 2013]. A major line of research focuses on correcting noisy labels by learning label corruption matrices [Patrini et al., 2017; Hendrycks et al., 2018; Zheng et al., 2021]. More related to our work are the instance re-weighting

approaches [Ren et al., 2018b; Shu et al., 2019], which learn to up-weight and down-weight instances with cleaner and noisy labels respectively. However, instance re-weighting methods operate only at instance-level and do not consider rule-specific importance. Our approach learns both instance- and rule-specific fidelity weights and substantially outperforms [Ren et al., 2018b] across all datasets.

Learning with multiple labeling rules. To address the challenges with multiple noisy rules, existing approaches learn rule weights based on mutual rule agreements with some strong assumptions. For instance, [Meng et al., 2018; Karamanolakis et al., 2019a; Mekala and Shang, 2020] denoise seed words using vector representations of their semantics. However it is difficult to generalize these approaches from seed words to more general labeling rules that only predict heuristic labels (as in our datasets). [Ratner et al., 2017; Sachan et al., 2018; Ratner et al., 2019] assume each rule to be equally accurate across all the instances that it covers. [Awasthi et al., 2020] learn rule-specific and instance-specific weights but assume access to *labeled exemplars* that were used to create the rules in the first place. Most importantly, all these works ignore unlabeled instances that are not covered by any of the rules, while our approach leverages all unlabeled instances via self-training.

Problem definition. Let $\mathcal{X}$ denote the instance space and $\mathcal{Y} = \{1, \ldots, K\}$ denote the label space for a K-class classification task. We consider a small set of manually-labeled examples $D_L = \{(s_l, y_l)\}$, where $s_l \in \mathcal{X}$ and $y_l \in \mathcal{Y}$ and a large set of unlabeled examples $D_U = \{s_i\}$. We also consider a set of pre-defined heuristic rules $R = \{r^j\}$, where each rule r^j has the general form of a labeling function that considers as input an instance $s_i \in \mathcal{X}$ (and potentially additional side information), and either assigns a *weak* label $\mathbf{q}_i^j \in \{0,1\}^K$ (one-hot encoding) or does not assign a label for s_i, in which case we say that r^j does not cover s_i. Our goal is to leverage D_L, D_U, and R to train a classifier that, given an unseen test instance $s' \in \mathcal{X}$, predicts a label $y' \in \mathcal{Y}$.

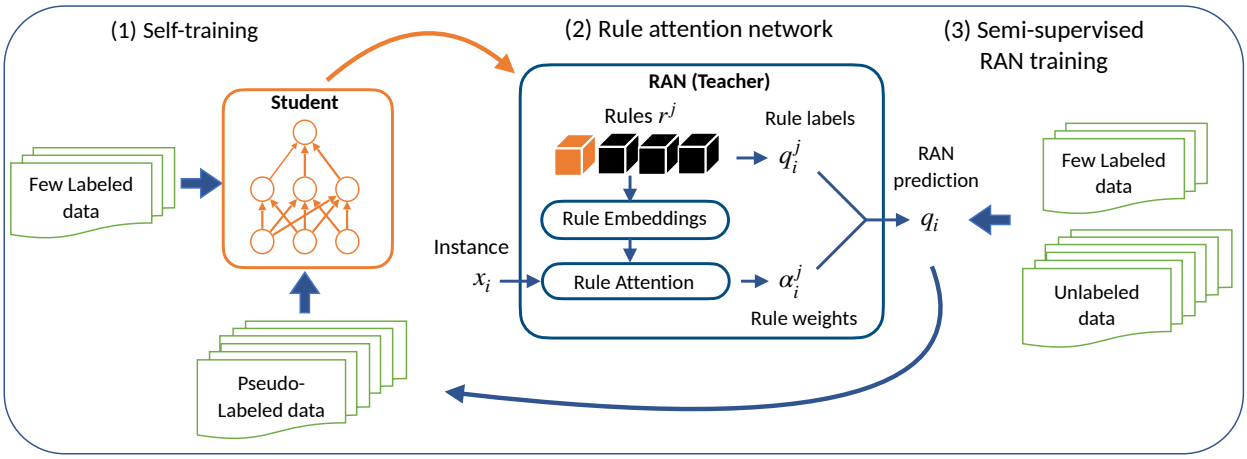


Figure 7.2: Our ASTRA framework for self-training with weak supervision.

7.3 Self-Training with Weak Supervision (ASTRA)

We now present our ASTRA framework for addressing the problem defined in Section 7.2 by effectively leveraging D_L, D_U, and R. In contrast to previous weak supervision methods, ASTRA considers all unlabeled examples in D_U, including examples that are not covered by any rules in R. Our architecture has two main components, namely the base student model (Section 7.3.1) and the rule attention teacher network (Section 7.3.2), which are iteratively co-trained in a self-training framework.

7.3.1 Base Student Model

Our self-training framework starts with a base model trained on the available small labeled set D_L. The model is then applied to unlabeled data D_U to obtain pseudo-labeled instances. In classic self-training [Riloff, 1996; Nigam and Ghani, 2000], the student model's pseudo-labeled instances are directly used to augment the training dataset and iteratively re-train the student. In our setting, we augment the self-training process with weak labels drawn from our teacher model that also considers rules in R (described in the next section).

The overall self-training process can be formulated as:

$$\min_{\theta} \; \mathbb{E}_{s_l, y_l \in D_L}[-\log \; p_\theta(y_l \mid s_l)] + \lambda \mathbb{E}_{s \in D_U} \mathbb{E}_{y \sim q_{\phi^*}(y|s)}[-\log \; p_\theta(y \mid s)] \tag{7.1}$$

where, $p_\theta(y|s)$ is the conditional distribution under student's parameters θ; $\lambda \in \mathbb{R}$ is a hyper-parameter controlling the relative importance of the two terms; and $q_{\phi^*}(y \mid s)$ is the conditional distribution under the teacher's parameters ϕ^* from the last iteration that is fixed in the current iteration.

7.3.2 Rule Attention Teacher Network (RAN)

Our Rule Attention Teacher Network (RAN) aggregates multiple weak sources of supervision with trainable weights and computes a soft weak label $\mathbf{q}_i$ for an unlabeled instance s_i. One of the potential drawbacks of relying only on heuristic rules is that a lot of data get left behind. Heuristic rules (e.g., regular expression patterns, keywords) usually cover just a subset of the data. Therefore, a substantial number of instances are not covered by any rules and thus are not considered in prior weakly supervised learning approaches [Ratner et al., 2017; Awasthi et al., 2020]. To address this challenge and leverage contextual information from all available task-specific unlabeled data, we leverage the corresponding pseudo-labels predicted by the base student model (from Section 7.3.1). To this end, we apply the student to the unlabeled data $s \in D_U$ and obtain pseudo-label predictions as $p_\theta(y|s)$. These predictions are used to augment the set of already available weak rule labels to increase rule coverage.

Let $R_i \subset R$ be the set of all heuristic rules that *cover* instance s_i. The objective of RAN is to aggregate the weak labels predicted by all rules $r^j \in R_i$ and the student pseudo-label $p_\theta(y|s_i)$ to compute a soft label $\mathbf{q}_i$ for every instance s_i from the unlabeled set D_U. In other words, RAN considers the student as an additional source of weak rule. Aggregating all rule labels into a single label $\mathbf{q}_i$ via simple majority voting (i.e., predicting the label assigned by the majority of rules) may not be effective as it treats all rules equally, while in practice

certain rules are more accurate than others.

RAN predicts pseudo-labels $\mathbf{q}_i$ by aggregating rules with trainable weights $a_i^{(\cdot)} \in [0,1]$ that capture their fidelity towards an instance s_i as:

$$\mathbf{q}_i = \frac{1}{Z_i}\Bigg(\sum_{j:\ r^j \in R_i} a_i^j \mathbf{q}_i^j + a_i^S p_\theta(\cdot|s_i) + a_i^u \mathbf{u}\Bigg), \tag{7.2}$$

where a_i^j and a_i^S are the fidelity weights for the heuristic rule labels q_i^j and the student assigned pseudo-label $p_\theta(y|s_i)$ for an instance s_i, respectively; $\mathbf{u}$ is a uniform rule distribution that assigns equal probabilities for all the K classes as $\mathbf{u} = (\frac{1}{K}, \ldots, \frac{1}{K})$; a_i^u is the weight assigned to the "uniform rule" for s_i, which is computed as a function of the rest of the rule weights: $a_i^u = (|R_i| + 1 - \sum_{j:\ r^j \in R_i} a_i^j - a_i^S)$; and Z_i is a normalization coefficient to ensure that $\mathbf{q}_i$ is a valid probability distribution. $\mathbf{u}$ acts as a uniform smoothing factor that prevents overfitting for sparse settings, for instance, when a single weak rule covers an instance.

According to Eq. (7.2), a rule r^j with higher fidelity weight a_i^j contributes more to the computation of $\mathbf{q}_i$. If $a_i^j = 1\ \forall r^j \in \{R_i \cup p_\theta\}$, then RAN reduces to majority voting. If $a_i^j = 0\ \forall r^j \in \{R_i \cup p_\theta\}$, then RAN ignores all rules and predicts $\mathbf{q}_i = \mathbf{u}$. Note the distinction of our setting to recent works like Snorkel [Ratner et al., 2017], which learns global rule-weights $a_i^j = a^j\ \forall s_i$ by ignoring the instance-specific rule fidelity. Our proposed approach is flexible as it can assign different rule weights a_i^j to different instances, but it is challenging to learn how to assign these weights as we do not assume prior knowledge of the internal structure of the labeling rules in R.

In order to effectively compute rule fidelities, RAN considers instance embeddings that capture the context of instances beyond the shallow patterns considered by rules. In particular, we model the weight a_i^j of rule r^j as a function of the context of the instance s_i and r^j through an attention-based mechanism. Consider $\mathbf{h}_i \in \mathbb{R}^{d'}$ to be the hidden state representation of s_i from the base student model. Also, consider the (trainable) embedding of each rule r^j as $\mathbf{e}_j = g(r^j) \in \mathbb{R}^d$. We use $\mathbf{e}_j$ as a query vector with *sigmoid attention* to

compute instance-specific rule attention weights as:

$$a_i^j = \sigma(f(\mathbf{h}_i)^T \cdot \mathbf{e}_j) \in [0,1], \tag{7.3}$$

where f is a multi-layer perceptron that projects $\mathbf{h}_i$ to $\mathbb{R}^d$ and $\sigma(\cdot)$ is the sigmoid function. Rule embedding allows us to exploit the similarity between different rules in terms of instances they cover, and further leverage their semantics for modeling agreement. RAN computes the student's weight a_i^S using the same procedure as for computing the rule weights a_i^j.

Note that the rule predictions $\mathbf{q}_i^j$ are considered fixed, while we estimate their attention weights. The above coupling between rules and instances via their corresponding embeddings $\mathbf{e}_j$ and $\mathbf{h}_i$ allows us to obtain representations where similar rules cover similar contexts, and model their agreements via the attention weights a_i^j. To this end, the trainable parameters of RAN (f and g) are shared across all rules and instances. Next, we describe how to train RAN.

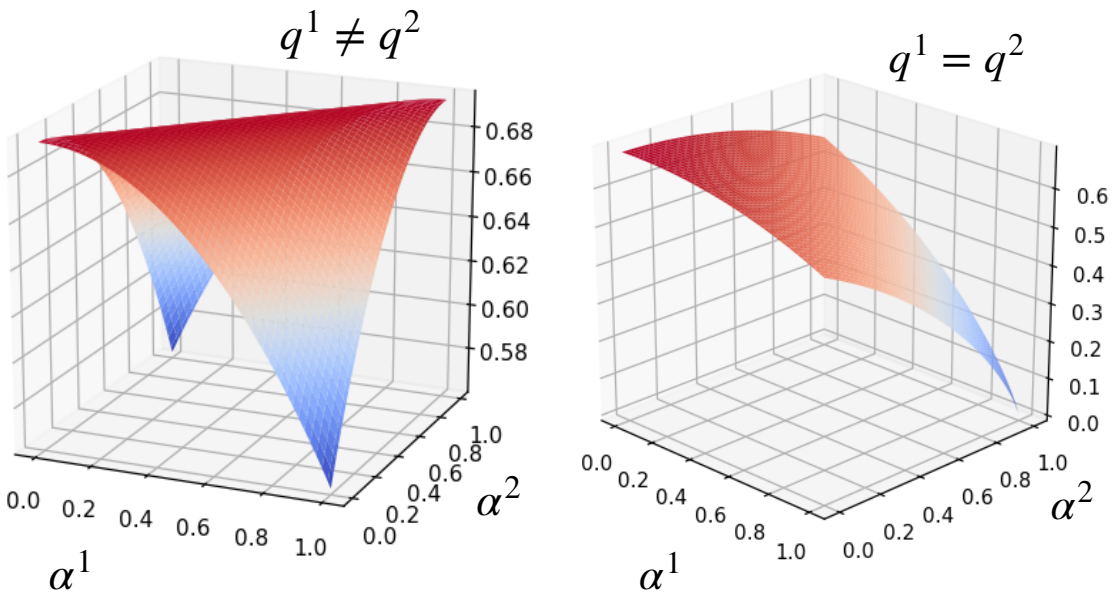
7.3.3 Semi-Supervised Learning of ASTRA

Learning to predict instance-specific weights $a_i^{(\cdot)}$ for the weak sources (including rules and student pseudo-labels) is challenging due to the absence of any explicit knowledge about the source quality and limited amount of labeled training data. We thus treat the weights $a_i^{(\cdot)}$ as latent variables and propose a semi-supervised objective for training RAN with supervision on the coarser level of $\mathbf{q}_i$:

$$\mathcal{L}^{RAN} = -\sum_{(s_i,y_i)\in D_L} y_i \log \mathbf{q}_\phi(y_i|s_i) - \gamma \sum_{s_i \in D_U} \sum_{y_i \in \mathcal{Y}} \mathbf{q}_\phi(y_i|s_i) \log \mathbf{q}_\phi(y_i|s_i). \tag{7.4}$$

The hyperparameter γ controls the relative importance of the two loss terms in Eq. (7.4) that we describe below.

Given task-specific labeled data D_L, the first term in Eq. (7.4) is the cross-entropy loss



(a) Rule predictions disagree.

(b) Rule predictions agree.

Figure 7.3: Variation in unsupervised entropy loss with instance-specific rule predictions and attention weights encouraging rule agreement. Consider this illustration with two rules for a given instance. When rule predictions disagree ($\mathbf{q}^1 \neq \mathbf{q}^2$), minimum loss is achieved for attention weights a^1=0, a^2=1 or a^1=1, a^2=0. When rule predictions agree ($\mathbf{q}^1$=$\mathbf{q}^2$), minimum loss is achieved for attention weights a^1=a^2=1. For instances covered by three rules, if $\mathbf{q}^1$=$\mathbf{q}^2$≠$\mathbf{q}^3$, the minimum loss is achieved for a^1=a^2=1 and a^3=0.

Algorithm 2 Self-training with Weak Supervision

Input: Small amount of labeled data D_L; task-specific unlabeled data D_U; weak rules R

Outputs: Student $p^*_\theta(\cdot)$, RAN Teacher $q^*_\phi(\cdot)$

1: Train student $p_\theta(\cdot)$ using D_L

2: **Repeat until convergence:**

2.1: Train teacher $q_\phi(\cdot)$ using D_L, D_U through Eq. (7.2) and (7.4)

2.2: Apply $q_\phi(y \mid s, R, p_\theta)$ to $s \in D_U$ to obtain pseudo-labeled data: $D_{RAN} = \{(s_i, \mathbf{q}_i)\}_{s_i \in D_U}$ through Eq. (7.2)

2.3: Train $p_\theta(\cdot)$ using D_L, D_{RAN} through Eq. (7.1)

between the teacher's label $\mathbf{q}_i$ and the corresponding clean label y_i for the instance s_i. This term penalizes weak sources that assign labels $\mathbf{q}_i^{(\cdot)}$ that contradict with the ground-truth label y_i by assigning a low instance-specific fidelity weight $a_i^{(\cdot)}$.

The second term in Eq. (7.4) is the entropy of the aggregated pseudo-label $\mathbf{q}_i$ on unlabeled data D_U. Minimum entropy regularization is effective in settings with small amounts of labeled data by leveraging unlabeled data [Grandvalet and Bengio, 2005], and is highly beneficial in our setting because it encourages RAN to predict weights that maximize rule agreement. Since the teacher label $\mathbf{q}_i$ is obtained by aggregating weak labels $\mathbf{q}_i^{(\cdot)}$, entropy minimization encourages RAN to predict higher instance-specific weights $a_i^{(\cdot)}$ to sources that agree in their labels for s_i, and lower weights when there are disagreements between weak sources – aggregated across all the unlabeled instances.

Figure 7.3 plots the minimum entropy loss over unlabeled data over two scenarios where two rules disagree and agree with each other for a given instance. The optimal instance-specific fidelity weights $a_i^{(\cdot)}$ are 1 when rules agree with each other, thereby, assigning credits to both rules, and only one of them when they disagree. We use this unsupervised entropy loss in conjunction with cross-entropy loss over labeled data to ensure grounding.

End-to-end learning. Algorithm 2 presents an overview of our learning mechanism. We first use the small amount of labeled data to train a base student model that generates

pseudo-labels and augments heuristic rules over unlabeled data. Our RAN network computes fidelity weights to combine these different weak labels via minimum entropy regularization to obtain an aggregated pseudo-label for every unlabeled instance. This is used to re-train the student model with the above student-teacher training repeated till convergence.

7.4 Experimental Settings

Datasets. We evaluate our framework on the following six benchmark datasets for weak supervision from [Ratner et al., 2017] and [Awasthi et al., 2020]:

- **TREC:** Question classification from TREC-6 into 6 categories: Abbreviation, Entity, Description, Human, Location, Numeric-value. Table 7.1 reports a sample of regular expression rules out of the 68 rules used in the TREC dataset. TREC has 13 keyword-based (coverage=62%) and 55 regular expression-based (coverage=57%) rules.

- **SMS:** Binary Spam vs. Not Spam classification of SMS messages. SMS has 16 keyword-based (coverage=4%) and 57 regular expression-based (coverage=38%) rules.

- **YouTube:** Binary Spam vs. Not Spam classification of YouTube comments.[2] YouTube has 5 keyword-based (coverage=48%), 1 regular expression-based (coverage=23%), 1 length-based (coverage=23%), and 3 classifier-based (coverage=46%) rules.

- **CENSUS:** Binary income classification on the UCI CENSUS dataset on whether a person earns more than $50K or not. This is a non-textual dataset and is considered to evaluate the performance of our approach under the low sparsity setting, since the 83 rules are automatically extracted and have a coverage of 100%.

- **MIT-R:** Slot-filling in sentences on restaurant search queries in the MIT-R dataset: each token is classified into 9 classes (Location, Hours, Amenity, Price, Cuisine, Dish,

[2] `https://archive.ics.uci.edu/ml/machine-learning-databases/00380/YouTube-Spam-Collection-v1.zip`

	TREC	SMS	YouTube	CENSUS	MIT-R	Spouse
Labeled Training Data ($\|D_L\|$)	68	69	100	83	1842	100
Unlabeled Training Data ($\|D_U\|$)	5K	5K	2K	10K	65K	22K
Test Data	500	500	250	16K	14K	3K
#Classes	6	2	2	2	9	2
#Rules	68	73	10	83	15	9
Rule Accuracy (Majority Voting)	60.9%	48.4%	82.2%	80.1%	40.9%	44.2%
Rule Coverage (instances covered by ≥ 1 rule)	95%	40%	87%	100%	14%	25%
Rule Overlap (instances covered by ≥ 2 rules)	46%	9%	48%	94%	1%	8%

Table 7.2: Dataset statistics.

Restaurant Name, Rating, Other). MIT-R has 5 keyword-based (coverage=6%) and 10 regular expression-based (coverage=10%) rules.

- **Spouse:** Relation classification in the Spouses dataset[3], whether pairs of people mentioned in a sentence are/were married or not. Spouse has 6 keyword-based (coverage=23%), 1 heuristic-based (coverage=4%), and 2 distant supervision-based (coverage=0.2%) rules.

Table 7.2 shows the dataset statistics along with the amount of labeled, unlabeled data and domain-specific rules for each dataset. For a fair comparison, we use exactly the same set of rules as in the previous work for the benchmark datasets. These rules include regular expression patterns, lexicons, and knowledge bases for weak supervision. Most of these rules were constructed manually, except for the CENSUS dataset, where rules have been automatically extracted with a coverage of 100%.

On average across all the datasets, 66% of the instances are covered by fewer than 2 rules, whereas 40% are not covered by any rule at all – demonstrating the sparsity in our setting. We also report the accuracy of the rules in terms of majority voting on the task-specific unlabeled datasets.

Evaluation. We train ASTRA five times for five different random splits of the labeled training data and evaluate on held-out test data. We report the average performance as well

[3] `https://www.dropbox.com/s/jmrvyaqew4zp9cy/spouse_data.zip`

Method	Learning to Weight Rules	Learning to Weight Instances	Unlabeled (no rules)
Majority	-	-	-
Snorkel [Ratner et al., 2017]	✓	-	-
PosteriorReg [Hu et al., 2016]	✓	-	-
L2R [Ren et al., 2018a]	-	✓	-
ImplyLoss [Awasthi et al., 2020]	✓	✓	-
Self-train	-	-	✓
ASTRA	✓	✓	✓

Table 7.3: ASTRA learns rule-specific and instance-specific attention weights and leverages task-specific unlabeled data covered by no rules.

as the standard deviation across multiple runs. We report the same evaluation metrics as used in prior works [Ratner et al., 2017; Awasthi et al., 2020] for a fair comparison.

Model configuration. Our student model consists of embeddings from pre-trained language models like ELMO [Peters et al., 2018] or BERT [Devlin et al., 2019] for generating contextualized representations for an instance, followed by a softmax classification layer. The RAN teacher model considers a rule embedding layer and a multilayer perceptron for mapping the contextualized representation for an instance to the rule embedding space.

Configuration for iterative Teacher-Student training. At each iteration, we train the RAN teacher on unlabeled data and fine-tune on clean labeled data. We found this to be simpler than and at least as effective as jointly training on unlabeled and clean labeled data, where in the latter we had to fine-tune the hyperparameter γ (see Eq. (7.4))). Also at each iteration, we train the student on pseudo-labeled teacher data and fine-tune on clean labeled data. We consider a maximum number of 25 self-training iterations (with early stopping of patience 3 epochs) and keep the models' performances for the iteration corresponding to the highest validation performance.

Baselines. We compare our method with the following methods:

- **Majority** predicts the majority vote of the rules with ties resolved by predicting a random class.
- **LabeledOnly** trains classifiers using only labeled data (fully supervised baseline).
- **Self-train** [Nigam and Ghani, 2000; Lee, 2013] leverages both labeled and unlabeled data for iterative self-training on pseudo-labeled predictions over task-specific unlabeled data. This baseline ignores domain-specific rules.
- **Snorkel+Labeled** [Ratner et al., 2017] trains classifiers using weakly-labeled data with a generative model. The model is trained on unlabeled data for computing rule weights in an unsupervised fashion, and learns a single weight per rule across all instances. It is further fine-tuned on labeled data.
- **L2R** [Ren et al., 2018b] learns to re-weight noisy or weak labels from domain-specific rules via meta-learning. It learns instance-specific but not rule-specific weights.
- **PosteriorReg** [Hu et al., 2016] trains classifiers using rules as soft constraints via posterior regularization [Ganchev et al., 2010].
- **ImplyLoss** [Awasthi et al., 2020] leverages *exemplar*-based supervision as additional knowledge for learning instance-specific and rule-specific weights by minimizing an implication loss over unlabeled data. This requires maintaining a record of all instances used to create the weak rules in the first place.

Table 7.3 shows a summary of the different methods contrasting them on how they learn the weights (rule-specific or instance-specific) and if they leverage task-specific unlabeled data that are not covered by any rules.

7.5 Experimental Results

Overall results. Table 7.4 summarizes the main results across all datasets. Among all the semi-supervised methods that leverage weak supervision from domain-specific rules, ASTRA

	TREC (Acc)	SMS (F1)	YouTube (Acc)	CENSUS (Acc)	MIT-R (F1)	Spouse (F1)
Majority	60.9 (0.7)	48.4 (1.2)	82.2 (0.9)	80.1 (0.1)	40.9 (0.1)	44.2 (0.6)
LabeledOnly	66.5 (3.7)	93.3 (2.9)	91.0 (0.7)	75.8 (1.7)	74.7 (1.1)	47.9 (0.9)
Snorkel+Labeled	65.3 (4.1)	94.7 (1.2)	93.5 (0.2)	79.1 (1.3)	75.6 (1.3)	49.2 (0.6)
PosteriorReg	67.3 (2.9)	94.1 (2.1)	86.4 (3.4)	79.4 (1.5)	74.7 (1.2)	49.4 (1.1)
L2R	71.7 (1.3)	93.4 (1.1)	92.6 (0.5)	82.4 (0.1)	58.6 (0.4)	49.5 (0.7)
ImplyLoss	75.5 (4.5)	92.2 (2.1)	93.6 (0.5)	80.5 (0.9)	75.7 (1.5)	49.8 (1.7)
Self-train	71.1 (3.9)	95.1 (0.8)	92.5 (3.0)	78.6 (1.0)	72.3 (0.6)	51.4 (0.4)
ASTRA (ours)	**80.3** (2.4)	**95.3** (0.5)	**95.3** (0.8)	**83.1** (0.4)	**76.9** (0.6)	**62.3** (1.1)

Table 7.4: Overall result comparison across multiple datasets. Results are aggregated over five runs with random training splits and standard deviation across the runs in parentheses.

outperforms Snorkel by 6.1% in average accuracy across all datasets by learning instance-specific rule weights in conjunction with self-training over unlabeled instances that are not covered by any rules. Similarly, ASTRA also improves over a recent work and the best performing baseline ImplyLoss by 3.1% on average. Notably, our method does not require additional supervision at the level of exemplars used to create rules in contrast to ImplyLoss.

Self-training over unlabeled data. Recent works for tasks like image classification [Li et al., 2019; Xie et al., 2020; Zoph et al., 2020], neural sequence generation [Zhang and Zong, 2016; He et al., 2019] and few-shot text classification [Mukherjee and Awadallah, 2020; Wang et al., 2021] show the effectiveness of self-training methods in exploiting task-specific unlabeled data with stochastic regularization techniques like dropouts and data augmentation. We also make similar observations for our weakly supervised tasks, where classic self-train methods ("Self-train") leveraging only a few task-specific labeled examples and lots of unlabeled data outperform weakly supervised methods like Snorkel and PosteriorReg that have additional access to domain-specific rules.

Self-training with weak supervision. Our framework ASTRA provides an efficient method to incorporate weak supervision from domain-specific rules to augment the self-training framework and improves by 6% over classic self-training.

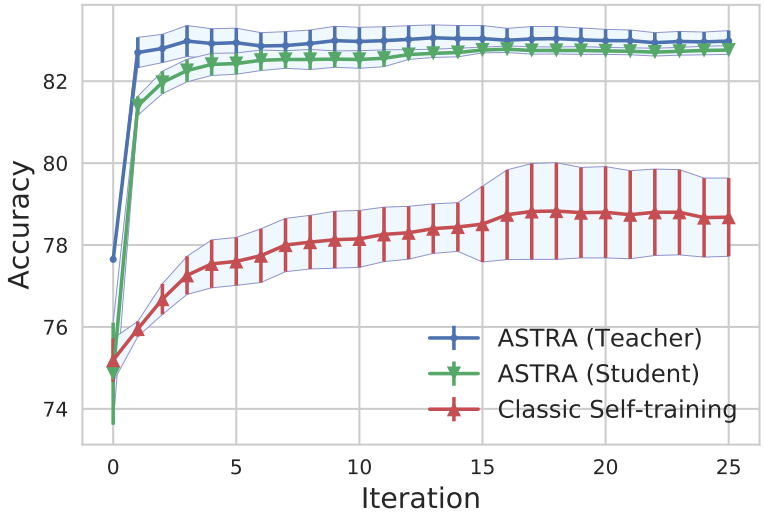


Figure 7.4: Gradual accuracy improvement over self-training iterations in the CENSUS dataset. ASTRA (Student) performs better than Classic Self-training (Student) being guided by a better teacher.

To better understand the benefits of our approach compared to classic self-training, consider Figure 7.4, which depicts the gradual performance improvement over iterations. The student models in classic self-training and ASTRA have exactly the same architecture. However, the latter is guided by a better teacher (RAN) that learns to aggregate noisy rules and pseudo-labels over unlabeled data.

Impact of rule sparsity and coverage for weak supervision. In this experiment, we compare the performance of various methods by varying the proportion of available domain-specific rules. To this end, we randomly choose a subset of the rules (varying the proportion from 10% to 100%) and train various weak supervision methods. For each setting, we repeat experiments with multiple rule splits and report aggregated results in Figure 7.5. We observe that ASTRA is effective across all settings with the most impact at high levels of rule sparsity. For instance, with 10% of domain-specific rules available, ASTRA outperforms ImplyLoss by 12% and Snorkel+Labeled by 19%.

This performance improvement is made possible by incorporating self-training in our

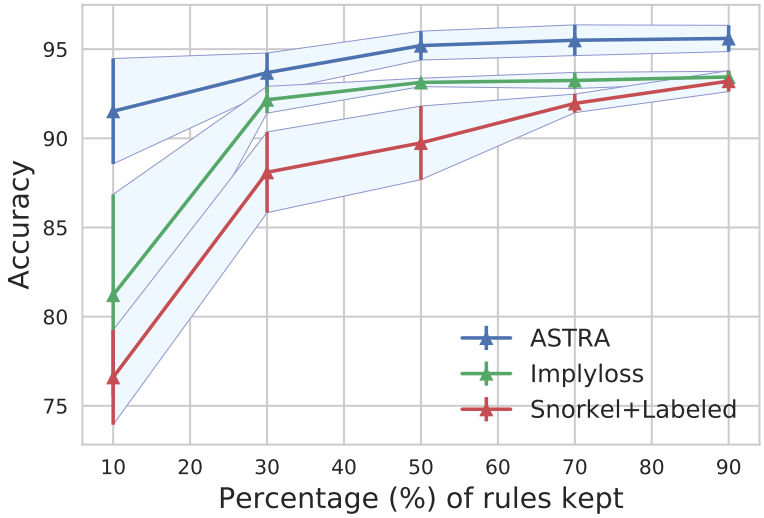


Figure 7.5: Performance improvement on increasing the proportion of weak rules in YouTube. For each setting, we randomly sample a subset of rules, aggregate and report results across multiple runs. ASTRA is effective across all settings with strongest improvements under high rule sparsity (left region of the x-axis).

framework to obtain pseudo-labels for task-specific unlabeled instances, and further re-weighting them with other domain-specific rules via the rule attention network. Correspondingly, Table 7.5 shows the increase in data coverage for every task given by the proportion of unlabeled instances that are now covered by at least two weak sources (from multiple rules and pseudo-labels) in contrast to just considering the rules.

Ablation study. Table 7.6 reports ablation experiments to evaluate the impact of various components in ASTRA.

ASTRA teacher marginally outperforms the student model on an aggregate having access to domain-specific rules. ASTRA student that is self-trained over task-specific unlabeled data and guided by an efficient teacher model significantly outperforms other state-of-the-art baselines.

Through minimum entropy regularization in our semi-supervised learning objective (Eq. (7.4)), ASTRA leverages the agreement between various weak sources (including rules and pseudo-labels) over task-specific unlabeled data. Removing this component results in an accuracy

% Overlap	**TREC**	**YTube**	**SMS**	**MITR**	**CEN.**	**Spouse**
Only Rules	46	48	9	1	94	8
ASTRA	**95**	**87**	**40**	**14**	**100**	**25**
Increase	+49	+39	+31	+13	+6	+17

Table 7.5: ASTRA substantially increases overlap (%) determined by the proportion of unlabeled instances that are covered by at least 2 weak sources (from multiple rules and student pseudo-labels, as applicable).

Configuration	**Acc**
ASTRA (Teacher)	**88.1**
ASTRA (Student)	87.7 (↓ 0.4%)
No min. entropy regularization in Eq. (7.4)	86.9 (↓ 1.4%)
No student fine-tuning on D_L (step 2.3)	86.7 (↓ 1.6%)
No student pseudo-labels in RAN in Eq. (7.2)	85.3 (↓ 3.2%)

Table 7.6: Summary of ablation experiments aggregated across multiple datasets.

Text	*What was President Lyndon Johnson 's reform program called ?*		
Clean Label	ENTY		
ASTRA Teacher	**ENTY**		
Weak Source	**Label**	**Weight**	**Feature / Regular expression pattern**
Student	**ENTY**	a=1.0	h_i (contextualized instance embedding)
Rule 8	**HUM**	a=1.0	`( |^)(who|what)[^\w] *(\w+ ){0,1}(person|man|woman|human|president)[^\w]*( |$)`
Rule 24	**ENTY**	a=1.0	`( |^)(what)[^\w]*(\w+ ){0,1}(is|is)[^\w]**([^\s]+ )*(surname|address|name)[^\w]*( |$)`
Rule 42	**DESC**	a=0.0	`( |^)(explain|describe|what)[^\w]*( |$)`
Rule 61	**HUM**	a=0.0	`( |^)(called|alias|nicknamed)[^\w]*( |$)`

Table 7.7: Snapshot of a question in TREC-6 and corresponding predictions. Top: instance text, clean label, and the aggregated prediction from ASTRA teacher. Bottom: several weak rules with regular expression patterns and predicted weak labels, along with the student and its pseudo-label (DESC: description, ENTY: entity, NUM: number, HUM: human). The weights depict the fidelity computed by RAN for each weak source for this specific instance.

drop of 1.4% on an aggregate demonstrating its usefulness.

Fine-tuning the student on labeled data is important for effective self-training: ignoring D_L in the step 2.3 in Algorithm 1, leads to 1.6% lower accuracy than ASTRA.

There is significant performance drop on removing the student's pseudo-labels ($p_\theta(\cdot)$) from the rule attention network in Eq. (7.2). This significantly limits the coverage of the

Instance Text (Question in TREC-6)	Teacher	Student	Set of Heuristic Rule Labels
1. *Which president was unmarried?*	**HUM**	HUM(1)	{}
2. *What is a baby turkey called?*	**ENTY**	DESC(1)	{ENTY(1), DESC(0), HUM(0)}
3. *What currency do they use in Brazil?*	**ENTY**	ENTY(1)	{DESC(0), DESC(0)}
4. *What is the percentage of water content in the human body?*	**NUM**	DESC(0)	{HUM(0), NUM(0.2), DESC(0)}

Table 7.8: Snapshot of answer-type predictions for questions in TREC-6 from ASTRA teacher and student along with a set of labels assigned by various weak rules (DESC: description, ENTY: entity, NUM: number, HUM: human) with corresponding attention weights (in parentheses). Correct and incorrect predictions are colored in green and red respectively.

teacher ignoring unlabeled instances that are not covered by rules, thereby, degrading the overall performance by 3.2%.

Case study: TREC-6 dataset. Table 7.7 shows a question in the TREC-6 dataset that was correctly classified by the ASTRA teacher as an "Entity" type (ENTY). Note that the majority voting of the four weak rules that cover this instance (Rule 8, 24, 42, and 61) leads to an incorrect prediction of "Human" (HUM) type. The ASTRA teacher aggregates all the heuristic rule labels and the student pseudo-label with their (computed) fidelity weights for the correct prediction.

Refer to Table 7.8 for more illustrative examples on how ASTRA aggregates various weak supervision sources with corresponding attention weights shown in parantheses. In Example 1 that is not covered by any rules, the student leverages the context of the sentence (e.g., semantics of "president") to predict the `HUM` label. While in Example 2, the teacher downweights the incorrect student (as well as conflicting rules) and upweights the appropriate rule to predict the correct `ENTY` label. In example 3, ASTRA predicts the correct label `ENTY` relying only on the student as both rules report noisy labels.

7.6 Conclusions

In this chapter, we presented a weak supervision framework, ASTRA, which efficiently trains classifiers by integrating task-specific unlabeled data, few labeled data, and domain-specific knowledge expressed as rules. We summarize the contributions of this chapter as

follows: (i) we presented an iterative self-training mechanism for training deep neural networks by augmenting the weak supervision signals with instances that are not covered by rules (Section 7.3.1); (ii) we presented a rule attention teacher network (RAN) for combining multiple rules and student model predictions with instance-specific weights conditioned on the corresponding contexts and constructed a semi-supervised learning objective for training RAN (Sections 7.3.2 and 7.3.3); and (iii) we evaluated our ideas by conducting an experimental evaluation on text classification benchmarks (Sections 7.4 and 7.5).

Our findings show that even simple self-training without human-provided rules sometimes outperforms existing weak supervision approaches that consider rules, highlighting the effectiveness of self-training with pre-trained models, which effectively leverage contextualized representations of instances. By combining the supervision signals from self-training and existing rules, our ASTRA framework improves data coverage by employing self-training with a student model that considers contextualized representations of instances and predicts pseudo-labels for all instances, leading to significant performance improvements over state-of-the-art weak supervision methods and over our self-training baseline. ASTRA is particularly stronger than other approaches at settings with high levels of rule sparsity, highlighting the promise of its effective adoption in emerging tasks with a limited number of human-provided rules. In the next chapter, we will show how the insights from ASTRA can help address even more challenging settings where no rules are available and will present an interactive method for getting human feedback on automatically generated labeling rules.

Chapter 8: Interactive Machine Teaching by Labeling Rules and Instances

In Chapter 7, we presented an architecture-agnostic framework for training text classifiers using labeling rules collected from humans. In practice, however, a complete set of accurate rules may be hard to obtain all in one shot as this requires substantial time, creativity, and foresight. In this chapter, we develop a method that guides human annotators *during* the teaching process with the goal to efficiently discover high-quality labeling rules. First, we motivate the problem of interactive machine teaching (Section 8.1). Second, we define our problem of focus and review related work (Section 8.2). Third, we present our human-in-the-loop framework for discovering accurate labeling rules for training deep neural networks (Section 8.3). We continue by describing our experimental evaluation for classification across several weak supervision benchmarks (Sections 8.4 and 8.5). Then, we present our new benchmarks to facilitate future research on machine teaching (Section 8.6). Finally, we conclude by summarizing the contributions of this chapter (Section 8.7).

8.1 Overview and Motivation

The machine teaching approaches discussed in the previous chapter work in two disjoint steps: (i) humans are asked to provide labeling rules; and (ii) labeling rules are used to train a machine learning model. All work discussed so far focuses on effective methods for addressing the second step of this process [Ratner et al., 2016; Ratner et al., 2017; Karamanolakis et al., 2019a; Bach et al., 2019; Awasthi et al., 2020]. However, there has been less effort to provide guidelines for and support humans in creating labeling rules. In practice, humans find it difficult to directly come up with sufficiently large sets of rules in

one shot [Varma and Ré, 2018]. Considerable time and creativity are required for inspecting unlabeled instances and creating rules that add predictive value by effectively covering a substantial number of instances. Therefore, it is important to also support the first step of the teaching process and provide guidelines and tools to assist humans in rule creation.

In this chapter, we focus on the earlier stages of a machine teaching task where the existing supervision signals are not sufficient to train an accurate machine learning model, and we investigate how to efficiently exploit a domain expert's limited time to collect sufficient supervision. Our main idea is to *automatically* extract labeling rules with non-negligible coverage of unlabeled data, and then rely on domain expertise to validate the candidate rules. Interaction with domain experts is important: in the absence of a large labeled dataset, automatically extracted rules could introduce too many wrong labels and have harmful effects on the model performance. Therefore, we assume that a domain expert will be able judge whether a candidate rule is accurate, similar to the assumption that an expert can create accurate labeling rules in the standard two-step approach. In contrast to active learning methods where the machine queries the human for labels of individual examples [Settles, 2009], providing feedback at the *rule* level can lead to several (albeit weak) labels, even within a single round of interaction. Interaction with rule level feedback can thus be more powerful than active learning.

Developing efficient frameworks with rich forms of interaction is challenging under this low-resource setting with limited teaching budget. First, given a restricted number of rules that can be created or validated by a human, it is not clear what properties these rules should have to train an accurate student model. For example, should one prioritize rules that cover more examples but with relatively lower precision or a few rules that have higher precision but lower coverage? Second, existing algorithms for rule extraction require many labeled data and it is not clear how to extract and rank candidate rules when we only have limited labeled data and few human-validated rules. Third, when provided the option to ask for feedback on both rules and instances, one must balance the costs and potential benefits

of each type of feedback when there is a shared budget of human interaction. For example, in some types of tasks it might be expensive to label long documents while there might be many good rules that can be labeled quickly. For other tasks, however, there might not be many accurate rules and therefore the time a human spends rejecting candidate rules might be better spent labeling more documents. In general, there are very few guidelines in the literature for creating effective rules for efficiently teaching machines.

To address these challenges, we perform an extensive analysis of existing datasets and propose a new human-in-the-loop framework. First, we analyze existing datasets that include human-defined rules and find patterns across datasets that could inform guidelines for rule creation. Second, we propose an adaptive interactive framework that assists human annotators by automatically creating candidate labeling rules, and effectively considers all the resources for training a classifier. To facilitate future research, we also propose new benchmarks for teaching machines with various types of supervision.

Our work presents the following contributions:

- We perform an extensive analysis of existing datasets that include human-defined rules and evaluate multiple weak supervision approaches by simulating low-resource rule settings, where just a subset of the human rules are considered in the teacher for training a student. We associate teacher properties with the student's performance and, even though rules are dataset-specific, we find prevalent patterns across datasets. For example, as we will see, a better teacher does not necessarily lead to a better student. Instead, the teacher's precision is more important than coverage for training an accurate student.

- We propose a new rule family with high-level rule predicates and present a method that extracts such rules using few labeled and many unlabeled data. In contrast to previous interactive approaches based on n-gram rules, our method extracts rules that can capture higher-level features. Furthermore, as we will demonstrate, these rules are highly effective.

- We present a human-in-the-loop machine teaching framework, namely INTERVAL,[1] which queries a human on both instances and rules and effectively uses all resources to train a classifier. We quantify the trade-off between labeling rules vs. instances and show that our framework is more efficient than existing work.
- We present new benchmarks to facilitate future research in machine teaching with different types of interaction.

We start with a review of related work on interactive machine teaching and define our problem of focus (Section 8.2). We continue as follows:

- We present an interactive machine teaching framework that adaptively queries a human for labeling rules and instances (Section 8.3).
- We evaluate our interactive method by conducting an experimental evaluation on multiple text classification datasets (Sections 8.4 and 8.5).
- We present new benchmarks for machine teaching with multiple rules and task instructions (Section 8.6).

The material described in Section 8.6 appears in [Zheng et al., 2022; Wang et al., 2022].

8.2 Problem Definition and Related Work

In this section, we define our problem of focus (Section 8.2.1) and describe related work on non-interactive weak supervision (Section 8.2.2), and interactive learning with instance- and feature-level feedback (Sections 8.2.3 and 8.2.4, respectively).

8.2.1 Problem Definition

Let $\mathcal{X}$ denote the instance space and $\mathcal{Y} = \{1, \ldots, K\}$ denote the label space for a K-class classification task. We consider a set of manually-labeled examples $D_L = \{(s_l, y_l)\}$, where

[1]INTERVAL: INTEractive Rule discoVery for weAkly supervised Learning.

$s_l \in \mathcal{X}$ and $y_l \in \mathcal{Y}$, and a set of unlabeled examples $D_U = \{s_i\}$. We also consider a set of pre-defined human-provided labeling rules $R = \{r^j\}$. A rule $r^j : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ maps an example s_i into a label $z_i^j \in \mathcal{Y} \cup \{\perp\}$. Predicting $z_i^j = \perp$ indicates that r^j does not cover s_i. Labeling rules can include rich procedures (e.g., capturing regular expression patterns in s_i) independently of the choice of a learning model, and can consider extra information (e.g., metadata, knowledge bases), which might not be available at test time. The goal of (non-interactive) weakly-supervised learning is to leverage D_L, D_U, and R to train a classifier that, given an unseen test instance $s' \in \mathcal{X}$, predicts a label $y' \in \mathcal{Y}$.

We are primarily interested in the case where the size of D_L is small in comparison to that of D_U and where R contains just a small number of human-provided rules. Additionally, we assume that we have a budget of T cost units (e.g., time, money) for querying a subject matter expert. Specifically, we assume that the expert can answer two different types of queries: (i) a query to provide a label $y_i \in \mathcal{Y}$ for an instance s_i at a cost of T_I (standard instance labeling); or (ii) a query to provide a label $z^j \in \mathcal{Y} \cup \{\perp\}$ for a candidate labeling rule r^j at a cost of T_R (rule labeling). Candidate labeling rules are defined by a boolean predicate $v^j(s_i)$ and a label $z^j \in \mathcal{Y} \cup \{\perp\}$, meaning that the rule predicts z^j for s_i if $v^j(s_i)$ is true, and otherwise predicts $\perp$. The candidate rule family is defined by the type of v^j, which can consider extra information, similarly to human-provided rules. Assigning the label z^j to r^j indicates the expert's intention to automatically label all the instances in D_U that are covered by r^j with a label z^j. Alternatively, the expert could skip the rule r^j by providing a label $z^j = \perp$, which indicates that no instances would be covered by r^j.

Our goal is to leverage D_L, D_U, and R, and interact with the expert within the specified budget T to train a classifier that, given an unseen test instance $s' \in \mathcal{X}$, predicts a label $y' \in \mathcal{Y}$. We would also like to consider the extreme low-resource setting where both $D_L = \emptyset$ and $R = \emptyset$, which is often the case in new tasks.

We now describe how previous work falls under this problem setting.

8.2.2 Non-Interactive Approaches

Non-interactive weak supervision approaches assume that $T = 0$, in other words, there is no human in the loop. Supervised learning methods consider just D_L, semi-supervised learning methods consider D_L and D_U, and weakly-supervised learning methods consider D_L, D_U, and R [Ratner et al., 2017; Bach et al., 2019; Badene et al., 2019; Fu et al., 2020; Awasthi et al., 2020]. Note that our ASTRA method (Chapter 7) also falls under this category. In addition to the weakly-supervised learning methods described so far, our method is related to prompt-based fine-tuning [Schick and Schütze, 2021; Perez et al., 2021]. Prompt-based techniques convert the classification task into a cloze-style task and leverage pre-trained language models to "answer" the cloze-style question. By directly using the outputs of the pre-trained language models for classification, prompt-based techniques are sensitive to the selection of prompts [Gao et al., 2021], labeled examples [Zhao et al., 2021; Perez et al., 2021], and other hyperparameters [Tam et al., 2021]. Our work explores prompt-based approaches to construct labeling rules, which we assume are only weakly indicative of the true labels. Additionally, we show that prompt-based rules that are extracted automatically by our method can be highly effective for machine teaching.

8.2.3 Interactive Learning with Instance Feedback

One type of interaction that has been studied extensively in the literature is active learning, in which the machine queries the human for just a small number of labels for examples that are chosen adaptively from abundant unlabeled data [Lewis and Gale, 1994; Cohn et al., 1996; Roy and McCallum, 2001; Dasgupta et al., 2007; Dasgupta and Hsu, 2008; Settles, 2009; Beygelzimer et al., 2010; Houlsby et al., 2011; Zhang and Chaudhuri, 2015; Shen et al., 2017; Kirsch et al., 2019; Ash et al., 2019; Brantley et al., 2020; Yuan et al., 2020a; Margatina et al., 2021]. Nearly all previous active learning methods solicit the expert's judgment to just label instances. In other words, they assume that $T_R = \infty$ (i.e., these approaches do not allow queries about labeling rules) and query the expert about $\lfloor \frac{T}{T_I} \rfloor$ instance labels.

Soliciting labels for a single instance at a time, requires multiple interactions to create a sufficiently large training set. On the other hand, adding a new *rule* could lead to weak labels for *many* examples (i.e., all the examples that are covered by the rule) and as a result, a large weakly-labeled dataset can be created with a relatively smaller number of rules.

8.2.4 Interactive Learning with Rule Feedback

Our work is related to previous interactive methods that support expert queries on automatically generated rules from the n-gram family [Druck et al., 2008; Melville et al., 2009; Settles, 2011; Jagarlamudi et al., 2012; Poulis and Dasgupta, 2017; Dasgupta et al., 2018; Boecking et al., 2020]. These methods extract simple candidate rules based on n-grams appearing in s_i. As we will show, n-gram based candidate rules have limited effectiveness and different characteristics than human-provided rules in R. One exception is [Zhang et al., 2022b], which considers rules based on the output of pre-trained language models prompted with task-specific templates and shows that humans can successfully provide feedback on rules from such family. Most of the above methods do not allow instance-labeling queries (i.e., these methods assume that $T_I = \infty$). In contrast, our method attempts to unify active learning with rule labeling by querying a human for both instances and rules from a new rule family with high-level features.

8.3 Interactive Machine Teaching with Instance and Rule Feedback

In this section, we describe our interactive machine teaching framework to address the problem defined in Section 8.2.1. The core question is, given a limited budget T for interaction with a domain expert, how to solicit the expert's feedback efficiently to teach a learning algorithm. Our framework, namely INTERVAL, interacts with humans via queries on both instances and automatically discovered rules, and uses all the available resources for weakly-supervised learning. INTERVAL can be used with several different methods for weakly supervised learning and any learning model.

As an important component of INTERVAL, we propose a method that uses D_L and D_U to automatically extract candidate rules, which capture a rich family of features beyond n-grams and can cover many instances in D_U. Given limited resources, however, our method can extract "noisy" rules that predict the wrong labels for many instances in D_U and could have a negative impact if added on R. By supporting rule queries, INTERVAL can help exploit the candidate rules that are considered accurate by the expert, and can discard rules that are noisy. Moreover, by supporting instance queries, INTERVAL can augment D_L, which is essential for "denoising" rules and for training the learning model, and can effectively be applied for tasks where there might not be many good rules. To efficiently interact with a human within a budget T, we design a method that adaptively chooses which instances and rules to query for feedback.

In the rest of this section, we describe the individual steps followed by INTERVAL on each iteration, namely teacher-student co-training (Section 8.3.1), querying for instance feedback (Section 8.3.2), candidate rule extraction (Section 8.3.3), and querying for rule feedback (Section 8.3.4), and then we summarize the main ideas of our interactive machine teaching algorithm (Section 8.3.5).

8.3.1 Teacher-Student Co-Training

In the first step of each iteration, we use D_L, D_U, and R to train a learning model. This has been the main objective in non-interactive weakly-supervised learning. Here, we unify a class of several weakly-supervised learning methods [Dawid and Skene, 1979; Ratner et al., 2016; Ratner et al., 2019; Karamanolakis et al., 2021; Zhang et al., 2022a] by employing the teacher-student abstraction from Chapter 7.

The teacher model $q_\phi(\cdot)$ considers D_L, D_U, and R, and predicts labels q_i for all examples $s_i \in D_U$ except for examples covered by no rules in R, which are not covered by the teacher either. Similar to Chapter 7, the student model $p_\theta(\cdot)$ is the base learning model that is

trained using D_L, D_U, and the teacher:

$$\min_{\theta} \ \mathbb{E}_{s_l, y_l \in D_L}[-\log \ p_\theta(y_l \mid s_l)] + \lambda \mathbb{E}_{s \in D_U} \mathbb{E}_{y \sim q_{\phi^*}(y|s)}[-\log \ p_\theta(y \mid s)], \tag{8.1}$$

where $\lambda \in \mathbb{R}$ is a hyper-parameter controlling the relative importance of the manually labeled data (first term) and the weakly labeled data (second term). The above teacher-student abstraction models different approaches for weakly-supervised learning [Zhang et al., 2022a]. For example, in simple majority voting, the teacher aggregates the predictions of rules in R. In Snorkel [Ratner et al., 2017], the teacher is a probabilistic graphical model that estimates weights for all rules in R. In our ASTRA method from Chapter 7, the teacher is our Rule Attention Network (Section 7.3.2) that is iteratively co-trained with the student.

In our problem of focus, where the size of D_L is small and R contains just a small number of rules, the student model might have far less than satisfying accuracy for our target task. Especially in extreme low-resource settings, where both $D_L = \emptyset$ and $R = \emptyset$, the student does not cover any examples in D_U and as a result, we define the student as a classifier that predicts classes randomly. Fortunately, we additionally have a budget T to interact with an expert, which we exploit as discussed next.

8.3.2 Querying for Instance Feedback

After having trained the student, INTERVAL queries the label y_i for an instance s_i from the unlabeled set D_U. To efficiently interact with a human, we design a method that adaptively chooses which instance to query for feedback, as some instances might be more "informative" than others. First, we identify a diverse collection of unlabeled instances for which the student's predicted probabilities have high entropy. To do this, we hierarchically cluster the unlabeled data D_U and then use the active learning algorithm of [Dasgupta and Hsu, 2008] to select a sample instance in a cluster-adaptive manner as guided by the aforementioned entropy heuristic. After having selected an instance s_i, the system queries

the expert's label y_i at a cost of T_I. At the end of the iteration, the labeled pair (s_i, y_i) will be added in D_L with the hope to train a better teacher and student at the next iteration.

8.3.3 Candidate Rule Extraction

After getting the label y_i for an unlabeled instance s_i, our framework extracts candidate rules r^j that cover s_i. One reason behind the student being uncertain for an instance s_i is that it captures multiple implicit "rules" with conflicting labels. In this case, identifying the correct rule could improve the student in the next iteration by augmenting its training data with all the unlabeled examples covered by the rule. Our method uses D_U and D_L to extract a pool R_C of candidate labeling rules that cover s_i. We first describe the types of rules and then how to extract them.

Rule family. As described in Section 8.2.1, candidate labeling rules are defined by a boolean predicate $v^j(s_i)$ and a label z^j. Our method extracts rules r^j whose predicates $v^j(s_i)$ are disjunctions of features that can have three different types:

- n-grams: $v^j(s_i)$ is true if a specific n-gram appears in s_i.
- Linguistic features: we extract linguistic features such as part-of-speech tags and named entities from s_i and then, we construct rules based on the counts of such features. In spam classification for example, our system extracts a rule that classifies s_i as "Spam" if at least two entities of type "MONEY" appear in s_i.
- Prompt-based features: we extract features from the outputs of a pre-trained language model. First, we use task-specific templates with a "[MASK]" token [Bach et al., 2022] to prompt a pre-trained language model and extract features as the top-k tokens with the highest predicted probabilities. Note that these extracted tokens do not need to appear in s_i.

While most work on interactive learning with rule feedback has focused on n-gram features, additionally leveraging linguistic and prompt-based features lets us find common patterns across instances that might not even share any n-gram features, such as in tasks with short documents. As we will show, our rule family extracts more accurate rules than the n-gram rules considered in most previous interactive methods, and thus, rules from our family are promising to improve the overall effectiveness of our machine teaching method. Note that, at test time, our method does not require access to the above resources (e.g., tools for extracting linguistic features, pre-trained model to extract prompt-based features) as the student model predicts labels directly based on s_i.

Rule extraction. We extract rules r from the above family, which cover at least t_{cov} examples in D_U *including* s_i, and which have a precision of at least t_{prec} in D_L. Both t_{cov} and t_{prec} are hyper-parameters in our framework. Given the above coverage and precision constraints, we extract disjunctions of high-level features using the Apriori algorithm [Agrawal et al., 1994]. Specifically, we first exhaustively search all rules with a single feature from the above family and keep all rules that satisfy all constraints. (The constraint that all rules have to cover s_i is especially strong and allows efficient search.) Then, we create rules as disjunctions of two features selected before and select just the resulting rules that satisfy all the above constraints. Our method considers rules with disjunctions of up to t_{len} features, where t_{len} is a hyper-parameter. The set R_C contains all candidate rules that are extracted by our method and satisfy our constraints.

Automatically identifying a good rule is hard in our setting with limited labeled data D_L. For example, a candidate rule r^j with high coverage on D_U might have low coverage in D_L (D_L might contain just a few labeled examples), and therefore it is hard to estimate the true precision of r^j. Therefore, we rely on human feedback for selected candidate rules from R_C, as discussed next.

8.3.4 Querying for Rule Feedback

After having extracted the set of R_C candidate rules that cover s_i, our framework selects up to β candidate rules r^j and queries for their labels z^j, where β is a hyper-parameter. Specifically, we first select in R'_C all rules from R_C that predict a label $z^j = y_i$ (thus agreeing with the expert's label for s_i) and then, we select from R'_C the top β rules with the highest precision. Note that R'_C might have fewer than β rules in total, thus we use $\beta_i \leq \beta$ to indicate the actual number of rules selected by our algorithm.

Next, we query the labels z^j for the β_i selected rules at a cost of $\beta_i T_R$. At the end of the iteration, the β_i labeled rules, which we denote as $\{(r^j, z^j)\}_{\beta_i}$, will be added in R with the hope to train a better teacher and student at the next iteration. Our method will ignore rules labeled with $z^j = \perp$.

Through this interaction design, we assume that the domain expert can judge whether r^j provides the correct label for most of the examples that the rule covers, and is aware that (i) a rule r^j does not need to have perfect accuracy but rather represents a pattern that the expert intends to exploit to label examples more efficiently than by hand labeling; (ii) rule predictions will be aggregated to train a model with a noise-aware way.

8.3.5 Interactive Machine Teaching Algorithm

Building upon the previous ideas, we present our interactive method for machine teaching (Algorithm 3). First, our method clusters D_U into hierarchical clusters (line 1) and creates a pool of candidate rules (line 2). Each round of our interactive approach consists of the following steps: (1) we train the teacher and student using labeled data, unlabeled data, and human-validated rules (line 3.1); (2) we apply the student on unlabeled data to get soft labels (line 3.2); (3) we pick a candidate unlabeled instance (line 3.3) and obtain its label from a human (line 3.4); (5) we extract candidate rules (line 3.5) and obtain the labels for β_i rules from a human (line 3.6); and (6) we update the set of labeled data, the set of human-validated rules, and the remaining budget (line 3.7). We repeat this procedure until

Algorithm 3 Interactive Machine Teaching

Input: Small amount of labeled data D_L; task-specific unlabeled data D_U; small set of weak rules R; budget of T cost units for interaction with a subject matter expert
Outputs: Student $p^*_\theta(\cdot)$, Teacher $q^*_\phi(\cdot)$, augmented labeled data D'_L, augmented set of weak rules R'

1: Cluster all data $s_i \in D_U$ into hierarchical clusters (agglomerative clustering; Ward's linkage; Euclidean distance of instance embeddings)
2: Initialize $D'_L = D_L$, $R' = R$
3: **Repeat until the budget T runs out:**
  3.1: Train teacher $q^*_\phi(\cdot)$ and student $p_\theta(\cdot)$ using D_L, D_U, R
  3.2: Apply $p_\theta(\cdot)$ to $s \in D_U$ to obtain soft labels: $D_{Student} = \{(s_i, \mathbf{p}_i)\}_{s_i \in D_U}$
  3.3: Pick a candidate instance $s_i \in D_U$
  3.4: Query the label y_i for s_i (cost $= T_I$)
  3.5: Extract candidate rules r^j that cover s_i
  3.6: Query the labels z^j for β_i rules r^j (cost $= \beta_i T_R$)
  3.7: Update $D'_L = D'_L \cup \{(s_i, y_i)\}_{\beta_i}$, $R' = R' \cup \{r^j : (v^j(\cdot), z^j)\}$, $T = T - T_I - \beta_i T_R$

the budget runs out.

By associating r^j with a specific instance s_i, we give the expert extra context (e.g., the text of s_i) for deciding z^j. Also, we hypothesize that, in practice, reading the text of the instance can help reduce the cost T_R for deciding z^j. In fact, previous work with n-gram based rules assumes that $T_R = 0$, i.e., labeling rules comes at "no extra cost" in this sequential type of interaction [Poulis and Dasgupta, 2017]. In our evaluation, we assume that $T_R > 0$. The hyper-parameter β_i can control how to distribute the budget T. Specifically, setting $\beta_i = 0$ reduces to standard active learning, as INTERVAL will perform $\lfloor \frac{T}{T_I} \rfloor$ queries on instances only. By setting $\beta_i \geq 1$, one can exploit feedback on rules that apply to s_i. As we will show, rule feedback leads to performance improvement compared to instance feedback only.

Discussion. Different options could be considered for our framework's components.

When querying for instance feedback (Section 8.3.2), different active learning strategies could be deployed. In INTERVAL, an instance s_i is selected regardless of the rules that cover

	YouTube	SMS	IMDB	Yelp	TREC	AGNews
Classification task	spam	spam	sentiment	sentiment	question type	topic
Domain	user comments	text messages	movies	reviews	web queries	news
# Classes (K)	2	2	2	2	6	4
# All train instances	1586	4571	20,000	30,400	4965	96,000
# $\lvert D_U \rvert$	1546	4531	19,960	30,360	4725	95,840
# $\lvert D_L \rvert$	40	40	40	40	240	160
# Test instances	250	500	2500	3800	500	12,000
# Prompt templates	5	5	15	12	6	9
# Human-provided rules	10	73	5	8	68	9
Rule coverage	87%	40%	86%	81%	95%	65%

Table 8.1: Statistics for available datasets with human-labeled rules.

s_i. In the future, it would be interesting to adaptively choose which instance s_i to query for feedback by also considering the fact that s_i will be used to get rule feedback. Another interesting option would be to provide the option to selectively skip instance feedback and perform rule queries only.

Our rule extraction method (Section 8.3.3) considers a simple family of rules, which predict the same label for all instances that they cover. In the future, it would be interesting to explore altenative rule families and consider alternative ways to select candidate rules, for example, based on rule diversity criteria.

When querying for rule feedback (Section 8.3.4), our approach assumes that the expert can only provide a label z^j for a rule r^j, while it could be beneficial to allow more expressive types of feedback, such as editing r^j to make it more accurate. Also, INTERVAL assumes that all rules (or instances) have the same cost T_R (T_I). In practice, different rules (or instances) might have different costs.

8.4 Experimental Settings

We now present our experimental setting for interactive machine teaching on several text classification datasets.

Datasets. For our analysis and to evaluate our framework, we consider six benchmark datasets with human-made rules that are provided by [Zhang et al., 2021]:

- **YouTube:** Binary ("Spam" vs. "Not Spam") classification of YouTube comments [Ratner et al., 2017].[2] YouTube has 5 keyword-based (coverage=48%), 1 regular expression-based (coverage=23%), 1 length-based (coverage=23%), and 3 classifier-based (coverage=46%) rules.

- **SMS:** Binary ("Spam" vs. "Not Spam") classification of SMS messages [Ratner et al., 2017]. SMS has 16 keyword-based (coverage=4%) and 57 regular expression-based (coverage=38%) rules.

- **IMDB:** Binary ("Positive" vs. "Negative") classification of IMDB movie reviews [Maas et al., 2011]. IMDB has 5 keyword-based rules [Ren et al., 2020].

- **Yelp:** Binary ("Positive" vs. "Negative") classification of Yelp business reviews [Zhang et al., 2015]. Yelp has 8 keyword-based rules [Ren et al., 2020].

- **TREC:** Question classification from TREC-6 into 6 categories: "Abbreviation," "Entity," "Description," "Human," "Location," and "Numeric Value." TREC has 13 keyword-based (coverage=62%) and 55 regular expression-based (coverage=57%) rules [Awasthi et al., 2020].

- **AGNews:** Topic classification of news documents into 4 topics: "World," "Sports," "Business," and "Science/Technology" [Zhang et al., 2015]. AGNews has 9 keyword-based rules [Ren et al., 2020].

Table 8.1 reports dataset statistics. Datasets come from diverse domains and the rules have different types and characteristics. To create our prompt-based rules, we use prompt templates available in [Bach et al., 2022]. For example, Yelp has 12 human-written prompt templates, including the following: "*Overall, the experience is* [MASK]}. [TEXT]}.", where "[MASK]" is the token to be predicted by the pre-trained language model and "[TEXT]"

[2] `https://archive.ics.uci.edu/ml/machine-learning-databases/00380/YouTube-Spam-Collection-v1.zip`

is replaced by the text of s_i. Next, we describe our experimental procedure for simulating low-resource settings.

Experimental procedure. To simulate the low-resource setting for each dataset, we split the training examples into D_L and D_U by sampling 20 labeled examples per class uniformly at random, which we use in D_L, while we use the rest in D_U. We sample the same number of examples from the validation set to be consistent with our low-resource assumptions. For interactive approaches, we consider the extreme low-resource setting where $R = \emptyset$. We simulate human feedback using all labels in D_U: a candidate rule is accepted if it correctly classifies more than $t_{oracle}\%$ of the instances in D_U that it covers. We experiment with different values of t_{oracle}: 25%, 50%, 75%, 90%, and 100%.

For a robust evaluation, for each method we run 10 different experiments with different random seeds, thus each run corresponds to a different version of D_L, D_U, and R. We report the average test performance over the 10 different runs. As evaluation metric, we use the macro-averaged F1 of the student model on the test set.

Model configuration. For a fair comparison, we use exactly the same text pre-processing (tokenization, embedding) as in [Zhang et al., 2021]. We represent each segment as a vector using pre-trained BERT [Devlin et al., 2019], similarly to previous chapters, for all datasets except TREC, where we found that tf-idf weighted bag-of-words representations are more effective. For student training, we experiment with multiple values for the relative weight of manual and weakly labeled data ($\lambda \in \{0, 0.01, 0.1, 1.0\}$). For instance queries, we cluster data into hierarchical clusters via agglomerative clustering[3] by minimizing cluster variances (Ward's linkage), where Euclidean distances are computed based on instance embeddings.

Table 8.2 summarizes types of features extracted by our rule extraction module. We extract n-grams with $n = 1, 2, 3$, linguistic features (part-of-speech tags and named entities)

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html`

Name	Feature Types
n-grams	unigrams, bigrams, trigrams
Named entities	"WORK OF ART," "CARDINAL," "FAC," "LOC," "EVENT," "LAW," "PERSON," "ORDINAL," "NORP," "PERCENT," "LANGUAGE," "ORG," "QUANTITY," "TIME," "DATE," "MONEY," "PRODUCT," and "GPE."
POS tags	"ADJ," "ADP," "ADV," "AUX," "CCONJ," "DET," "INTJ," "NOUN," "NUM," "PART," "PRON," "PROPN," "PUNCT," "SCONJ," "SYM," "VERB"
Prompt features	Top 10 tokens predicted by BERT for each task-specific prompt template

Table 8.2: Types of features considered by our rule extraction module.

using the spaCy library[4]. We extract prompt-based features as the top 10 tokens predicted by pre-trained BERT for each of the templates provided by [Bach et al., 2022][5]. We consider disjunctions of up to $t_{len} = 3$ features and experiment with different values for the minimum coverage on D_U ($t_{cov} \in \{10, 100, 1000\}$) and the minimum precision based on D_L ($t_{prec} \in \{25\%, 50\%, 75\%, 100\%\}$). For interaction, we set $\beta = 1$ and unless otherwise mentioned assume $T_R = T_I$. We leave the study of higher values of β and different values of T_R for future work.

Model comparison. For a robust evaluation of our approach, we compare several approaches that utilize different resources:

- **"Fully supervised":** a fully-supervised learning method trained in the high-resource setting using *all* labeled data. This approach is not directly comparable with other methods that are trained with limited labeled data or rules.
- **"Low supervised":** a supervised learning baseline trained in the low-resource setting using only D_L.
- **"Semi supervised":** a semi-supervised learning method trained using D_L and D_U. Here, we consider self-training [Nigam and Ghani, 2000; Lee, 2013] for up to 25 iterations with early stopping based on the performance on the validation set (note that

[4] `https://spacy.io/usage/linguistic-features/`

[5] Prompt templates are available at `https://github.com/bigscience-workshop/promptsource`.

this is similar to our baseline in Section 7.4).

- **"Weakly supervised":** a weakly-supervised learning method trained using D_L, D_U, and R. We experiment with different methods, including unweighted majority voting and weighted aggregation of rule predictions with different techniques [Ratner et al., 2017; Ratner et al., 2019; Fu et al., 2020; Karamanolakis et al., 2021].

- **"Active Instances":** active learning methods that use D_L, D_U and spending all the interaction budget T to perform $\lfloor \frac{T}{T_I} \rfloor$ queries on instances only. We experiment with different acquisition functions for active learning, including random instance selection, uncertainty-based sampling, hierarchical sampling [Dasgupta and Hsu, 2008], and contrastive active learning [Margatina et al., 2021].

- **"Active Rules":** interactive machine teaching methods that use D_L, D_U and spending the interaction budget T to perform queries on just rules. In the future, we plan to evaluate IWS [Boecking et al., 2020], which considers n-gram rule families and can be applied only for binary classification, and PRBoost [Zhang et al., 2022b], which considers prompt-based rules.[6]

- **"INTERVAL"**: Our interactive machine teaching method that uses D_L, D_U and spends the interaction budget T to perform queries on both instances and rules.

For a fair comparison, we use exactly the same text pre-processing (tokenization, embedding) as in [Zhang et al., 2021] across all methods. We represent each segment as a vector using pre-trained BERT [Devlin et al., 2019] (similarly to previous chapters) for all datasets except TREC, where we found that tf-idf weighted bag-of-words representations are more effective.

[6]Unfortunately, the code repository for PRBoost [Zhang et al., 2022b], `https://github.com/rz-zhang/PRBoost`, does not contain any code as of July 2022.

8.5 Experimental Results

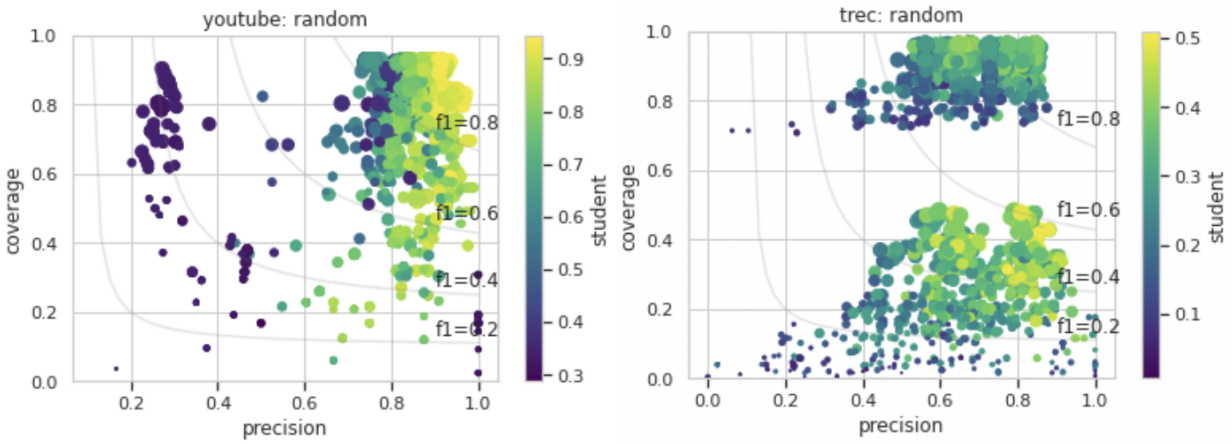
We now present our analysis of human-provided rules (Section 8.5.1), results on automatic rule extraction (Section 8.5.2), and our experiments for interactive machine teaching with queries on instances and rules (Section 8.5.3).

8.5.1 Analysis of Human-Provided Rules

In this section, we analyze existing datasets with human-labeled rules and simulate low-resource rule settings to understand the impact of properties on the performance of the student model. Our goal is to discover patterns across datasets that could influence the design of guidelines for rule creation.

Analysis of the precision vs. coverage trade-off. In Section 8.1, we highlighted one challenging question: should one prioritize rules that cover more examples but have a relatively lower precision or a few rules that have higher precision but lower coverage? To analyze the precision-coverage trade-off, we create different Teacher versions using different subsets of the human-labeled rules and evaluate the performance of Student using each Teacher separately. For a robust analysis, we evaluate multiple Teacher types (majority voting, Snorkel [Ratner et al., 2016], Dawid Skene [Dawid and Skene, 1979], MeTaL [Ratner et al., 2019], FlyingSquid [Ratner et al., 2019]), and multiple Student types (bag-of-words logistic regression, multilayer perceptron, BERT). For each Teacher type, we keep different randomly-selected subsets of the rules in R ranging from 1% to 100%. For each resulting Teacher-Student combination, we run 10 different experiments with different random seeds. This results to more than 1,000 different Teacher-Student configurations for each dataset.

Figure 8.1 summarizes the results across all experiments for YouTube, Yelp, and TREC. While different datasets have Teacher-Student pairs with different characteristics, there are patterns that are prevalent across datasets. First, a better Teacher does not necessarily lead to a better Student. For example, in Youtube (Figure 8.1a) there exist Teachers with



(a) YouTube. (b) TREC.

Figure 8.1: Precision-coverage scatterplots reporting the precision (x-axis) and coverage (y-axis) of the teacher. Each data point corresponds to a different Teacher-Student pair and its color indicates the F1 score of the student.

	YouTube	**SMS**	**Yelp**	**IMDB**	**TREC**	**AGNews**
Coverage weight	0.20	0.00	0.22	0.23	0.30	0.46
Precision weight	**0.80**	**1.00**	**0.78**	**0.77**	**0.70**	**0.54**

Table 8.3: Quantifying the relative importance of Teacher coverage and precision for training an accurate Student. Across all datasets, precision is more important than coverage.

F1 ≥ 0.6 that train a Student with F1 ≥ 0.5 while other Teachers with F1 ≤ 0.2 train a student with F1 ≥ 0.8. This result implies that naively optimizing the teacher's performance (according to the standard "data programming" paradigm [Ratner et al., 2016]) might not lead to the best performing student model, so to efficiently teach the student, a new strategy is required.

As another contribution of this work, we identify that Teacher's precision is more important than coverage for training an accurate Student. In the scatterplots of Figure 8.1, most Teachers with high precision train high-quality Students, while many Teachers with high coverage train low-quality Students. To quantify this observation, we compute precision-coverage weights using the teacher's precision and coverage to predict the student's F1 score. Specifically, we compute the student's F1 score as the weighted geometric average of the teacher's precision and coverage and we tune the corresponding weights using grid

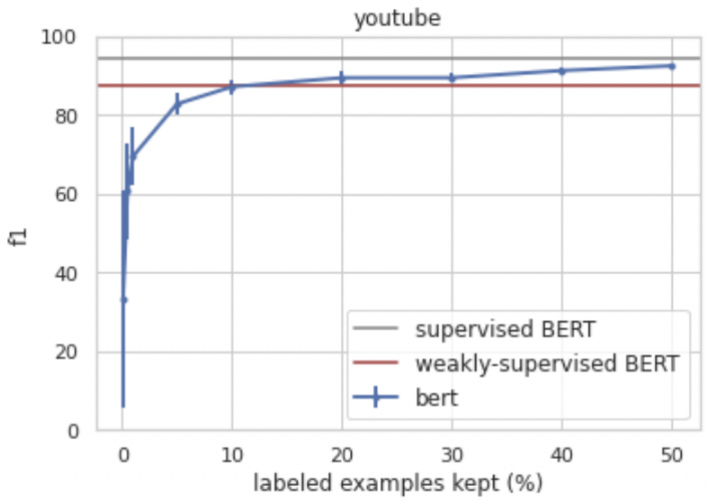


Figure 8.2: Supervised learning results in YouTube by varying the labeled data sizes ($|D_L|$). "Low Supervised" BERT matches the performance of "Weakly Supervised" BERT (trained with 10 rules) when $|D_L| = 10\% = 160$. Thus, on average, 1 rule is worth 16 labeled examples.

search. A higher weight thus indicates that the corresponding feature is more important for the prediction of the student's F1 score. Table 8.3 shows the estimated precision and coverage weights for all datasets. Across all datasets, precision is more important than coverage: more precise Teachers lead to more accurate Students. Note that this experiment was performed with human-provided rules that have non-trivial coverage and precision.

Our observation that rule precision is more important than coverage explains recent design choices for weakly supervised learning [Awasthi et al., 2020; Hsieh et al., 2022], such as the "contextualized LF modeling" component of [Hsieh et al., 2022], which explicitly reduces rule coverage to improve rule precision. Moreover, our observation could potentially inform guidelines for rule creation. In YouTube, for instance, if we reject all teacher models with coverage lower than 0.5, then the precision's importance weight increases from 0.75 to 0.84, indicating that focusing on precision would be beneficial. Therefore, one potential guideline is that if the teacher has a coverage higher than 50%, then the main focus should be on improving its precision.

	YouTube	SMS	Yelp	IMDB
1 rule = x labeled examples	x=16	x=6	x=38	x=160

Table 8.4: Quantifying the relative value between human-provided rules and labeled examples.

Analysis of the relative benefits of rules and labeled data. Given a limited budget T for interaction with a domain expert, should one label rules or individual instances? We first explore this question by analyzing the relative performance of the student model trained with rules vs. labeled data. Specifically, we evaluate the "Low supervised" approach by varying the amount of labeled data in D_L (selected randomly) and compare its performance to the "Weakly supervised" approach trained using *all* human-provided rules R and no labeled data (i.e., $|D_L| = 0$). Figure 8.2 shows the results on the YouTube dataset where $|R| = 10$. "Weakly supervised" trained with 10 rules is matched by "Supervised" trained with 10% of the labeled examples (= 160 labeled examples). In other words, on average, one rule is worth 16 labeled examples. Table 8.4 summarizes the corresponding results for YouTube, SMS, Yelp, and IMDB. As expected, the relative value of rules and labeled examples varies per dataset and depends on several factors, such as rule quality, task difficulty, and model quality. In IMDB, one rule is worth 160 labeled examples and thus a single interaction with a human could save 160 data labelings. On SMS, one rule is worth 6 labeled examples, so if the cost of rule creation is more than 6 times higher than that of data labeling, it would be more efficient to label instances than create rules.

As a caveat of our analysis, we might have underestimated the relative benefits of labeled data. Specifically, the value of instance labeling might be higher if labeled instances were selected based on their informativeness (instead of randomly). Also, labeled instances might lead to better performance if few-shot (instead of standard supervised) learning approaches were adopted [Pan and Yang, 2009; Hospedales et al., 2021]. Therefore we identify two important and challenging directions. First, it is important to develop methods for automatically extracting rules that are worth many examples. Second, it is important to develop

Dataset	Template Name	Template
Yelp	EXPERIENCE	Overall, the experience is [MASK]. [TEXT].
Yelp	RECOMMEND	[TEXT]. Would I recommend it? The answer is [MASK].
Yelp	RATING	[TEXT]. On a scale of 1 to 5, I would give this place a [MASK]
SMS	ASKS_FOR	The following SMS message asks for [MASK]: [TEXT]
SMS	IS_ABOUT	The following SMS message is about [MASK]: [TEXT]

Table 8.5: Examples of templates used to prompt pre-trained language models.

adaptive methods that can take advantage of both rules and labeled examples and balance the trade-off of their relative costs. Next, we continue with the analysis of automatically extracted rules and then we evaluate our interactive machine teaching method.

8.5.2 Analysis of Automatically Extracted Rules

In Section 8.3.3, we showed how to automatically extract rules with high-level features (n grams, linguistic features, and prompt-based features). In this section, we show examples of automatically extracted rules and compare our rule family to a simpler family of n-gram rules and to human-provided rules.

Table 8.5 shows examples of templates used by our method to extract prompt-based rules. Table 8.6 shows examples of rules extracted by our method. Such rules can have higher coverage and precision than n-gram rules.

Figure 8.3 shows precision-coverage scatterplots for rules automatically extracted by our method. For this analysis, we have included all rules regardless of their coverage and precision (i.e., $t_{prec} = 0$ and $t_{cov} = 0$), thus explaining the symmetry in the plots: symmetric data points are rules that have the same predicate but predict different classes. Rules with high-level predicates can achieve relatively high precision and coverage compared to n-gram predicates and thus are promising to improve the overall performance of interactive machine teaching.

Table 8.7 reports the performance of the "Weakly Supervised" with automatically extracted rules extracted by our method using $t_{cov} = 100$ (minimum rule coverage), $t_{prec} = 0.75$ (minimum rule precision). Across all datasets, rules based on high-level predicates are more

Dataset	Rule Predicate	Rule Label
Yelp	PROMPT-EXPERIENCE=“appalling”	Negative
Yelp	PROMPT-EXPERIENCE=“terrible”	Negative
Yelp	PROMPT-EXPERIENCE=“fantastic”	Positive
Yelp	PROMPT-RECOMMEND=“certainly”	Positive
Yelp	PROMPT-RATING=“five”	Positive
Yelp	PROMPT-RATING=“one”	Negative
SMS	PROMPT-IS_ABOUT=“prizes”	Spam
SMS	NGRAM=“http” AND PROMPT-ASKS_FOR=“donations”	Spam
SMS	SPACY-NER=“CARDINAL” AND PROMPT-ASKS_FOR=“information”	Spam

Table 8.6: Examples of rules extracted by our method. “NGRAM=a” means that a appears as an n-gram in the text. “SPACY-NER=a” means that SpaCy extracts at least one entity of type a from the text. “PROMPT-b=a” means that a appears in the top-k tokens predicted by the pre-trained model to fill in the [MASK] token for template b.

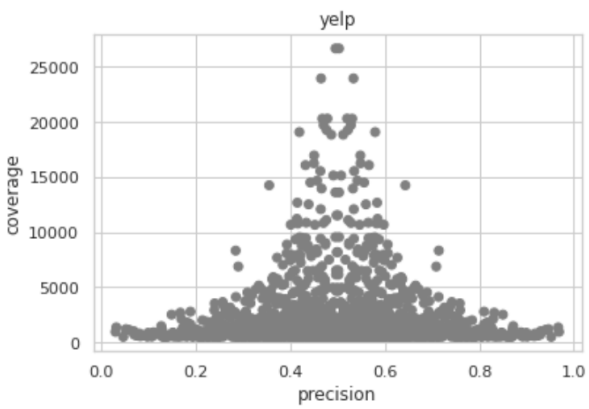
Rule family	YouTube	SMS	IMDB	Yelp	TREC	AGNews	AVG
human	90.0	86.8	71.2	80.2	57.0	75.9	76.8
n-gram	76.4	79.7	49.1	54.9	52.7	74.8	64.6
high-level	**82.7**	**91.4**	**73.5**	**86.8**	**53.3**	**78.1**	**77.6**

Table 8.7: F1 score of the “Weakly Supervised” method trained with human rules and automatically extracted rules from two different families, namely n-gram rules and high-level rules. Automatically extracted rules with high-level features lead to better performance than human rules and n-gram rules.

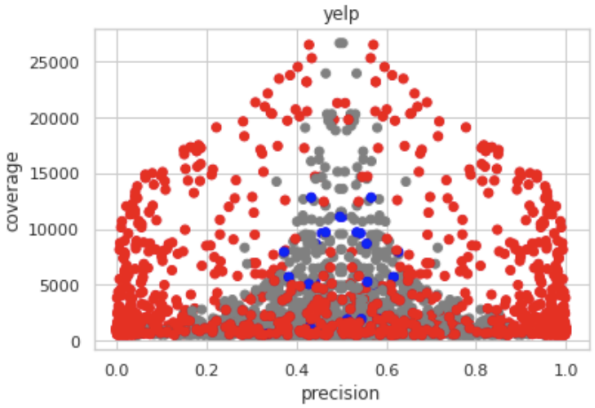
effective than rules based on n-gram predicates: considering our proposed rule family can improve the effectiveness of automatic rule extraction. Also, across most datasets (except TREC), rules with high-level features are more effective than human-provided rules: our rule extraction method can effectively use D_U and D_L to discover high-quality rules. TREC is an exception as it contains the highest number of manually-crafted rules compared to the rest of the datasets. As we will show next, human interaction can lead to further improvements.

8.5.3 Interactive Machine Teaching

Table 8.8 reports classification results of different methods for each dataset. The rightmost column (AVG) reports the average F1 score across datasets. For brevity, we report the best method under each category.



(a) Rules with n-gram predicates. Each rule is a data point (grey).



(b) Rules with high-level predicates: n-grams (grey), named entities (blue), prompt features (red).

Figure 8.3: Precision-coverage scatterplots for rules that were automatically extracted by our method. Rules with high-level predicates can achieve relatively high precision and coverage.

	D_L	D_U	R	T (T_I, T_R)	**YouTube**	**SMS**	**IMDB**	**Yelp**	**TREC**	**AGNews**	**AVG**
Fully Supervised	100%	-	-	-	94.0	95.6	79.6	87.5	90.3	80.7	88.0
Low Supervised	20·K	-	-	-	79.8	82.5	61.6	70.4	55.0	58.8	68.0
Semi Supervised	20·K	yes	-	-	80.7	83.2	63.4	72.0	55.0	60.7	69.2
Weakly Supervised (ASTRA)	20·K	yes	100%	-	**90.0**	**86.8**	**71.2**	**80.2**	**57.0**	**75.9**	**76.8**
Active Instances (random)	20·K	yes	-	100 (100, 0)	83.3	90.1	66.7	77.2	62.7	68.5	74.7
Active Instances (hierarchical)	20·K	yes	-	100 (100, 0)	85.3	89.9	67.6	78.8	61.4	71.4	75.7
INTERVAL	20·K	yes	-	100 (50, 50)	**87.4**	**96.2**	**71.5**	**81.2**	**66.6**	**71.7**	**79.1**

Table 8.8: F1 score reported for various methods on 6 datasets. For each category of baselines, we report the best performing method.

Non-interactive approaches. Across non-interactive approaches, the weakly supervised ASTRA method performs best: using both labeled instances and human-provided rules is more effective than using just labeled instances (in Low Supervised or Semi Supervised), which agrees with our conclusions from Chapter 7. ASTRA outperformed other weakly supervised approaches, including majority voting and Snorkel.

Interactive approaches with queries on instances only. Using the extra interaction budget T in Active Instances (random) improves over Low Supervised: labeling extra instances leads to important performance boosts, as expected. Choosing which instances to label in Active Instances (hierarchical) leads to further performance improvements. The hierarchical sampling method of [Dasgupta et al., 2007] performs better than uncertainty-based

sampling (average F1 = 75.3) and contrastive active learning (average F1 = 74.1). Active Instances (hierarchical) with a budget of $T = 100$ does not outperform Weakly Supervised (ASTRA), which highlights that human-provided rules are worth many examples.

Interactive learning with queries on rules and instances. INTERVAL with a budget of $T = 100$ performs better than the best active learning approach (hierarchical) with the same budget: leveraging feedback on both instances and rules within a limited budget leads is more effective than feedback on instances only. Interestingly, even without using any human-provided rules, INTERVAL outperforms the weakly-supervised ASTRA method. This indicates that automatically-generated rules (analyzed in Section 8.5.2) are effective. While the ASTRA student might capture implicit rules via self-training, many of such rules could be inaccurate, thus highlighting the importance of interaction with a human.

Limitations and future work. Our experimental evaluation of interactive methods has several limitations. First, we evaluated our method by assuming that $T_R = T_I$, which might not be true in practice. For a robust evaluation, it would be important to consider multiple setting with different relative costs of labeling rules and instances. Second, it would be interesting to evaluate interactive approaches by assuming at least a few human-provided rules in R rather than $R = \emptyset$. Third, we evaluated interactive approaches by simulating human feedback using held-out labeled data. In the future, it is important to design user studies and experiment with real humans for the evaluation of our approach and for estimating the costs and benefits of rule feedback.

8.6 New Benchmarks for Machine Teaching

To better advocate and facilitate research on machine teaching, we propose new benchmarks for teaching machines with two different types of supervision, namely labeling rules [Zheng et al., 2022] and task instructions [Wang et al., 2022].

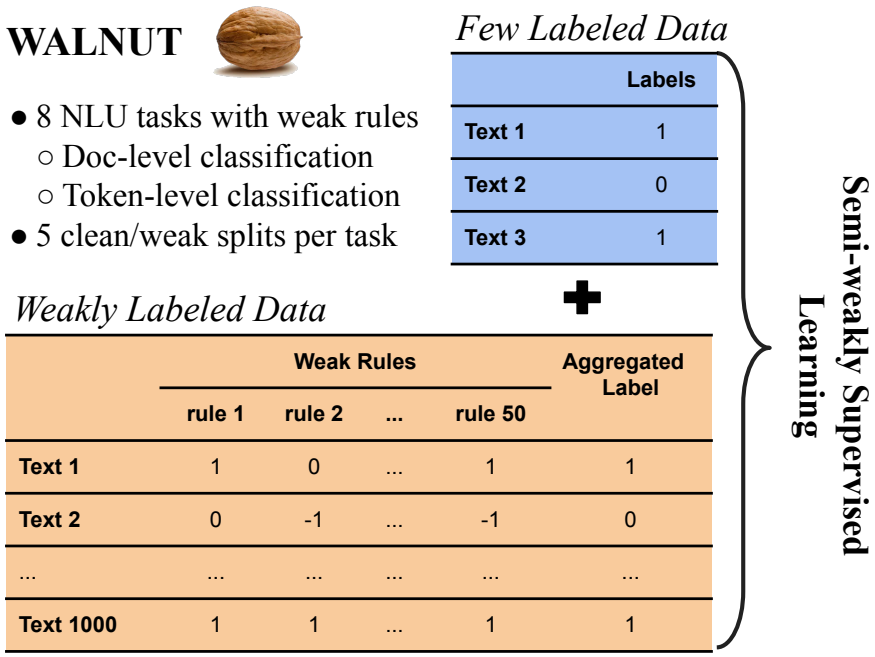


Figure 8.4: WALNUT, a benchmark with 8 NLU tasks with real-world weak labeling rules. Each task in WALNUT includes few labeled data and weakly labeled data for semi- and weakly-supervised learning.

Teaching machines with labeling rules. Throughout our effort for appropriate evaluation of semi- and weakly-supervised learning techniques (both in this chapter and in Chapter 7), we discovered that a unified and systematic evaluation benchmark for Natural Language Understanding (NLU) tasks is rather limited. Existing approaches are evaluated on different data with different metrics and weak supervision sources, making it difficult to understand and compare with each other. To facilitate research on leveraging weak supervision for NLU, in [Zheng et al., 2022] we propose WALNUT (Figure 8.5), a semi- and weakly-supervised learning benchmark of NLU tasks with real-world weak supervision signals. Following the tradition of existing benchmarks [Wang et al., 2018; Wang et al., 2019], WALNUT covers different types of NLU tasks across domains, provides few labeled and many weakly labeled examples for each task (Figure 8.5), and encourages a consistent and robust evaluation of different techniques.

In addition to the proposed benchmark, we demonstrate the benefit of weak supervision for NLU tasks in a collective manner, by evaluating several representative methods and sev-

Figure 8.5: SUP-NATINST covers a 1,616 NLP tasks with the corresponding natural instructions. Bubble size represents the number of tasks of each type in log scale.

eral models of various sizes (e.g., BiLSTM, BERT, RoBERTa), leading to more than 2,000 groups of experiments. Our large-scale analysis demonstrates that weak supervision is valuable for low-resource NLU tasks and that there is large room for performance improvement, thus motivating future research. We expect WALNUT to enable systematic evaluations of semi- and weakly-supervised learning methods and stimulate further research in directions such as more effective learning paradigms leveraging weak supervision.[7]

Teaching machines with task instructions. Our interactive machine teaching method explores prompt-based rules, which use task-specific templates to prompt pre-trained language models. Such templates are manually created with the pre-trained model in mind, for

[7]The benchmark and code for baselines are available at `aka.ms/walnut_benchmark`.

example, by manually compressing the task instructions into short cloze-style descriptions for masked word prediction [Devlin et al., 2019; Schick and Schütze, 2021; Bach et al., 2022]. Given the effectiveness of prompt-based approaches, it would be interesting to investigate (i) how to directly leverage the *original* task instructions, which are not tied to a specific model and are potentially much longer than prompt templates; and (ii) how well can NLP models generalize to a *variety* of unseen tasks when provided with the task instructions. To support future research, in [Wang et al., 2022] we introduce Sup-NatInst, a benchmark of 1,616 diverse NLP tasks and their expert-written instructions. Our collection covers 76 distinct task types, including but not limited to classification, extraction, infilling, sequence tagging, text rewriting, and text composition. This large and diverse collection of tasks enables rigorous benchmarking of cross-task generalization under instructions — training models to follow instructions on a subset of tasks and evaluating them on the remaining unseen ones. We hope our dataset will support research on machine teaching across domains and tasks and will facilitate future progress towards more general-purpose NLP models.[8]

8.7 Conclusions

In this chapter, we presented an interactive machine teaching approach that queries humans for feedback on both instances and automatically generated rules. We summarize the contributions of this chapter as follows: (i) we performed an extensive analysis of existing datasets with human-defined rules and evaluated multiple weak supervision approaches by simulating low-resource rule settings, where just a subset of the human rules are considered in the teacher for training a student; (ii) we proposed a new rule family with high-level rule predicates and present a method that extracts such rules using few labeled and many unlabeled data. In contrast to previous interactive approaches based on n-gram rules, our method extracts rules that can capture higher-level features; (iii) we presented a human-in-the-loop machine teaching framework that queries a human for feedback on both instances

[8]The benchmark is available at `https://github.com/allenai/natural-instructions`.

and rules and effectively uses all resources to train a classifier; (iv) we presented new benchmarks for machine teaching to facilitate future research in machine teaching with different types of interaction.

Our findings show that even though rules are domain specific and have diverse characteristics, there are patterns that are prevalent across datasets. Specifically, a better teacher does not necessarily lead to a better student. We identified that the teacher's precision is more important than coverage for training an accurate student. These findings could potentially inform guidelines for rule creation. Also, we showed that automatic rules based on high-level predicates are more accurate than rules based on n-gram predicates. We additionally showed that by asking queries on both instances and rules, our proposed method can be more effective than active learning methods asking queries on just instances. We hope that our proposed technique as well as our benchmarks for machine teaching will influence research on interactive machine teaching techniques beyond instance labeling.

Chapter 9: Conclusions

In this dissertation, we studied and presented resource-efficient frameworks for teaching machine learning models for NLP tasks across diverse domains and languages. We described our collaborations with experts across domains to integrate weakly-supervised neural networks into operational systems, and we presented efficient machine teaching frameworks that leverage flexible forms of declarative expert knowledge as supervision: coarse labels, large hierarchical taxonomies, seed words, bilingual word translations, and general labeling rules. Next, we summarize our main contributions:

- **Fine-grained classification with coarse-grained labels:** In Chapter 3, we presented a Multiple Instance Learning-based model for fine-grained text classification that requires only review-level labels for training but produces both review- and segment-level labels. We explored non-hierarchical baselines trained at the review level and applied at the segment level by treating each test segment as if it were a short "review." We also developed HSAN, a neural network with a new MIL aggregation function based on the sigmoid attention mechanism, which explicitly allows multiple segments to contribute to the review-level classification decision with different weights. We evaluated our ideas by conducting an experimental evaluation on sentiment classification. We further applied our weakly supervised approach to the important public health application of foodborne illness discovery in online restaurant reviews and demonstrated its deployment for health departments. Our findings show that our non-hierarchical baselines are surprisingly strong and perform comparably or better than MIL-based hierarchical networks with a variety of aggregation functions. By fixing all components except for the MIL aggregation function, we found that the sigmoid attention mecha-

nism in HSAN is the key modeling change needed for MIL-based hierarchical networks to outperform the non-hierarchical baselines for segment-level sentiment classification. Consequently, we believe that HSAN emerges as a promising approach for MIL, especially when the witness rate (i.e., the percentage of positive instances within a bag) is low. Importantly, we showed that HSAN has a higher chance than all previous models to identify unknown foodborne outbreaks, and demonstrated how its fine-grained segment annotations can be used to highlight the segments that were considered important for the computation of the review-level label. By deploying HSAN for inspections in health departments, we provide epidemiologists a new tool to interact with machine learning models, first by using coarse labels to teach segment classifiers, and second to inspect reviews by reading the most important sentences as highlighted by HSAN.

- **Knowledge extraction with hierarchical taxonomies of product categories:** In Chapter 4, we presented a novel method for large-scale attribute value extraction for products from a taxonomy with thousands of product categories. We developed TXtract, a taxonomy-aware deep neural network that extracts attribute values on all product categories in parallel and that captures the hierarchical relations between categories into category embeddings, which in turn are used as context to generate category-specific token embeddings via conditional self-attention. We also developed a multi-task learning framework to jointly extract attribute values and predicting product categories by sharing representations across the two tasks. We performed a large-scale evaluation of TXtract across 4,000 product categories and presented the integration of TXtract with Amazon's AutoKnow. Our findings show that TXtract is both effective and efficient: it leverages the taxonomy into a deep neural network to improve extraction quality and can extract attribute values on all categories in parallel. We also showed that TXtract substantially outperforms state-of-the-art models by up to 10% in F1 and 15% in coverage across all 4,000 product categories. We further demonstrated how TXtract plays an important role in knowledge fact collection for

tens of thousands of product categories at Amazon. Although this work focuses on e-commerce, our approach to leverage taxonomies can be applied to broader domains such as finance, education, and biomedical research.

- **Weakly-supervised text classification with seed words:** In Chapter 5, we presented a weakly supervised approach for leveraging a small number of seed words for segment classification. We showed how to leverage the predictive power of seed words as weak supervision through our teacher model that considers each individual seed word as a (noisy) aspect indicator. We also presented a technique that uses the seed-word based teacher to train an architecture-agnostic student classifier that leverages both seed words and their rich context in unlabeled segments. Then, we showed how iterative co-training can be used to cope with noisy seed words: the teacher effectively estimates the predictive quality of the noisy seed words in an unsupervised manner using the associated predictions by the student. We showed the advantages of our ideas by performing an extensive experimental evaluation on fine-grained aspect detection of restaurant and product reviews. We also applied our teacher-student method for a new application, the analysis of the effects of COVID-19 on restaurant reviews. Our findings show that our student-teacher approach leverages seed words more directly and effectively than previous weakly supervised approaches. The teacher model provides weak supervision to a student model, which we showed that generalizes better than the teacher by also considering non-seed words and by using pre-trained word embeddings. We further showed that iterative co-training leads to a better teacher and, in turn, a better student. Our proposed method consistently outperforms previous weakly supervised methods across al 12 datasets, allowing for seed words from various domains and languages to be leveraged for aspect detection. Our student-teacher approach could be applied for any classification task for which a small set of seed words describe each class. By applying ISWD for the analysis of COVID-19 aspects, we showed revealing trends, such as increased mentions of hygienic practices of restaurants, which could po-

tentially inform policies by public health departments, for example, to cover resource utilization.

- **Cross-lingual transfer of weak supervision with minimal resources:** In Chapter 6, we presented a cross-lingual text classification method, CLTS, that efficiently transfers weak supervision across languages using minimal cross-lingual resources. We presented an efficient method for transferring supervision across languages, which first transfers the most important seed words using the translation budget as a sparsity-inducing regularizer when training a classifier in the source language and second transfers seed words and the classifier's weights across languages, and initializes a teacher classifier in the target language that uses the translated seed words. Also, we effectively applied our weakly-supervised co-training approach from Chapter 5 to this cross-lingual setting for training accurate classifiers in the target language without any labeled target documents. We evaluated our ideas by performing an extensive experimental evaluation on document classification benchmarks across 18 diverse languages. We further applied CLTS for the detection of medical emergency situations in the low-resource Uyghur and Sinhalese languages by just translating a small number of English seed words across languages and presented a cross-lingual transfer method for extending our foodborne illness detection across languages without extra labeling efforts. Our findings show that CLTS effectively transfers supervision from English to all 18 languages for training classifiers using unlabeled-only target documents. Even a simple student outperforms the teacher across all languages by 59.6%, thus proving the effectiveness of our co-training approach for tasks beyond aspect detection, which was our main focus in Chapter 5. CLTS outperforms previous state-of-the-art approaches with more complex models and more expensive resources, highlighting the promise of generating weak supervision in the target language. We further showed that CLTS is robust to noisy translated seed words and therefore can be used even when there is no budget to hire a bilingual speaker by instead using automatically translated seed

words, e.g., via machine translation. Due to the resource-efficiency of our approach, we were able to apply it to low-resource languages and trained accurate classifiers for emergency event detection. Also, by applying our cross-lingual transfer ideas for foodborne illness detection, we trained classifiers that successfully identified reviews discussing food poisoning across several languages, which highlights the potential of our approach for successful, real-world deployment in health departments.

- **Self-training with labeling rules:** In Chapter 7, we presented a weak supervision framework, ASTRA, that efficiently trains classifiers by integrating task-specific unlabeled data, few labeled data, and domain-specific knowledge expressed as rules. We presented an iterative self-training mechanism for training deep neural networks by augmenting the weak supervision signals with instances that are not covered by rules. We also presented a rule attention teacher network (RAN) for combining multiple rules and student model predictions with instance-specific weights conditioned on the corresponding contexts and constructed a semi-supervised learning objective for training RAN. We evaluated our ideas by conducting an experimental evaluation on text classification benchmarks. Our findings show that even simple self-training without human-provided rules sometimes outperforms existing weak supervision approaches that consider rules, highlighting the effectiveness of self-training with pre-trained models, which effectively leverage contextualized representations of instances. By combining the supervision signals from self-training and existing rules, our ASTRA framework improves data coverage by employing self-training with a student model that considers contextualized representations of instances and predicts pseudo-labels for all instances, leading to significant performance improvements over state-of-the-art weak supervision methods and over our self-training baseline. ASTRA is particularly stronger than other approaches at settings with high levels of rule sparsity, highlighting the promise of its effective adoption in emerging tasks with a limited number of human-provided rules.

- **Interactive rule suggestion:** In Chapter 8, we presented a novel interactive learning framework that assists humans by suggesting labeling rules for weak supervision. We performed an extensive analysis of existing datasets with human-defined rules and evaluate multiple weak supervision approaches by simulating low-resource rule settings, where just a subset of the human rules are considered in the teacher for training a student. We proposed a new rule family with high-level rule predicates and present a method that extracts such rules using few labeled and many unlabeled data. We also presented a human-in-the-loop machine teaching framework that queries a human on both instances and rules and effectively uses all resources to train any classifier. Additionally, we presented new benchmarks for machine teaching to facilitate future research in machine teaching with different types of interaction. Our findings show that even though rules are domain specific and have diverse characteristics, there are patterns that are prevalent across datasets. Specifically, a better teacher does not necessarily lead to a better student. We identified that the teacher's precision is more important than coverage for training an accurate student. Also, we showed that automatic rules based on high-level predicates are more accurate than rules based on n-gram predicates. We additionally showed that by asking queries on both instances and rules, our proposed method can be more effective than active learning methods asking queries on just instances.

In summary, in this dissertation we described our collaborations with experts across domains to integrate weakly-supervised neural networks into operational systems, and we presented efficient machine teaching frameworks. Our objective was to support emerging real-world problems without the expensive requirement of large-scale manual data labeling. To address such labeled data bottleneck we presented techniques for assisting humans in teaching machines via more flexible types of interaction. Specifically, we demonstrated the importance of integrating declarative expert knowledge with deep representation learning approaches for effectively teaching machines across domains and languages. We hope that the contributions

of this thesis will serve useful tools, techniques, and benchmarks to the research community and will inspire further research towards more general and efficient frameworks for machine teaching.

Bibliography

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., et al. (2015). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

[Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

[Achille et al., 2019] Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C., Soatto, S., and Perona, P. (2019). Task2vec: Task embedding for meta-learning. *arXiv preprint arXiv:1902.03545*.

[Agrawal et al., 1994] Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Citeseer.

[Alonso and Plank, 2017] Alonso, H. M. and Plank, B. (2017). When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL 2017-15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–10.

[Ammar et al., 2016] Ammar, W., Mulcaire, G., Tsvetkov, Y., Lample, G., Dyer, C., and Smith, N. A. (2016). Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

[Andrews et al., 2003] Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 577–584.

[Angelidis and Lapata, 2018a] Angelidis, S. and Lapata, M. (2018a). Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

[Angelidis and Lapata, 2018b] Angelidis, S. and Lapata, M. (2018b). Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[Arora et al., 2013] Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., and Zhu, M. (2013). A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference on International Conference on Machine Learning*.

[Arora et al., 2016] Arora, S., Liang, Y., and Ma, T. (2016). A simple but tough-to-beat baseline for sentence embeddings.

[Artetxe et al., 2017] Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

[Artetxe et al., 2018] Artetxe, M., Labaka, G., and Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

[Artetxe and Schwenk, 2019] Artetxe, M. and Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

[Ash et al., 2019] Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. (2019). Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*.

[Augenstein et al., 2016] Augenstein, I., Rocktäschel, T., Vlachos, A., and Bontcheva, K. (2016). Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885.

[Awasthi et al., 2020] Awasthi, A., Ghosh, S., Goyal, R., and Sarawagi, S. (2020). Learning from rules generalizing labeled exemplars. In *International Conference on Learning Representations*.

[Ba and Caruana, 2014] Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, number 2010.

[Bach et al., 2022] Bach, S., Sanh, V., Yong, Z. X., Webson, A., Raffel, C., Nayak, N. V., Sharma, A., Kim, T., Bari, M. S., Févry, T., et al. (2022). Promptsource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104.

[Bach et al., 2019] Bach, S. H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., Sen, S., Ratner, A., Hancock, B., Alborzi, H., et al. (2019). Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 International Conference on Management of Data*, pages 362–375.

[Badene et al., 2019] Badene, S., Thompson, K., Lorré, J.-P., and Asher, N. (2019). Data programming for learning discourse structure. In *Association for Computational LInguistics (ACL)*.

[Bahdanau et al., 2015] Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

[Balcan et al., 2005] Balcan, M.-F., Blum, A., and Yang, K. (2005). Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems*.

[Berthelot et al., 2019] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32.

[Beygelzimer et al., 2010] Beygelzimer, A., Hsu, D. J., Langford, J., and Zhang, T. (2010). Agnostic active learning without constraints. In *Advances in neural information processing systems*, pages 199–207.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[Blum and Mitchell, 1998] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Annual Conference on Computational Learning Theory*.

[Boecking et al., 2020] Boecking, B., Neiswanger, W., Xing, E., and Dubrawski, A. (2020). Interactive weak supervision: Learning useful heuristics for data labeling. In *International Conference on Learning Representations*.

[Brantley et al., 2020] Brantley, K., Daumé III, H., and Sharaf, A. (2020). Active imitation learning with noisy guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[Buciluǎ et al., 2006] Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[Cao et al., 2021] Cao, I., Liu, Z., Karamanolakis, G., Hsu, D., and Gravano, L. (2021). Quantifying the effects of COVID-19 on restaurant reviews. In *9th International Workshop on Natural Language Processing for Social Media*.

[Carbonneau et al., 2018] Carbonneau, M.-A., Cheplygina, V., Granger, E., and Gagnon, G. (2018). Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353.

[Caruana, 1997] Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

[Chapelle et al., 2009] Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

[Chen et al., 2011] Chen, M., Weinberger, K. Q., and Blitzer, J. (2011). Co-training for domain adaptation. In *Advances in Neural Information Processing Systems*.

[Chen and Cardie, 2018] Chen, X. and Cardie, C. (2018). Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[Chiu and Nichols, 2016] Chiu, J. P. and Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Clark et al., 2018] Clark, K., Luong, M.-T., Manning, C. D., and Le, Q. (2018). Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[Cohn et al., 1996] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

[Collins and Singer, 1999] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

[Conneau and Lample, 2019] Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*.

[Dai and Le, 2015] Dai, A. M. and Le, Q. V. (2015). Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*.

[Dasgupta et al., 2018] Dasgupta, S., Dey, A., Roberts, N., and Sabato, S. (2018). Learning from discriminative feature feedback. In *Advances in Neural Information Processing Systems*, pages 3955–3963.

[Dasgupta and Hsu, 2008] Dasgupta, S. and Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215.

[Dasgupta et al., 2007] Dasgupta, S., Hsu, D. J., and Monteleoni, C. (2007). A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20:353–360.

[Daumé III, 2007] Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007*, pages 256–263.

[Dawid and Skene, 1979] Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

[Denil et al., 2014] Denil, M., Demiraj, A., and de Freitas, N. (2014). Extraction of salient sentences from labelled documents. *Technical report, University of Oxford*.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Di Renzo et al., 2020] Di Renzo, L., Gualtieri, P., Pivari, F., Soldati, L., Attinà, A., Cinelli, G., Leggeri, C., Caparello, G., Barrea, L., Scerbo, F., et al. (2020). Eating habits and lifestyle changes during covid-19 lockdown: an Italian survey. *Journal of translational medicine*, 18:1–15.

[Diao et al., 2014] Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J., and Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202.

[Dong and de Melo, 2019] Dong, X. and de Melo, G. (2019). A robust self-learning framework for cross-lingual text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

[Dong et al., 2020] Dong, X. L., He, X., Kan, A., Li, X., Liang, Y., Ma, J., Xu, Y. E., Zhang, C., Zhao, T., Blanco Saldana, G., Deshpande, S., Michetti Manduca, A., Ren, J., Pal Singh, S., Xiao, F., Chang, H.-S., Karamanolakis, G., Mao, Y., Wang, Y., Faloutsos, C., McCallum, A., and Han, J. (2020). Autoknow: Self-driving knowledge collection for products of thousands of types. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2724–2734.

[Druck et al., 2008] Druck, G., Mann, G., and McCallum, A. (2008). Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602.

[Duh et al., 2011] Duh, K., Fujino, A., and Nagata, M. (2011). Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*.

[Effland et al., 2018] Effland, T., Lawson, A., Balter, S., Devinney, K., Reddy, V., Waechter, H., Gravano, L., and Hsu, D. (2018). Discovering foodborne illness in online restaurant reviews. *Journal of the American Medical Informatics Association*.

[Efron and Tibshirani, 1994] Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC Press.

[Eisenschlos et al., 2019] Eisenschlos, J., Ruder, S., Czapla, P., Kadras, M., Gugger, S., and Howard, J. (2019). MultiFiT: Efficient multi-lingual language model fine-tuning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

[Faruqui and Dyer, 2014] Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

[Feng and Hirst, 2012] Feng, V. W. and Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

[Frénay and Verleysen, 2013] Frénay, B. and Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

[Fu et al., 2020] Fu, D., Chen, M., Sala, F., Hooper, S., Fatahalian, K., and Ré, C. (2020). Fast and three-rious: Speeding up weak supervision with triplet methods. In *International Conference on Machine Learning*, pages 3280–3291. PMLR.

[Furlanello et al., 2018] Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born-again neural networks. In *International Conference on Machine Learning*.

[Ganchev et al., 2010] Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

[Gao et al., 2021] Gao, T., Fisch, A., and Chen, D. (2021). Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830.

[Gärtner et al., 2002] Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. (2002). Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning*, volume 2, pages 179–186.

[Ghani et al., 2006] Ghani, R., Probst, K., Liu, Y., Krema, M., and Fano, A. (2006). Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48.

[Glavaš et al., 2019] Glavaš, G., Litschko, R., Ruder, S., and Vulić, I. (2019). How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[Gliozzo and Strapparava, 2006] Gliozzo, A. and Strapparava, C. (2006). Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[Goldberg, 2016] Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Gopalakrishnan et al., 2012] Gopalakrishnan, V., Iyengar, S. P., Madaan, A., Rastogi, R., and Sengamedu, S. (2012). Matching product titles using web-based enrichment. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 605–614. ACM.

[Gouws et al., 2015] Gouws, S., Bengio, Y., and Corrado, G. (2015). Bilbowa: Fast bilingual distributed representations without word alignments. In *International Conference on Machine Learning*.

[Gouws and Søgaard, 2015] Gouws, S. and Søgaard, A. (2015). Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Grandvalet and Bengio, 2005] Grandvalet, Y. and Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536.

[Griffiths et al., 2003] Griffiths, T., Jordan, M., Tenenbaum, J., and Blei, D. (2003). Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16.

[Harris et al., 2018] Harris, J. K., Hinyard, L., Beatty, K., Hawkins, J. B., Nsoesie, E. O., Mansour, R., and Brownstein, J. S. (2018). Evaluating the implementation of a Twitter-based foodborne illness reporting tool in the City of St. Louis Department of Health. *International Journal of Environmental Research and Public Health*, 15(5).

[Harris et al., 2014] Harris, J. K., Mansour, R., Choucair, B., Olson, J., Nissen, C., and Bhatt, J. (2014). Health department use of social media to identify foodborne illness-Chicago, Illinois, 2013-2014. *Morbidity and Mortality Weekly Report*, 63(32):681–685.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[He et al., 2019] He, J., Gu, J., Shen, J., and Ranzato, M. (2019). Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

[He et al., 2017] He, R., Lee, W. S., Ng, H. T., and Dahlmeier, D. (2017). An unsupervised neural attention model for aspect extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Hedderich et al., 2021a] Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., and Klakow, D. (2021a). A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568.

[Hedderich et al., 2021b] Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., and Klakow, D. (2021b). A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of NAACL-HLT*.

[Hendrycks et al., 2018] Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. (2018). Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems*, pages 10477–10486.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[Hospedales et al., 2021] Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1.

[Hosseini et al., 2018] Hosseini, P., Ramaki, A. A., Maleki, H., Anvari, M., and Mirroshandel, S. A. (2018). Sentipers: A sentiment analysis corpus for Persian. *arXiv preprint arXiv:1801.07737*.

[Houlsby et al., 2011] Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

[Howard and Ruder, 2018] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Hsieh et al., 2022] Hsieh, C.-Y., Zhang, J., and Ratner, A. (2022). Nemo: Guiding and contextualizing weak supervision for interactive data programming. *arXiv preprint arXiv:2203.01382*.

[Hu and Liu, 2004] Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[Hu et al., 2016] Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. (2016). Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420.

[Huang et al., 2015] Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

[Ilse et al., 2018] Ilse, M., Tomczak, J. M., and Welling, M. (2018). Attention-based deep multiple instance learning. In *Proceedings of the 36th International Conference on Machine Learning*.

[Iyyer et al., 2016] Iyyer, M., Guha, A., Chaturvedi, S., Boyd-Graber, J., and Daumé III, H. (2016). Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Jagarlamudi et al., 2012] Jagarlamudi, J., Daumé III, H., and Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213.

[Joachims et al., 1999] Joachims, T. et al. (1999). Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209.

[Johnson et al., 2017] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

[Karamanolakis et al., 2019a] Karamanolakis, G., Hsu, D., and Gravano, L. (2019a). Leveraging just a few keywords for fine-grained aspect detection through weakly supervised co-training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

[Karamanolakis et al., 2019b] Karamanolakis, G., Hsu, D., and Gravano, L. (2019b). Training neural networks for aspect extraction using descriptive keywords only. In *Proceedings of the Second Learning from Limited Labeled Data Workshop*.

[Karamanolakis et al., 2019c] Karamanolakis, G., Hsu, D., and Gravano, L. (2019c). Weakly supervised attention networks for fine-grained opinion mining and public health. In *Proceedings of the Fifth Workshop on Noisy User-generated Text*.

[Karamanolakis et al., 2020a] Karamanolakis, G., Hsu, D., and Gravano, L. (2020a). Cross-lingual text classification with minimal resources by transferring a sparse teacher. In *Proceedings of the 2020 Findings of Empirical Methods in Natural Language Processing*.

[Karamanolakis et al., 2020b] Karamanolakis, G., Ma, J., and Dong, X. L. (2020b). Txtract: Taxonomy-aware knowledge extraction for thousands of product categories. In *ACL*.

[Karamanolakis et al., 2021] Karamanolakis, G., Mukherjee, S., Zheng, G., and Hassan Awadallah, A. (2021). Self-training with weak supervision. In *NAACL*.

[Karthikeyan et al., 2019] Karthikeyan, K., Wang, Z., Mayhew, S., and Roth, D. (2019). Cross-lingual ability of multilingual BERT: An empirical study. In *International Conference on Learning Representations*.

[Khetan et al., 2018] Khetan, A., Lipton, Z. C., and Anandkumar, A. (2018). Learning from noisy singly-labeled data. In *Proceedings of the International Conference on Learning Representations*.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

[Kim et al., 2017] Kim, Y., Denton, C., Hoang, L., and Rush, A. M. (2017). Structured attention networks. In *Proceedings of the 5th International Conference on Learning Representations*.

[Kim and Rush, 2016] Kim, Y. and Rush, A. M. (2016). Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kirsch et al., 2019] Kirsch, A., Van Amersfoort, J., and Gal, Y. (2019). Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32.

[Klementiev et al., 2012] Klementiev, A., Titov, I., and Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*.

[Ko, 2012] Ko, Y. (2012). A study of term weighting schemes using class information for text classification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1029–1030.

[Koh et al., 2007] Koh, K., Kim, S.-J., and Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine learning research*, 8:1519–1555.

[Kotzias et al., 2015] Kotzias, D., Denil, M., De Freitas, N., and Smyth, P. (2015). From group to individual labels using deep features. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606.

[Kozareva et al., 2016] Kozareva, Z., Li, Q., Zhai, K., and Guo, W. (2016). Recognizing salient entities in shopping queries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 107–111.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

[Lample et al., 2018] Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *International Conference on Learning Representations*.

[Lan et al., 2019] Lan, L., Li, Z., Guan, X., and Wang, P. (2019). Meta reinforcement learning with task embedding and shared policy. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2794–2800. International Joint Conferences on Artificial Intelligence Organization.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[Lee, 2013] Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

[Leskovec et al., 2014] Leskovec, J., Rajaraman, A., and Ullman, J. D. (2014). *Mining of massive datasets*. Cambridge University Press.

[Lewis and Gale, 1994] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer.

[Li et al., 2019] Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.-S., and Schiele, B. (2019). Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, pages 10276–10286.

[Ling and Weld, 2012] Ling, X. and Weld, D. S. (2012). Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[Liu, 2012] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

[Liu et al., 2015] Liu, P., Joty, S., and Meng, H. (2015). Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

[Liu et al., 2020] Liu, Z., Karamanolakis, G., Hsu, D., and Gravano, L. (2020). Detecting foodborne illness complaints in multiple languages using english annotations only. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

[Lopez-Paz et al., 2016] Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. (2016). Unifying distillation and privileged information. In *Proceedings of the International Conference on Learning Representations*.

[Lu et al., 2011] Lu, B., Ott, M., Cardie, C., and Tsou, B. K. (2011). Multi-aspect sentiment analysis with topic models. In *2011 IEEE International Conference on Data Mining Workshops*. IEEE.

[Lund et al., 2017] Lund, J., Cook, C., Seppi, K., and Boyd-Graber, J. (2017). Tandem anchoring: A multiword anchor approach for interactive topic modeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Luong et al., 2015] Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

[Ma et al., 2019] Ma, C., Kang, P., and Liu, X. (2019). Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[Maas et al., 2011] Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[MacQueen, 1967] MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.

[Mann and McCallum, 2010] Mann, G. S. and McCallum, A. (2010). Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(2).

[Margatina et al., 2021] Margatina, K., Vernikos, G., Barrault, L., and Aletras, N. (2021). Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663.

[McAuley et al., 2015] McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[McClosky et al., 2006] McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the 2006 Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics.

[Mekala and Shang, 2020] Mekala, D. and Shang, J. (2020). Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333.

[Melville et al., 2009] Melville, P., Gryc, W., and Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284.

[Meng et al., 2018] Meng, Y., Shen, J., Zhang, C., and Han, J. (2018). Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992.

[Mihalcea et al., 2007] Mihalcea, R., Banea, C., and Wiebe, J. (2007). Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

[Mohammad et al., 2016] Mohammad, S. M., Salameh, M., and Kiritchenko, S. (2016). How translation alters sentiment. *Journal of Artificial Intelligence Research*, 55:95–130.

[Mozetič et al., 2016] Mozetič, I., Grčar, M., and Smailović, J. (2016). Twitter sentiment for 15 European languages. Slovenian language resource repository CLARIN.SI.

[Mukherjee and Awadallah, 2020] Mukherjee, S. and Awadallah, A. H. (2020). Uncertainty-aware self-training for text classification with few labels. *arXiv preprint arXiv:2006.15315*.

[Murphy, 2012] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

[Murphy, 2022] Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

[Naja and Hamadeh, 2020] Naja, F. and Hamadeh, R. (2020). Nutrition amid the covid-19 pandemic: A multi-level framework for action. *European Journal of Clinical Nutrition*, 74(8):1117–1121.

[Natarajan et al., 2013] Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. In *Advances in Neural Information Processing Systems*.

[Nickel and Kiela, 2017] Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347.

[Nigam and Ghani, 2000] Nigam, K. and Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93.

[Nigam et al., 2000] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134.

[Pan and Yang, 2009] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[Pappas and Popescu-Belis, 2014] Pappas, N. and Popescu-Belis, A. (2014). Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language processing*, pages 455–466.

[Pappas and Popescu-Belis, 2017] Pappas, N. and Popescu-Belis, A. (2017). Explicit document modeling through weighted multiple-instance learning. *Journal of Artificial Intelligence Research*, 58:591–626.

[Paszke et al., 2017] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[Patrini et al., 2017] Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Perez et al., 2021] Perez, E., Kiela, D., and Cho, K. (2021). True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

[Peters et al., 2018] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Petrovski and Bizer, 2017] Petrovski, P. and Bizer, C. (2017). Extracting attribute-value pairs from product specifications on the web. In *Proceedings of the International Conference on Web Intelligence*, pages 558–565. ACM.

[Pires et al., 2019] Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[Platanios et al., 2017] Platanios, E., Poon, H., Mitchell, T. M., and Horvitz, E. J. (2017). Estimating accuracy from unlabeled data: A probabilistic logic approach. *Advances in Neural Information Processing Systems*.

[Pontiki et al., 2016] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Mohammad, A.-S., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

[Poria et al., 2016] Poria, S., Cambria, E., and Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

[Poulis and Dasgupta, 2017] Poulis, S. and Dasgupta, S. (2017). Learning with feature feedback: from theory to practice. In *Artificial Intelligence and Statistics*, pages 1104–1113.

[Prettenhofer and Stein, 2010] Prettenhofer, P. and Stein, B. (2010). Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

[Putthividhya and Hu, 2011] Putthividhya, D. P. and Hu, J. (2011). Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567. Association for Computational Linguistics.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *https://blog.openai.com/language-unsupervised*.

[Raghavan and Allan, 2007] Raghavan, H. and Allan, J. (2007). An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79–86.

[Raghavan et al., 2006] Raghavan, H., Madani, O., and Jones, R. (2006). Active learning with feedback on features and instances. *The Journal of Machine Learning Research*, 7:1655–1686.

[Rasooli et al., 2018] Rasooli, M. S., Farra, N., Radeva, A., Yu, T., and McKeown, K. (2018). Cross-lingual sentiment transfer with limited resources. *Machine Translation*, 32(1-2):143–165.

[Ratner et al., 2017] Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

[Ratner et al., 2019] Ratner, A., Hancock, B., Dunnmon, J., Sala, F., Pandey, S., and Ré, C. (2019). Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771.

[Ratner et al., 2016] Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*.

[Rei and Søgaard, 2018] Rei, M. and Søgaard, A. (2018). Zero-shot sequence labeling: transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 293–302.

[Ren et al., 2018a] Ren, H., Stewart, R., Song, J., Kuleshov, V., and Ermon, S. (2018a). Learning with weak supervision from physics and data-driven constraints. *AI Magazine*, 39(1):27–38.

[Ren et al., 2018b] Ren, M., Zeng, W., Yang, B., and Urtasun, R. (2018b). Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*.

[Ren et al., 2020] Ren, W., Li, Y., Su, H., Kartchner, D., Mitchell, C., and Zhang, C. (2020). Denoising multi-source weak supervision for neural text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3739–3754.

[Rezk et al., 2019] Rezk, M., Alemany, L. A., Nio, L., and Zhang, T. (2019). Accurate product attribute extraction on the field. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1862–1873. IEEE.

[Riloff, 1996] Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

[Rogers et al., 2020] Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*.

[Roy and McCallum, 2001] Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. int. conf. on machine learning.

[Ruder, 2019] Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University Of Ireland, Galway.

[Ruder and Plank, 2018] Ruder, S. and Plank, B. (2018). Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Ruder et al., 2019] Ruder, S., Vulić, I., and Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

[Sachan et al., 2018] Sachan, M., Dubey, K. A., Mitchell, T. M., Roth, D., and Xing, E. P. (2018). Learning pipelines with limited data and domain knowledge: A study in parsing physics problems. In *Advances in Neural Information Processing Systems*, pages 140–151.

[Sadilek et al., 2016] Sadilek, A., Kautz, H. A., DiPrete, L., Labus, B., Portman, E., Teitel, J., and Silenzio, V. (2016). Deploying nEmesis: Preventing foodborne illness by data mining social media. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3982–3990.

[Salameh et al., 2015] Salameh, M., Mohammad, S., and Kiritchenko, S. (2015). Sentiment after translation: A case-study on Arabic social media posts. In *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language Technologies*.

[Sang and De Meulder, 2003] Sang, E. T. K. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

[Schick and Schütze, 2021] Schick, T. and Schütze, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

[Schwenk and Li, 2018] Schwenk, H. and Li, X. (2018). A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

[Seeger, 2006] Seeger, M. (2006). A taxonomy for semi-supervised learning methods. Technical report, MIT Press.

[Settles, 2009] Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

[Settles, 2011] Settles, B. (2011). Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478.

[Shen and Lee, 2016] Shen, S.-s. and Lee, H.-y. (2016). Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection. In *Proceedings of INTERSPEECH*, pages 2716–2720.

[Shen et al., 2017] Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., and Anandkumar, A. (2017). Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256.

[Sheth et al., 2017] Sheth, A. P., Ngonga, A., Wang, Y., Chang, E., Slezak, D., Franczyk, B., Alt, R., Tao, X., and Unland, R., editors (2017). *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*. ACM.

[Shi et al., 2010] Shi, L., Mihalcea, R., and Tian, M. (2010). Cross language text classification by model translation and semi-supervised learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Shu et al., 2019] Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*.

[Singh et al., 2019] Singh, J., McCann, B., Socher, R., and Xiong, C. (2019). Bert is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP*.

[Small et al., 2011] Small, K., Wallace, B. C., Brodley, C. E., and Trikalinos, T. A. (2011). The constrained weight space svm: learning with ranked features. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 865–872.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

[Søgaard et al., 2018] Søgaard, A., Ruder, S., and Vulić, I. (2018). On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

[Srivastava and Sutton, 2071] Srivastava, A. and Sutton, C. (2071). Autoencoding variational inference for topic models. In *Proceedings of the International Conference on Learning Representations*.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[Strassel and Tracey, 2016] Strassel, S. and Tracey, J. (2016). Lorelei language packs: Data, tools, and resources for technology development in low resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*.

[Täckström and McDonald, 2011] Täckström, O. and McDonald, R. (2011). Discovering fine-grained sentiment with latent variable structured prediction models. In *European Conference on Information Retrieval*, pages 368–374.

[Täckström et al., 2012] Täckström, O., McDonald, R., and Uszkoreit, J. (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

[Tam et al., 2021] Tam, D., Menon, R. R., Bansal, M., Srivastava, S., and Raffel, C. (2021). Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991.

[Tang et al., 2015] Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

[Vandic et al., 2012] Vandic, D., Van Dam, J.-W., and Frasincar, F. (2012). Faceted product search powered by the semantic web. *Decision Support Systems*, 53(3):425–437.

[Varma and Ré, 2018] Varma, P. and Ré, C. (2018). Snuba: Automating weak supervision to label training data. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 12, page 223. NIH Public Access.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

[Vulić et al., 2019] Vulić, I., Glavaš, G., Reichart, R., and Korhonen, A. (2019). Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

[Vyas and Carpuat, 2019] Vyas, Y. and Carpuat, M. (2019). Weakly supervised cross-lingual semantic relation classification via knowledge distillation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

[Wan, 2008] Wan, X. (2008). Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

[Wan, 2009] Wan, X. (2009). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics.

[Wang et al., 2019] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in neural information processing systems*, pages 3266–3280.

[Wang et al., 2018] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

[Wang, 2019] Wang, W. (2019). Everything old is new again: A multi-view learning approach to learning using privileged information and distillation. *arXiv preprint arXiv:1903.03694*.

[Wang et al., 2022] Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A. S., Naik, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H. G., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Patel, M., Kumar Pal, K., Moradshahi, M., Parmar, M., Purohit, M., Varshney, N., Rohitha Kaza, P., Verma, P., Singh, R. P., Karia, R., Sampat, S. K., Doshi, S., Mishra, S., Reddy, S., Patro, S., Dixit, T., Shen, X., Baral, C., Choi, Y., Smith, N. A., Hajishirzi, H., and Khashabi, D. (2022). Benchmarking generalization via in-context instructions on 1,600+ language tasks. *Submitted to EMNLP 2022*.

[Wang et al., 2021] Wang, Y., Mukherjee, S., Chu, H., Tu, Y., Wu, M., Gao, J., and Awadallah, A. H. (2021). Meta self-training for few-shot neural sequence labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1737–1747.

[Weston et al., 2011] Weston, J., Bengio, S., and Usunier, N. (2011). Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[Wieting et al., 2015] Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2015). Towards universal paraphrastic sentence embeddings. *In Proceedings of the International Conference on Learning Representations (ICLR) (2015)*.

[Wieting and Gimpel, 2017] Wieting, J. and Gimpel, K. (2017). Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2078–2088.

[Wolf et al., 2019] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

[Wu and Dredze, 2019] Wu, S. and Dredze, M. (2019). Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

[Xie et al., 2018] Xie, J., Yang, Z., Neubig, G., Smith, N. A., and Carbonell, J. G. (2018). Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[Xie et al., 2020] Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

[Xu et al., 2019] Xu, H., Wang, W., Mao, X., Jiang, X., and Lan, M. (2019). Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223.

[Xu and Yang, 2017] Xu, R. and Yang, Y. (2017). Cross-lingual distillation for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

[Xu et al., 2013] Xu, W., Hoffmann, R., Zhao, L., and Grishman, R. (2013). Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–670.

[Yadav and Bethard, 2018] Yadav, V. and Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

[Yang et al., 2017] Yang, Z., Salakhutdinov, R., and Cohen, W. W. (2017). Transfer learning for sequence tagging with hierarchical recurrent networks. In *Proceedings of the International Conference on Learning Representations*.

[Yang et al., 2016] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

[Yarowsky, 1995] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Annual Meeting of the Association for Computational Linguistics*.

[Yessenalina et al., 2010] Yessenalina, A., Yue, Y., and Cardie, C. (2010). Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1046–1056.

[Yuan et al., 2020a] Yuan, M., Lin, H.-T., and Boyd-Graber, J. (2020a). Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948.

[Yuan et al., 2020b] Yuan, M., Zhang, M., Van Durme, B., Findlater, L., and Boyd-Graber, J. (2020b). Interactive refinement of cross-lingual word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

[Zhang and Chaudhuri, 2015] Zhang, C. and Chaudhuri, K. (2015). Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pages 703–711.

[Zhang et al., 2022a] Zhang, J., Hsieh, C.-Y., Yu, Y., Zhang, C., and Ratner, A. (2022a). A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*.

[Zhang et al., 2021] Zhang, J., Yu, Y., Li, Y., Wang, Y., Yang, Y., Yang, M., and Ratner, A. (2021). Wrench: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[Zhang and Zong, 2016] Zhang, J. and Zong, C. (2016). Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

[Zhang et al., 2018] Zhang, L., Wang, S., and Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253.

[Zhang et al., 2022b] Zhang, R., Yu, Y., Shetty, P., Song, L., and Zhang, C. (2022b). Prompt-based rule discovery and boosting for interactive weakly-supervised learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 745–758.

[Zhang et al., 2015] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

[Zhang and Yang, 2021] Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.

[Zhao et al., 2021] Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

[Zheng et al., 2021] Zheng, G., Awadallah, A. H., and Dumais, S. (2021). Meta label correction for noisy label learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

[Zheng et al., 2022] Zheng, G., Karamanolakis, G., Shu, K., and Hassan Awadallah, A. (2022). Walnut: A benchmark on semi-weakly supervised learning for natural language understanding. In *NAACL*.

[Zheng et al., 2018] Zheng, G., Mukherjee, S., Dong, X. L., and Li, F. (2018). Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1049–1058. ACM.

[Zhou et al., 2016] Zhou, X., Wan, X., and Xiao, J. (2016). Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

[Zhou and Li, 2005] Zhou, Z.-H. and Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge & Data Engineering*, (11):1529–1541.

[Zhou et al., 2009] Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009). Multi-instance learning by treating instances as non-IID samples. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1249–1256.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.

[Zhu et al., 2015] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

[Zoph et al., 2020] Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. (2020). Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems*, 33.