DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Robotics

Deep End-to-End Learning for Noisy Annotations and Crowdsourcing in Natural Language Processing

Andreas Koch

DEPARTMENT OF INFORMATICS

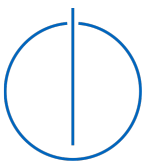
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Robotics

Deep End-to-End Learning for Noisy Annotations and Crowdsourcing in Natural Language Processing

Tiefes End-to-End-Lernen für verrauschte Annotationen und Crowdsourcing in der Verarbeitung von natürlicher Sprache

Author:	Andreas Koch
Supervisor:	Prof. Georg Groh
Advisor:	Gerhard Hagerer
Submission Date:	15.05.2021



I confirm that this master's thesis in robotics is my own work and I have documented all sources and material used.

Munich, 15.05.2021

Acknowledgments

I would like to thank my supervisor Gerhard Hagerer and my team mates David Szabo and Marisa Ripoll for a wonderful time working on this project and for making this thesis possible.

Abstract

Crowdsourcing is an important tool for generating new datasets. As it involves employing mostly non-expert annotators, several types of noise occur in crowdsourced datasets. This thesis compares traditional crowdsourcing approaches to a state-of-the-art end-to-end approach on sentiment analysis. The end-to-end approach is adapted from the computer vision model LTNet by Zeng et al. 2018. It captures annotator bias via annotator specific confusion matrices that reduce noise. LTNet is found to perform well as a crowdsourcing approach, directly providing a classifier for the respective task. Since it considers all available data during the training process, it requires a larger dataset compared to traditional ground truth estimators. In this comparison, the Dawid-Skene and MACE ground truth estimators are applied and the same classifier is trained with their estimates. In order to test LTNet for the task of combining multiple datasets to be treated as one crowdsourced dataset, the end-to-end approach is utilized with dataset specific confusion matrices. The results of this work show no performance increases, although the dataset for this task is not optimal in this regard. For the third aspect of this thesis, LTNet is examined as a ground truth estimator itself. When well trained, it was shown to perform comparable to traditional ground truth estimators. Overall, the end-to-end approach LTNet is found to be useful for most crowdsourcing scenarios.

Contents

1 Introduction

With the rise of the Internet it was possible to develop platforms for generating large crowdsourced datasets. An early example was the Amazon Mechanical Turk platform (Harinarayan et al. 2007). Any form of data could be labeled by employing many people producing one or more labels for each data sample together. This process is called crowdsourcing and is widely used today.
As the crowd workers annotating labels are not experts, they cannot be expected to provide perfect labels. Oftentimes, a sample has conflicting labels with only one being correct. The rest of them are then considered errors. If errors happen consistently according to some pattern it is called noise, owing to the term from signal processing. For classification problems, different kinds of noise are encountered in crowdsourcing: category noise in case of one category consistently having more errors than others, noisy labels produced by bad individual annotators and labeling inconsistencies due to a different understanding of the labeling task. Labeling inconsistencies entail almost only disparate labels when comparing two annotators whereas bad annotators choose labels at random.
To cope with noise and provide one label for each sample is the problem definition for crowdsourcing approaches and more specifically for ground truth estimators. The resulting label is defined as the ground truth that is being estimated. Most ground truth estimators represent each annotator by a probability distribution mapping the estimated true labels to the annotator's labels. This is true for both ground truth estimators evaluated in this work: the Dawid-Skene and the MACE algorithm.
Another aspect, arising with the advent of machine learning, is training a classifier on the input data points with either the ground truth labels or directly with the labels provided by annotators. Training directly on all annotators' labels as well as on the input data can be achieved by training an end-to-end model. Zeng et al. 2018 provide the framework IPA2LT to train this kind of model consisting of a neural network and one matrix for each annotator, mapping the ground truth estimates to each annotator. These matrices represent the bias of an annotator by capturing the different kinds of noise possible in the crowdsourcing problem. In this regard, they should be equivalent to the annotator specific confusion matrix.
Ground truth estimators are older than end-to-end learning approaches (Dawid and Skene 1979). To be comparable, the step of training a classifier is added for ground

truth estimators. Therefore, the performance of each crowdsourcing approach can be examined by measuring multiple machine learning performance metrics for their classifiers. Furthermore, the machine learning tasks are constricted to only sentiment analysis since comparing the crowdsourcing approaches is the focus of this work. For this reason, the first research question of this thesis is formulated as:

1. **How do traditional crowdsourcing approaches compare to a state-of-the-art end-to-end approach on sentiment analysis?**

An important feature of the IPA2LT end-to-end model is the ability to capture labeling inconsistencies with its bias matrices. When comparing two independent datasets for the same problem, they usually have different labeling guidelines. Each dataset's annotators potentially have a distinctive understanding of the categories that give rise to labeling inconsistencies. This is common in facial expression classification (Zeng et al. 2018). A solution to this problem can be combining these datasets into a larger crowdsourced dataset and capture the inconsistencies with a bias matrix for each separate dataset. This is the topic of the second research question:

2. **Can the bias of different crowdsourcing sentiment datasets be modeled successfully by the given approaches?**

With the third research question, a major aspect of crowdsourcing approaches is highlighted: the inferred labels. Since two traditional crowdsourcing approaches are compared with the Dawid-Skene and MACE algorithms in addition to the end-to-end model of IPA2LT, the result will be three different sets of ground truth estimates. Furthermore, the most basic crowdsourcing approach possible in the case of multiple labels per sample, majority voting, is applied. Its ground truth estimates are decided to be the most occurring label of all labels for one sample. In the case of equal occurrences, it is randomly chosen among the highest designated labels. With four sets of ground truth estimates, the overlap amidst these estimates is compared, as well as another metric to measure the agreement. All together this is outlined by:

3. **How similar is the estimated ground truth generated by an end-to-end approach to one produced by traditional ground truth estimators?**

Now in order to provide the relevant background for this work, we move on to the literature section before formerly introducing the previously mentioned crowdsourcing approaches[1]. Later on, the crowdsourced datasets are discussed, as well as how the experiments address the research questions and what their results are.

[1]Code for this thesis available at https://github.com/andiwashere/end-to-end-crowdsourcing

2 Related Work

2.1 Word Embeddings

After many natural language processing (NLP) systems were built with handwritten rules (Weizenbaum 1966) in the 1960s, machine learning became the dominant approach for solving NLP tasks. Since then, researchers tried capturing semantic relations between words with statistical language models (Salton et al. 1975), where one word is represented by a vector. Heightened performance was found when increasing the density of vector representations. At first, this was achieved with simple methods such as reducing the dimension of the word co-occurence matrix with a singular value decomposition (SVD). More refined neural approaches (Bengio et al. 2000) were developed thereupon to either represent words according to their co-occurence with other words or with multiple documents containing the same word (Lavelli et al. 2004). Mikolov et al. 2013 leveraged the increase of computational power to show the effectiveness of the Skip-Gram and the continuous bag-of-words (CBOW) models. Both maximize the prediction of words as a context window scans over the corpus. Pennington et al. 2014 improved on this by switching to a weighted least squares loss function and taking the logarithm of the word co-occurence. This produces a fixed vector representing a word.
The last major advance in word embeddings came from adapting the embedding vector of a word depending on its context. A deep neural network takes embedding vectors corresponding to words as input and outputs vectors of the same dimension. As it is designed to take into account all input vectors, it adapts the output vectors depending on context. The first such design was a Bi-LSTM introduced by Peters et al. 2018 and the state-of-the-art embedding as of now is a contextualized BERT embedding (Devlin et al. 2019). It used an attention mechanism to compare all words in a sentence simultaneously, while randomly masking words. Since this meant abandoning sequential models, the training process was much more efficient.

2.2 Approaches for Noisy Labels

Although noisy labels are a huge problem for training classifiers in machine learning, it is a field not as thoroughly studied as word embeddings. The terminology of noise was adapted from signal processing to refer to misclassified samples, often the result of a random labeling process. The noise occurring in crowdsourcing is split into category noise, noisy labels due to bad annotators and labeling inconsistencies (Hendrycks et al. 2018; Hovy et al. 2013; Zeng et al. 2018). But for now, a short overview of the most prominent solutions to noisy labels is provided.

2.2.1 Robust Loss Functions

The most popular loss functions are not robust to noise and consistently perform disproportionately worse when confronted with noisy data (Ghosh et al. 2017). Zhang and Sabuncu 2018 proposed a generalized cross entropy loss function that is more robust to noise. It is a truncated version of the negative Box-Cox transformation and a mix between mean absolute error and cross entropy. A simple normalization actually leads to noise tolerance as shown by Ma et al. 2020, but it also makes the classifier underfit. A possible solution is to use a combination of an active and a passive loss.

2.2.2 Sample Selection

Another approach for dealing with noise is to only select correct samples according to some metric. MentorNet (Jiang et al. 2018) learns which samples form a good curriculum to train a classifier with. This essentially is a weighting of samples to promote training on samples that are likely correctly labeled. Lee et al. 2019 learn the parameters of a generative classifier from the predictions of an arbitrary classifier while applying the minimum covariance determinant to filter noisy labels. Garg et al. 2021 train a noise model and a weighting function in addition to a regular classifier to detect noisy labels.

2.2.3 Label Correction

Correcting noisy labels is an obvious solution to this problem. Takamatsu et al. 2012 learn a generative model from the number of shared entity pairs among patterns to provide the probabilities of entity pairs belonging to different patterns. A very different approach is training two networks in parallel and switch their training sets in order to make noisy samples retain a high loss and avoid overfitting. It is possible to then generate labels for unlabeled samples from the predictions of both classifiers (J. Li et al. 2020). Essentially, this amounts to using pseudo labels. The end-to-end framework

PENCIL (Yi and Wu 2019) learns label distributions supplanting noisy labels. With a custom loss function any classifier can be made more robust to noisy labels, although performance drops for high noise rates. Pseudo labels in general also experience performance drops for high noise rates as the label-generating network does not retain high accuracy.

2.2.4 Loss Correction

Lastly, Hendrycks et al. 2018 generate a corruption matrix from a trusted data source to capture the noise ratios and then multiply the output from an untrained classifier with it. Their notion of a corruption matrix is closely related to a confusion matrix, with the difference being that probabilities of samples are taken as they are without applying the argmax function. This yields a prediction that is adjusted to category noise considering a higher loss for labels with much noise.
The IPA2LT framework (Zeng et al. 2018) takes a very similar route, insofar bias matrices are trained together with a classifier in an end-to-end fashion. As a bias matrix should represent a confusion matrix, it leads to higher loss in the case of noisy labels. In this work, a slightly adjusted version of IPA2LT is used. It mainly addresses the crowdsourcing setting, while also reducing category noise, labeling inconsistencies and possibly noisy labels as a result of bad annotators.

2.3 Crowdsourcing Approaches

Crowdsourcing is the process of generating labels for data samples by means of multiple people annotating these samples, i.e. giving each sample one or more labels. The result is a crowdsourced dataset containing multiple sets of samples with their corresponding labels from one annotator. This configuration is also referred to as a crowdsourcing setting. To reduce these sets of samples and labels to only one label per sample is the process of finding the ground truth labels. In the following, several approaches to resolve a crowdsourcing setting are introduced.

2.3.1 Majority Voting

Crowdsourcing approaches can be split into two categories: supervised learning from expert annotations and unsupervised learning from non-expert annotations (Whitehill et al. 2009). In this work, only the crowdsourcing setting with non-expert annotations is considered. A simple solution to the crowdsourcing problem is just taking the majority label as the ground truth. Any crowdsourcing algorithm has to perform favorably compared to majority voting. This is especially important for a low number

of annotators per sample. As the number of annotators per sample grows, so does the accuracy of majority voting. Therefore, crowdsourcing algorithms should converge to majority voting with more annotations per sample (Hovy et al. 2013). To evaluate different crowdsourcing algorithms researchers can use a separate test set with known expert annotations in addition to crowdsourced labels. Alternatively, custom metrics such as an error function (Karger et al. 2014) also work.

2.3.2 Generative Approaches

A prominent approach to resolve a crowdsourcing setting is the Dawid-Skene algorithm (Dawid and Skene 1979). It introduced a bayesian method for estimating true labels and models annotators by confusion matrices mapping the current true labels onto the annotators' labels. As the estimated values co-depend on each other, the method is optimized by an expectation maximization (EM) algorithm. This is expanded on in section 3.1 since a version of the Dawid-Skene method was used as a comparison in this work. The Dawid-Skene algorithm is categorized as a generative model as its probability distribution technically allows drawing new samples.
Many consequent bayesian models followed this approach of iteratively optimizing the true labels and a probabilistic representation of the annotators. Smyth et al. 1994 adapted it for an image labeling problem, Raykar and Yu 2012 subtracted the mean of rows from the confusion matrices and applied the L2 norm to get a criterium for spammer detection and Tian and Zhu 2015 regularized the Dawid-Skene likelihood with a maximization of the margin between the true class and competing classes. Of course, there were improvements on the original Dawid-Skene algorithm as well. Sinha et al. 2018 provide a fast, simplified version and Camilleri and Williams 2019 extend the original algorithm to accommodate multiple labeling schemas.
Alternative generative crowdsourcing algorithms include GLAD (Whitehill et al. 2009), also a probabilistic approach that models annotators just by their expertise and considers item difficulty, and MACE, which only learns the annotator behavior when suspecting spamming (Hovy et al. 2013). Moreover, Welinder et al. 2010 infer the ground truth labels with their Bayesian model designed for images and represent annotators by competence, expertise and bias, whereas Yan et al. 2010 model annotator expertise by their reliability and the data they observe to infer ground truth labels.

2.3.3 Discriminative Approaches

The above methods are all generative approaches except for majority voting. However, discriminative models were also proposed to solve this problem. Karger et al. 2011 introduced one that utilized a SVD with a form of belief propagation for inference. The spamming criterium of Tian and Zhu 2015 can also be seen as a discriminative method. Q. Li et al. 2014 optimize an objective function in a weighted voting based method to get ground truth estimates. Last but not least, neural networks are applicable as a purely discriminative solution to this problem (Zeng et al. 2018; Rodrigues and Pereira 2018).

2.3.4 End-To-End Learning

Another important trend in crowdsourcing is end-to-end learning. Instead of resolving multiple labels into one label to train a classifier with, a probabilistic model is trained directly on the observed labels. Contrary to the Dawid-Skene or MACE algorithm, the model also takes the input into account. Most of the times this means representing the ground truth inside of the model and then mapping it to the observed labels. Therefore, end-to-end learning is a combination of inferring the ground truth and modeling the annotator bias for debiasing purposes. In this fashion, Zeng et al. 2018 train a neural network to infer the ground truth. Annotators and their bias are represented by confusion matrices similar to the Dawid-Skene algorithm. Both network and matrices are optimized jointly on the observed labels with back-propagation. This thesis examines a version of this approach, which is formally introduced in section 3.2. Other end-to-end learning algorithms include Raykar, Yu, et al. 2009, a generative classifier to infer the ground truth while optimizing the annotator sensitivity, specificity and parameters for a logistic regression model using an EM algorithm, as well as Khetan et al. 2017 and Rodrigues and Pereira 2018, both of which represent annotator bias by confusion matrix estimates and also optimize their respective generative classifier with an EM algorithm. Only Raykar, Yu, et al. 2009 test their approach on text data, namely the emotion dataset by Snow et al. 2008. Otherwise, these methods have only been applied to image data.

2.4 Crowdsourced Datasets

2.4.1 Dataset types

Usually when referring to a crowdsourced dataset, one means a *multi-labeled* crowdsourced dataset. This class of datasets requires more than one person providing a label to each sample. The previously mentioned methods all infer the ground truth labels from multiple labels. Snow et al. 2008 provide several small multi-labeled datasets for natural language tasks, one of which is used for experiments. More details can be found in section 4.3. Further examples of multi-labeled datasets are the Wikipedia toxicity dataset (Wulczyn et al. 2017), the GoEmotion dataset by Google (Demszky et al. 2020) and the SEWA database (Kossaifi et al. 2021).
As the creation of multi-labeled datasets can be quite expensive, many datasets are *singly-labeled* only, i.e. each sample has one label. This means that each worker labels his or her own subset of the overall dataset. Although not ideal, this form of crowdsourcing is quite common (Thelwall 2018; Danner and Menapace 2020). Therefore, another criterium to crowdsourcing algorithms is applicability to singly-labeled datasets. Khetan et al. 2017 specifically provide a crowdsourcing algorithm for this setting. As they argue, the Dawid-Skene algorithm is ineffective for singly-labeled datasets. With only one label per example, the EM algorithm estimates that all the workers are perfect since ground truth labels are initialized by a majority vote. In contrast to this, end-to-end learning approaches are optimized directly on the noisy labels of each annotator. If these approaches include a mechanism to reduce noise, they are well suited for singly-labeled datasets.

2.4.2 Annotator Types

As the annotation can take different forms depending on the dataset and context, a short breakdown of different types of annotation schemes relevant for the datasets in this work is provided.

Reviewer Annotators

In datasets generated from product reviews, it is common that the reviewer provides a rating in addition to his review. These reviewers will be referred to as *annotator reviewers*. Taking the reviewers' ratings as labels is a cost-effective way of creating singly-labeled crowdsourced datasets. This dataset generation process is easy but yields lower quality crowdsourced datasets due to the high number of different annotators. A solution is to group annotators together and then apply crowdsourcing algorithms for different groups (Thelwall 2018). As machine learning approaches thrive with large

datasets, the performance of different crowdsourcing algorithms for this type of dataset is examined in this work.

External Annotators

With the exception of product reviews, crowdsourced datasets are usually externally annotated. This means a crowd worker is tasked with providing labels for already existing samples. Thereby, workers are divided into *expert annotators* and *non-expert annotators*. Expert annotators are assumed to have domain-specific knowledge or other qualifications and thus produce higher quality labels. The aspect-based sentiment analysis datasets from SemEval 2014, 2015 and 2016 (Kirange and Deshmukh 2014; Pontiki, Galanis, Papageorgiou, Manandhar, et al. 2015; Pontiki, Galanis, Papageorgiou, Androutsopoulos, et al. 2016) are for example all annotated by expert annotators, mainly linguists. A non-expert annotator does not have domain-specific knowledge. This is the default assumption when creating a crowdsourced dataset from a crowdsourcing platform. Snow et al. 2008 demonstrated the effectiveness of using this approach for crowdsourced dataset generation and Danner and Menapace 2020 provide a singly-labeled crowdsourced dataset for aspect-based sentiment analysis labeled by non-expert annotators.

2.4.3 Data Curation

In order to improve the overall data quality of a crowdsourced dataset or just to gain insight, researchers can apply different approaches of data curation. These usually do not involve inferring ground truth labels.
One of these methods is the detection of spammers. Besides representing sentences by feature vectors from all observed labels, Aroyo and Welty 2013 detect spammers that often disagree with the majority.
Similar to spammer detection, modeling the annotator performance or task difficulty are popular options for data curation, even when not inferring ground truth labels. Carpenter 2008 uses the parameters of a multi-level beta distribution to either calculate the annotator specificity and sensitivity as performance indicators or the task difficulty of samples.
Another approach in order to gain insight into a crowdsourced dataset, is to capture sources of bias, instead of modeling the labeling bias of annotators with a confusion matrix for example. Wauthier and Jordan 2011 describe annotators as influenced by shared random effects and each is expressed with a sum of weighted vectors drawn from their own zero-mean Gaussian distribution. Zhuang et al. 2015 propose a method for capturing the perceived order of samples when they are presented to annotators in

batches.
Lastly, clustering annotators together also provides useful information. Peldszus and Stede 2013 presented methods for clustering and studying the agreement in a larger group of annotators. Several of the previously introduced approaches utilize their representation of annotators to form clusters (Welinder et al. 2010; Raykar and Yu 2012).

Now that all relevant topics for this thesis were established, let us return to the research questions. The main goal is to compare traditional crowdsourcing approaches to an end-to-end approach. As the most promising development in crowdsourcing, an end-to-end approach takes the input data into account and its model is trained directly on all observed labels. Although, since it is trained on the observed labels directly, a mechanism to reduce noise and the influence of bad annotators is needed. The end-to-end framework IPA2LT presented by Zeng et al. 2018 provides a solution for this problem with the addition of annotator specific bias matrices on top of an arbitrary discriminative classifier. A version of this framework is introduced in the following section. As it shares a lot of similarities with the more traditional Dawid-Skene approach, the same discriminative classifier is trained on the Dawid-Skene inferred labels for comparison. MACE is the last approach that is applied. It models spamming behavior to infer the ground truth labels and thus represents a different direction of research. As with Dawid-Skene, the discriminative classifier is trained on the inferred labels. A simple attention based neural network is chosen as the discriminative classifier because this allows a focus on the evaluation of the three different crowdsourcing approaches. Zeng et al. 2018 originally applied their framework on multiple datasets by treating them as a singly-labeled crowdsourced dataset. This behavior is investigated in the second research question.
Lastly for the third research question, the inferred labels of all crowdsourcing approaches are compared. Thereby, the quality of the neural network's predictions is evaluated when considering them as the inferred labels of the IPA2LT framework.

3 Methodology

In this section the crowdsourcing approaches necessary for answering the research questions are introduced. Firstly, the Fast Dawid-Skene (FDS) algorithm is presented. As the main end-to-end framework Inconsistent Pseudo Annotations to Latent Truth (IPA2LT) is modeled after the original Dawid-Skene (DS) algorithm. The FDS approach is an alternation of the DS algorithm and also infers ground truth labels from all annotators' labels. Since it is schematically almost identical to the original DS algorithm, the term Dawid-Skene algorithm or approach is used to just refer to the implementation of the FDS algorithm for this thesis. Consecutively, the probabilistic model of IPA2LT is explained with all of its benefits, being an end-to-end model capable of capturing category noise and annotator bias. Lastly, the third crowdsourcing approach, Multi-Annotator Competence Estimation (MACE), is presented. It also infers ground truth estimates based on all annotators' labels. Although unlike Dawid-Skene, it filters possible spamming behavior.

3.1 Fast Dawid-Skene Algorithm

To figure out which label is the ground truth label for each sample given all the observed crowdsourced labels, many generative crowdsourcing approaches employ an expectation maximization algorithm. One of the first methods to do so was the Dawid-Skene algorithm. The Fast Dawid-Skene algorithm by Sinha et al. 2018 provides significant computational gains over the regular Dawid-Skene algorithm while still preserving convergence. This is achieved by introducing a simplification for the representation of estimated labels. However, the difference is minor. In the following, both methods are introduced and the simplification is explained. As every other EM algorithm, the FDS algorithm can be divided into an expectation step and a maximization step.
Given the data $\mathcal{X} = \{x_1, \ldots, x_N\}$ with L different labels $\{1, ..., L\}$ and C annotators $\{1, ..., C\}$, y_n^c denotes the label assigned to sample x_n by coder c. Let $i \in \{1, ..., L\}$ be the best guess for the correct choice y_n with $P\left(y_n = i | y_n^1, ..., y_n^C\right)$ at a maximum. Applying Bayes' theorem and the independence assumption among the annotators' assigned

labels, Sinha et al. 2018 derive

$$\begin{aligned} & P(y_n = i | y_n^1, ..., y_n^C) \\ &= \frac{P(y_n^1, ..., y_n^C | y_n = i) P(y_n = i)}{\sum_{j=1}^{L} P(y_n^1, ..., y_n^C | y_n = j) P(y_n = j)} \\ &= \frac{\left(\prod_{c=1}^{C} P(y_n^c | y_n = i) \right) P(y_n = i)}{\sum_{j=1}^{L} \left(\prod_{c=1}^{C} P(y_n^c | y_n = j) \right) P(y_n = j)} \,. \end{aligned} \tag{3.1}$$

With a formulation for the probability distribution required to make a guess, I_{ni} will be introduced as an indicator function mapping sample x_n to its proposed label i:

$$I_{ni} = \begin{cases} 1 & i = \underset{j \in \{1,...,L\}}{\operatorname{argmax}} P(y_n = j | y_n^1, ..., y_n^C) \\ 0 & \text{otherwise} \end{cases} \,. \tag{3.2}$$

Instead of applying the argmax function, the original DS algorithm does not necessarily treat the indicator values as binary, allowing factors with only decimal values for I_{ni} to contribute to the overall likelihood (Sinha et al. 2018).
Utilizing the newfound labels, expressions for $P(y_n^c | y_n = i)$ and $P(y_n = i)$ are formulated by counting samples. With the sets,

$$\begin{aligned} \mathcal{X}_c^{(i)} &= \{x_n | y_n = i \ \wedge c \text{ has annotated sample } x_n\} \\ \mathcal{X}_{cj}^{(i)} &= \{x_n | y_n = i \ \wedge c \text{ has annotated sample } x_n \text{ with label } j\} \\ \mathcal{X}_i &= \{x_n | I_{ni} = 1\} \,, \end{aligned}$$

and the cardinality of a set $|\cdot|$, the likelihood of coder c answering with label j becomes

$$P(j | y_n = i) = \frac{\left| \mathcal{X}_{cj}^{(i)} \right|}{\left| \mathcal{X}_c^{(i)} \right|} \,. \tag{3.3}$$

Furthermore, the prior of a given label i can be described as

$$P(y_n = i) = \frac{|\mathcal{X}_i|}{|\mathcal{X}|} \,. \tag{3.4}$$

Overall, these probabilities are computed iteratively with the EM algorithm. The E-step estimates new labels using equations (3.1) and (3.2), while the M-step calculates new

values for $P(j|y_n = i)$ and $P(y_n = i)$ with equations (3.3) and (3.4). In the first step, labels are assigned by majority voting. The complete FDS algorithm is presented in algorithm 3.1 stated below. After completion, the inferred labels are utilized to train a classifier with. This will be expanded on further in section 3.2.2.

Algorithm 3.1: Fast Dawid-Skene Algorithm as in Sinha et al. 2018

Input : Crowdsourced labels of N questions assigned by C coders from L possible labels

Output: Proposed true labels - I_{ni}

Estimate I_{ni} using majority voting;

repeat

  M-step: Obtain the parameters, $P(j|y_n = i)$ and $P(y_n = i)$ using the equations (3.3) and (3.4);

  E-step: Estimate Is with the parameters, $P(j|y_n = i)$ and $P(y_n = i)$ and the equations (3.1) and (3.2).

until *convergence*;

3.2 IPA2LT Framework

In the beginning of this section it was mentioned that IPA2LT is modeled after the Dawid-Skene algorithm. There are many similarities, although it is still fundamentally different. The Dawid-Skene algorithm infers ground truth labels based on all annotators' labels. IPA2LT is a framework that trains the underlying end-to-end model, Latent Truth Network (LTNet). Besides the actual network parameters, it only takes the input data into account for estimating ground truth labels. The overall structure of LTNet is depicted in figure 3.1. It consists of a neural network to produce ground truth estimates and one matrix per annotator that maps the estimates to annotator specific predictions. LTNet is then optimized via backpropagation. The matrices on top of the neural networks are supposed to represent the annotator bias. This is true if they are akin to the respective confusion matrices. Furthermore, LTNet's bias matrices allow noisy labels to have a lower influence on training the neural network as small bias matrix values mean a higher loss as well as smaller gradient updates. In this context, noisy labels can be a result of either category noise, bad annotators or labeling inconsistencies. These topics are expanded on in the following sections.

3.2.1 LTNet

Using the same notation as before, labeling inconsistency can be expressed by

$$P(y_n^c|x_n) \neq P(y_n^d|x_n),\ \forall x_n \in \mathcal{X},\ c \neq d \tag{3.5}$$

meaning coder c has a different tendency towards labeling than coder d. For a given ground truth of i, coder c chooses j with the probability

$$\tau_{ij}^c = P(y_n^c = j|y_n = i)\ . \tag{3.6}$$

The IPA2LT framework trains an end-to-end classifier LTNet with parameters Θ and provides its best estimation for the ground truth as a latent parameter, called latent truth y_n. By processing the corresponding sample x_n and yielding an output vector of dimension L, the probability of the latent truth becomes $P(y_n = i|x_n;\Theta)$ for category i. Therefore, sample x_n is annotated as label j by coder c according to

$$P(y_n^c = j|x_n;\Theta) = \sum_{i=1}^{L} P(y_n^c = j|y_n = i)P(y_n = i|x_n;\Theta)\ . \tag{3.7}$$

Contrary to the Dawid-Skene approach, LTNet has the objective goal of maximizing the log-likelihood of all annotations

$$\max_{\Theta,\boldsymbol{T^1},...,\boldsymbol{T^C}} \log(P(\boldsymbol{y^1},...,\boldsymbol{y^C}|\mathcal{X};\Theta))\ , \tag{3.8}$$

with $\boldsymbol{T^c} = [\tau_{ij}^c]_{L\times L}$ as the transition matrix for coder c based on (3.6) and $\boldsymbol{y^c} = [y_1^c,...,y_N^c]^T$ as all annotations by coder c. Applying the independence assumption and (3.7), the loglikelihood becomes

$$\begin{aligned}\log(P(\boldsymbol{y^1},...,\boldsymbol{y^C}|\mathcal{X};\Theta)) &= \log\left(\prod_{n=1}^{N}\prod_{c=1}^{C} P(y_n^c|x_n;\Theta)\right)\\ &= \sum_{n=1}^{N}\sum_{c=1}^{C}\sum_{j=1}^{L} \mathbf{1}(y_n^c = j)\log\left(\tau_{ij}^c P(y_n = i|x_n;\Theta)\right)\ ,\end{aligned} \tag{3.9}$$

with $\mathbf{1}(\cdot)$ as the indicating function. It is 1 only if the condition holds.
As stated before, LTNet is implemented as a neural network with one transition matrix per coder on top of a basic neural network, see Figure 3.1. Each row of a transition matrix needs to be normalized to guarantee a probability distribution over all categories, $\sum_{j=1}^{L} P(y_n^c = j|y_n = i) = \sum_{j=1}^{L} \tau_{ij}^c = 1$ for coder c. With this constraint, the negative log-likelihood (NLL) loss function, which will be minimized during the training procedure,

is formulated as following:

$$\min_{\Theta, \boldsymbol{T^1}, ..., \boldsymbol{T^C}} - \sum_{n=1}^{N} \sum_{c=1}^{C} \sum_{k=1}^{L} \mathbf{1}(y_n^c = j) \log\left(\hat{p}_n^c(k)\right)) \tag{3.10}$$
$$s.t. \sum_{j=1}^{L} \tau_{ij}^c = 1, \ \forall i \in \{1, ..., L\} \ .$$

With the latent truth vector $\boldsymbol{p} = [P(y_n = 1|x_n, \Theta), ..., P(y_n = L|x_n, \Theta)]^T$, the predicted probability of coder c's annotation is defined by Zeng et al. 2018 as

$$\hat{\boldsymbol{p}}_{\boldsymbol{n}}^{\boldsymbol{c}} = \boldsymbol{p}^T \boldsymbol{T}^c = \left[\sum_{i=1}^{L} P(y_n = i|x_n, \Theta) \tau_{i1}^c, ..., \sum_{i=1}^{L} P(y_n = i|x_n, \Theta) \tau_{iL}^c \right] \ .$$

Hagerer et al. 2021 explain an inconsistency with this approach when trying to capture the bias of an annotator. The logarithm in (3.10) prevents backpropagation of meaningful information to the bias parameters. Instead, its derivative weighs inaccurate results more severely such that parameters of the bias matrix do not reflect the number of true positives etc. accurately. The bias matrix however should be similar to the confusion matrix. At least, this is a more precise way of capturing the tendencies of annotators. As the interest of this work lies in the classification performance and Hagerer et al. 2021 also show it is comparable or slightly better without the log function, the scope of this work is constricted to only use the negative likelihood (NL) loss function,

$$\min_{\Theta, \boldsymbol{T^1}, ..., \boldsymbol{T^C}} - \sum_{n=1}^{N} \sum_{c=1}^{C} \sum_{k=1}^{L} \mathbf{1}(y_n^c = j) \hat{p}_n^c(k)) \tag{3.11}$$
$$s.t. \sum_{j=1}^{L} \tau_{ij}^c = 1, \ \forall i \in \{1, ..., L\} \ .$$

In the implementation of LTNet, randomized batch-wise backpropagation is applied for the optimization. Thereby, all samples are masked except for the samples by one annotator and execute the according gradient update step for the annotator's bias matrix and the underlying basic classifier of section 3.2.2. This step is executed for every annotator before moving on to the next batch. Furthermore, stochastic gradient descent (SGD) is used as an optimizer because it does not provide a skewed weighting of gradients compared to other optimizers, e.g. the Adam optimizer. This is the same optimizer as was used in Zeng et al. 2018.

3.2.2 Basic Classifier

As demonstrated in figure 3.1, the basic classifier is a simple attention model. Since the focus of this work lies in comparing different crowdsourcing approaches, multiple

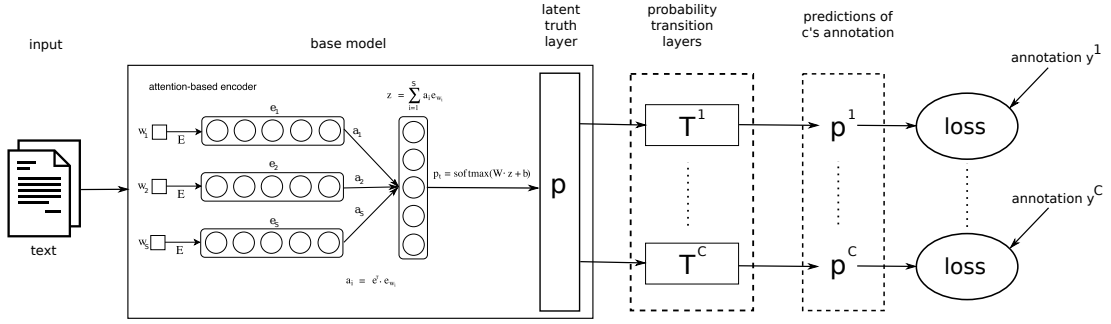


Figure 3.1: Complete architecture of LTNet from Hagerer et al. 2021. Each word is transformed to its embedded representation and a simple attention model (see section 3.2.2) is applied to produce the latent truth vector. The latent truth is then multiplied with the bias matrices to produce predictions for each annotator.

classifiers and complex features were avoided. For an input sentence x_n consisting of S words, several preprocessing steps are applied as explained in section 4. In these steps, the words are converted into word vectors w_s using an embedding of size d. Accordingly, the word vectors are $w_s \in \mathbb{R}^d, \forall s \in \{1, \ldots, S\}$. Each of them is multiplied with a trainable vector $e \in \mathbb{R}^d$ to give the weight a_s. Then the sum of all word vectors weighted with their respective a_s is computed as

$$z = \sum_{s=1}^{S} a_s w_s = \sum_{s=1}^{S} (e^T w_s) w_s \ , \tag{3.12}$$

resulting in a sentence representation $z \in \mathbb{R}^d$. Finally, a linear layer is applied to z including a non-linearity and thus provide the capability for the model to act as a general function approximator yielding the latent truth vector $\boldsymbol{p}$,

$$\boldsymbol{p} = \text{softmax}(W \cdot z + b) \ . \tag{3.13}$$

3.2.3 Noisy Labels

As was mentioned, LTNet employs the noise reduction technique of adding confusion matrices on top of its neural network. In case the network learned to always predict the label of a specific annotator, this annotator's confusion matrix would be an identity matrix. This can be interpreted as the network treating the annotator as a perfect annotator. Adding confusion matrices allows LTNet to capture three different types of noise.

Category Noise

Hagerer et al. 2021 show bias matrices adequately capture noise for the different categories. Considering only one category, in case of a large portion of samples being predicted differently to the corresponding labels of an annotator, its bias matrix parameters would show high noise. This means that the non-diagonal elements are relatively high.
Now with the ability of capturing noise for different classes, high noise for one class increases the loss of only the samples with this label. With a lower weight of the bias matrix, the prediction is less certain and naturally the loss is higher.
Besides increasing the loss, a lower diagonal weight in the bias matrix also discounts the gradient update for the corresponding prediction. Essentially, this leads to the network being more ambiguous over time when faced with this sample. Therefore, one can argue that the sample has less influence in the decision making of the network. At the same time, this could improve the ability of LTNet to learn the underlying task.
Capturing noise of different categories only works if one or more categories have significantly more noise than the others. If all categories have similar confusion matrix parameters, the losses are not increased, gradient updates are discounted in the same way and LTNet has no effect.

Dealing with Bad Annotators

Another source of noise in a crowdsourced dataset is a bad annotator that annotates close to random. The annotator specific bias matrix would pick up on this and exhibit lower diagonal values compared to all other bias matrices. Thereby, it might have various levels of noise for different categories although the main discrepancy would be between bias matrices and not category noise. Labels of bad annotators are discounted the same as samples of a noisy category both with an increase of loss and a lower gradient update.

Inconsistent Labeling

As introduced before, inconsistent labeling refers to one annotator constantly giving samples disparate labels than another annotator. This is the result of understanding the labeling task differently. Inconsistent labeling is a common occurrence in emotion recognition as every person has a distinct understanding of emotion. The same as before, inconsistent labeling can be accounted for by discounting the influence of inconsistent annotators.
Another common occurrence for inconsistent labels are separate datasets with the exact same task. Usually when creating and labeling a dataset, the labelers have a

common understanding of the categories. The meaning of certain categories might be different for other datasets though. Therefore, the result will be inconsistent labels when comparing them. LTNet is applicable to this problem when representing one dataset with one bias matrix, treating all datasets as one singly-labeled crowdsourced dataset. Similarly, the influence of inconsistent labels is discounted with an increase of loss and a lower gradient update. This feature of LTNet is investigated in the third research question.

3.2.4 Embeddings

For this work, the GloVe embedding by Pennington et al. 2014 with size $d = 50$ will be utilized. The embedding is obtained by optimizing the word vectors according to an error function. It includes taking the logarithm of the co-occurence probability of two words and comparing it to the scalar product of the respective word vectors. Finally, each word pair is weighted according to a non-linear function of the co-occurence.
As this thesis contains experiments with several domain-specific, small datasets, the GloVe embedding will be fine-tuned for each. Essentially, the training process is repeated with a combined vocabulary of all words originally used plus the words specific to one dataset and its data is taken to build the co-occurence matrix. For this purpose, the Mittens model by Dingwall and Potts 2018 is used to train domain-specific GloVe embeddings.

3.3 MACE

The results from LTNet and Dawid-Skene are compared to another probabilistic approach for crowdsourcing, the MACE model by Hovy et al. 2013. It proposes that the labels by each annotator depend on latent true labels and on whether the annotator is spamming at random. Thereby, the assumption is made that an annotator only answers with the true label if not spamming. In contrast to IPA2LT, this process solely derives its best guess for the true labels. It does not provide a classifier in and of itself. Accordingly, the same classifier as in section 3.2.2 is trained on the final ground truth labels. For each sample x_n and coder c, a latent true label y_n is drawn from a uniform distribution and the binary variable S_n^c indicating spamming from a Bernoulli distribution with parameter $1 - \theta^c$. The coder is not spamming if $S_n^c = 0$ and in that case, it is assumed to copy the latent true label as its own label $y_n^c = y_n$. The labels of a spamming coder are modeled with a multinomial distribution depending on parameter ξ^c. Altogether, this results in the marginal data likelihood of the observed labels of

$$P(\boldsymbol{y}^1, \ldots, \boldsymbol{y}^C; \theta, \xi) = \sum_{p,S} \left[\prod_{n=1}^{N} P(y_n) \cdot \prod_{c=1}^{C} P(S_n^c; \theta^c) \cdot P(y_n^c | S_n^c, y_n; \xi^c) \right] , \tag{3.14}$$

where $\boldsymbol{y}^c$ indicates all labels by coder c, θ and ξ are the parameters of MACE, p is the vector of all true labels $p = [y_1, \ldots, y_N]^T$ and S the matrix of the spamming parameters $(S)_{nc} = S_n^c$. Similar to the Dawid-Skene algorithm, this probability is maximized with an EM algorithm. In a two-step process, Hovy et al. 2013 alternate between deriving the next model parameters and calculating the marginal data likelihood. The process can be improved by adding priors of annotator behavior to each of the model parameters. This essentially leads to non-linear probabilities in (3.14) and thus allows a more accurate depiction of annotator proficiency. The best model provided by Hovy et al. 2013 will be used in their Java-based script to generate ground truth estimates. However, only the classifier performance achieved with these labels is compared to the performance of the other two methods.

4 Datasets

Before it is possible to examine how traditional crowdsourcing approaches compare to an end-to-end approach, several crowdsourced datasets are needed. For the first research question that focuses on crowdsourcing, a multi-labeled crowdsourced dataset with many annotators is desirable. Alternatively, a singly-labeled crowdsourced dataset with many annotators is also acceptable.
As the tendencies of an annotator to give a certain answer are modeled, annotators can be grouped together to form the bias of a specific group of people. This becomes useful only if specific details about the individuals are available, such as race or gender. Then the underlying machine learning model could be assessed in terms of bias of these groups of people. As this might be a good application for LTNet, its performance is tested on datasets with background information on annotators. However, it is important to mention that the actual bias of certain groups of people, potentially relevant for social sciences, is not being studied in this work.
Another useful application of LTNet is the ability to combine multiple datasets with the same task, even though their labeling scheme might be inconsistent. This feature will be explored in the second research question, in which the performance of all mentioned crowdsourcing approaches is compared. The datasets to be combined do not have to be generated by crowdsourcing themselves. Although put together, they can be seen as one greater crowdsourced dataset with multiple disjunct singly-labeled parts.
As far as datasets with extra information about their annotators are concerned, two of the implemented datasets contain the gender of their annotators. However, both of them are singly-labeled only. For this reason, a multi-labeled crowdsourced dataset with multiple annotations per sample is also included.

4.1 TripAdvisor Dataset

The first dataset was generated from the TripAdvisor.com website by Thelwall 2018. It consists of 720,897 hotel reviews and 571,569 restaurant reviews about hotels and restaurants in the UK, written by UK residents from February to March 12th, 2017. Each review is rated with one of five possible ratings: 10, 20, 30, 40 or 50 with 50 as the best rating. Furthermore, both the hotel and restaurant subset of the dataset have the same amount of reviews for each rating and only one review per reviewer is allowed. It is important to note that the person who wrote the review also decided the rating, thus making this dataset reviewer annotated.
For this work, the 5 ratings are mapped to either negative or positive sentiment ignoring neutral ratings in order to perform binary sentiment analysis and simplify the task at hand. All reviews are moreover grouped together according to the annotator's gender, meaning the annotator will be either male or female. In total, these restrictions limit the actual amount of samples to 11900 for hotel reviews and 34008 restaurant reviews. Their sample lengths are depicted in figure 4.1. Considering these samples, they are split in a training-validation-test split of 70-20-10.
For the research question of combining multiple datasets with the same task, the different portions of this dataset are utilized, i.e. LTNet compares the bias of restaurant reviews with the bias of hotel reviews. If there are major differences, an improvement in performance could be expected when compensating for possible noise. In this context, the gender information is not considered.
Thelwall 2018 trains a regression model on the original 5 ratings with uni-, bi- and trigrams as features and only compares the Pearson correlation for predictions and labels among the subsets for each gender and both genders. This is very different from the methods in this work and thus will not be considered for comparison.

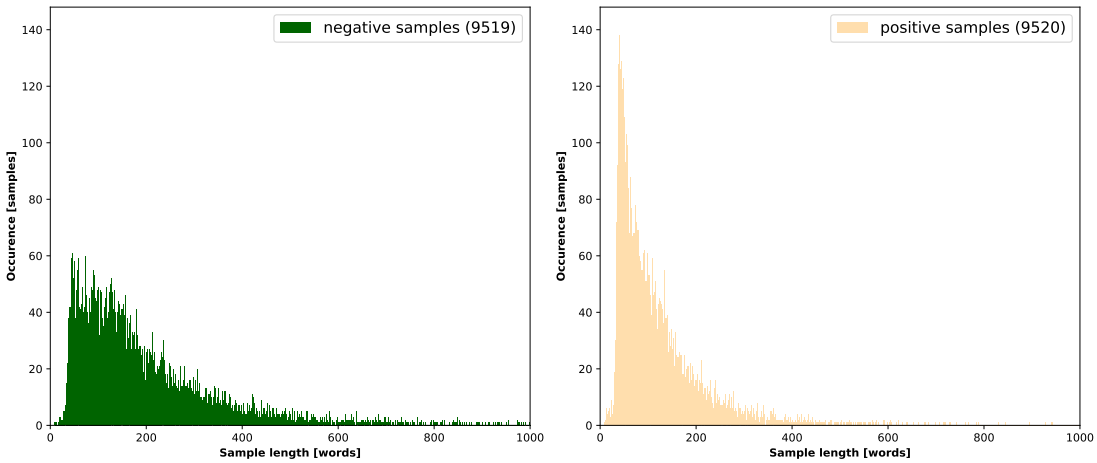


Figure 4.1: The Tripadvisor dataset's hotel reviews' sentence lengths for both negative and positive samples. With an average of 196 words compared to 117 words, negative reviews are longer than positive reviews. Samples are distributed equally among both classes for hotel reviews. When considering hotel and restaurant reviews jointly, a similar sample length distribution is obtained for classes. Also, there is no notable difference between both dataset parts in this regard.

4.2 Organic Dataset

Contrary to the previous dataset, the labeling task by Danner and Menapace 2020 was done separately to the data generation by external annotators. Randomly selected online comments posted on several news websites and forums inbetween 2007 and 2017 concerning discussion about organic food were extracted. With the help of a content analysis tool, 65 beliefs about the topic were identified. Then the beliefs are split up into entities, attributes and sentiment to formulate an aspect-based sentiment analysis problem. While the sentiment indicates if a positive, negative or neutral opinion is expressed in the respective sentence, the entity describes the target of the sentiment (Hagerer et al. 2021). Its target label is either organic food, conventional food or genetically modified organisms (GMO) since discussion revolved around these topics. Although, the majority of samples involve an organic entity and thus this dataset is referred to as the organic dataset. Lastly, attributes are divided into healthiness, price, trust, quality, environment and general. With a diverse distribution of samples among these attribute categories, predicting a sample's attribute is another option for classification. Nevertheless as a simplification, this task is limited to only organic entities and predict primarily the sentiment as depicted in figure 4.2, which shows the sample length distribution for sentiment analysis. This dataset's split is going to be identical to the original split of Danner and Menapace 2020.
Burak and Restrepo 2020 apply a Multi-Instance Network (MilNet) on this dataset to obtain an F1-score of approximately 0.60 for the sentiment analysis task. This amounts to classifying segments of each sample with an RNN and using an attention mechanism to recover the sample's label. This score will be used as a comparison during the experiments.

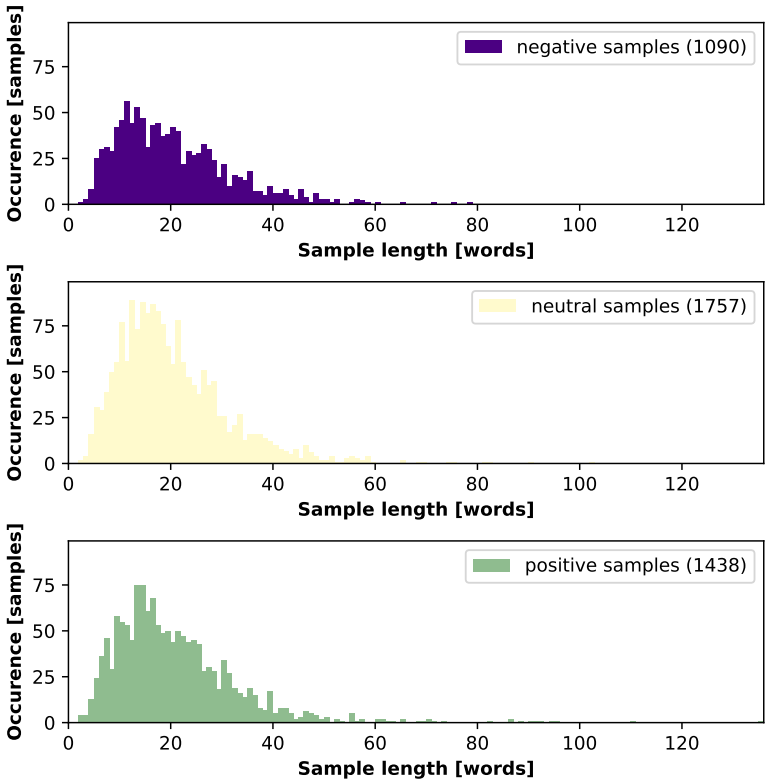


Figure 4.2: Organic dataset's sample length distribution for each class in the case of sentiment analysis. Notable, are the different numbers of samples for each class. There are 1757 neutral, 1438 positive and 1090 negative samples. Other than that, the distributions are very similar with the mean sample length at 20 words for all three classes.

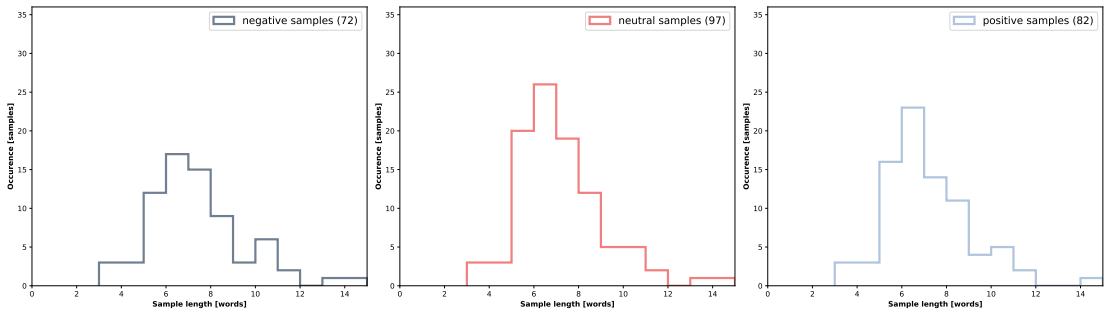


Figure 4.3: Sample length distribution for each sentiment class of the emotion dataset. With a total of 100 samples, there is some overlap for the different classes due to conflicting labels. This results in very similar sample length distributions and a mean sample length of 7 words for each class.

4.3 Emotion Dataset

In order to test the quality of annotations by Amazon's Mechanical Turk crowdsourcing system, Snow et al. 2008 generated amongst others a dataset for the task of affect recognition. Participants were asked to provide annotations for headlines consisting of an integer rating in $[0, 100]$ for six emotions: anger, disgust, fear, joy, sadness and surprise. Additionally, they gave a valence score inbetween $[-100, 100]$ that is similar to the headline's sentiment. The class distribution is very one-sided in favor of the rating 0. For this reason, a mapping to three classes is introduced: negative, neutral and positive. As a result, the valence score is somewhat balanced and can be used to perform sentiment analysis. In comparison to the previous datasets, every sample has 10 labels allowing all crowdsourcing baselines to work without pseudo-labeling. This offers a better opportunity to test IPA2LT's performance in case of a crowdsourced dataset. Although, with a total of only 100 samples the representative power of any experiments with this dataset is questionable. The sample length distribution is portrayed in figure 4.3. Since there are multiple labels per sample, many labels conflict with each other. In the figure, a sample with different labels is attributed to all assigned classes. For the experiments, a dataset split of 70-20-10 is once again applied.
Computing an emotional score for each item in the training set, Snow et al. 2008 take the mean emotional score among all items in a headline as its predicted score. Then, they evaluate these scores via Pearson correlation with expert annotations. This is rather a solution to a regression problem than a classification problem and therefore cannot be compared.

4.4 Preprocessing Steps

First of all, these datasets are available publicly and can be downloaded as text files. The individual samples are extracted either by native python operations or the whole dataset is imported with the help of the Pandas framework (2020). After discarding possible NaN values and transforming all datasets into a custom Torch dataset class (Collobert et al. 2011), tokenization of the input text is performed with the NLTK library (Loper and Bird 2002). Next, the lowercase words are mapped to their word vectors in the dataset specific embedding described in 3.2.4. During this process, padding is added to all word vectors, limiting their number of tokens to 100 for both the TripAdvisor and emotion datasets. This is done due to computational reasons or because of short sample texts in case of the latter. The organic dataset has a maximum sample token length of 136, which is adopted as the token padding length. Lastly, the data is shuffled according to a constant seed to guarantee the same exact dataset for multiple initializations.

5 Experiments

With the datasets now clearly established, it is time to fomulate multiple experiments and expectations of their results to answer the research questions. In order to compare the traditional crowdsourcing approaches with the end-to-end model of this thesis, let us first consider the datasets with many annotators.

For a typical crowdsourcing setting, the emotion dataset as well as the organic dataset are picked. The bias of each annotator is modeled as described in chapter 3 for each crowdsourcing approach. Thereby, performance of LTNet is tested as well as the performance of only its basic neural network with and without pseudo labels.
Traditional crowdsourcing approaches need multiple labels per sample. Accordingly, pseudo labels are provided for the organic dataset before applying these approaches to infer ground truth estimates. Consequently, a similar training process is performed with the estimates for both traditional approaches. For this purpose, the same basic neural network as before is trained. The only difference is the fact that only one label per sample is available instead of multiple labels. All together, the first experiment consists of training LTNet and the basic network with and without pseudo labels in addition to training the basic network on the ground truth estimates of Dawid-Skene and MACE. The next two experiments both follow this training procedure, although with other datasets. This allows capturing different biases.

Also addressing the first research question, the second experiment is designed as a crowdsourcing problem with reviewer annotators. Provided with the Tripadvisor dataset, this precondition requires combining subsets of individual annotators to form two larger subsets, one for each gender. As a result, each crowdsourcing approach has the goal of capturing group bias including category noise and incosistent labeling. Since this means dealing with a singly-labeled crowdsourced dataset, pseudo labels are required for the Dawid-Skene and MACE algorithms.

As discussed, LTNet has the capability to capture inconsistent labeling for multiple datasets with the same task. Ideally, this feature would be investigated with multiple emotion recognition datasets since they are more likely to exhibit inconsistent labeling. Instead, capturing inconsistent labeling for multiple datasets is examined

with the different parts of the Tripadvisor dataset. The intuition is that there might be inconsistent labeling due to words being associated with a different meaning for hotel reviews compared to restaurant reviews. This represents the third experiment and answers the second research question of this thesis.
In these first three experiments, a neural network is trained for each crowdsourcing approach. As an evaluation method, the classifier's accuracy and F1 score are measured on a separate test set as well as the validation set. The F1 score is defined as the harmonic mean of specificity and sensitivity. Furthermore, overfitting behavior is examined by comparing the traing and validation loss curves for the best classifiers of each approach.

Finally with the fourth experiment concerning the third research question, all mentioned crowdsourcing approaches are examined in a different light by comparing their ground truth estimates. This only makes sense for a crowdsourcing setting with many annotations per sample. Therefore the evaluation of this question is constricted to the emotion as well as the organic dataset. For the latter, pseudo labels are required once again. The inferred labels of all relevant crowdsourcing approaches will be compared by measuring the overlap of samples as well as by calculating the Krippendorff's alpha scores that represent label agreement. Ground truth estimates of LTNet are obtained by recovering the highest prediction of the basic network. As another comparison, the majority labels are also considered.
Despite now explaining all experiments briefly, this section is structured according to the mentioned crowdsourcing approaches. This means explaining the necessary setup for every approach concerning the first three experiments. Afterwards, the fourth experiment is expanded on separately.

5.1 LTNet

Before describing how LTNet is applied for the experiments in detail, let us focus on the initialization process of relevant parameters and the hyperparameter selection that is performed in the same fashion for the first three experiments.
A complete overview of the training procedure for LTNet in all experiments including all initialization steps is available in procedure 5.1. The Basic model refers to the neural network inside LTNet without the confusion matrices on top.

Initialization

For the initialization of the basic classifier, the established process of Zeng et al. 2018 is closely followed. Firstly, a basic classifier is trained on the complete dataset of the respective experiment. This step will be referred to as pretraining and the

hyperparameters are chosen as outlined in the next section. Once the training is finished, metrics of the different models are compared and the model with the highest micro F1 score is selected. It serves as the initialization for the basic classifier part of LTNet in all hyperparameter configurations of the respective experiment.
Secondly, the bias matrices on top of the basic classifier are initialized identical to the initialization in Zeng et al. 2018. They are set to the sum of an identity matrix with small random positive values and normalized along each row to maintain a probability distribution over all categories.

Hyperparameters

As always, the most important hyperparameter is the optimizer learning rate. Instead of a grid search to find a promising learning rate, learning rates are drawn randomly from the interval $[1e{-}6, 1e{-}3]$. Actually, to account for this interval being larger towards the upper bound, learning rates are sampled randomly from $\log_{10}$ space.
Introduced in chapter 3, the negative likelihood loss from equation (3.11) is used as a loss function for LTNet and the basic classifier. The SGD optimizer will be initialized with a momentum of 0.9 and a weight decay of 0.0005. As the batch size was found to be unimportant as a hyperparameter, it was fixed to 64 samples. In the training script, 20 different learning rates are drawn during pretraining as well as in the regular LTNet training phase. The training goes on for a maximum of 300 epochs, but an early stopping mechanism is employed in case of early convergence. If the change in validation loss is below the margin of $1e{-}5$ for 10 consecutive epochs, convergence is assumed and the training procedure is stopped. These hyperparameters reflect a compromise of exploring different learning rates and training duration, both limited by the available hardware.
Tables 5.1 and 5.2 depict the main hyperparameters of the pretraining and training phases for the first three experiments.

Training Procedure 5.1: LTNet Experiment

Input : Dataset with N samples, observable labels $\boldsymbol{y^c}_{true} = \{y_j^c | j \in \mathcal{N}_c \subseteq \mathcal{N}\}$ and index sets $\mathcal{N}_c = \{j | \text{c provided label for sample } x_j\}$ and $\mathcal{N} = \{1, \ldots, N\}$ for annotator c. There are C annotators and L categories.

Output: Multiple LTNet/Basic models with/without pseudo labels

Initialization

Pretraining Train Basic model Θ_{init} on all annotations $\boldsymbol{y^1}_{true}, \ldots, \boldsymbol{y^C}_{true}$ disregarding the information about different annotators. Set the basic model part to $\Theta = \Theta_{init}$ for all training experiments

Bias matrices Set every bias matrix to $T^c = \mathbb{1}_L + 0.1 \cdot \mathbb{U}_L$ and normalize its rows. $\mathbb{1}_L$ refers to the identity matrix and $\mathbb{U}_L$ to a $L \times L$ matrix with values $\mathbb{U}_{ij} \in [0, 1]$ drawn from a uniform distribution

Indivdual Training Train machine annotator Θ^c on observed labels $\boldsymbol{y^c}_{true}$ to provide pseudo labels $\boldsymbol{y^c}_{pseudo} = \{y_j^c | j \in \mathcal{N} \setminus \mathcal{N}_c\}$ and complement already existing labels. This is repeated for every annotator and together with the true labels form the complete vector of labels for an annotator $\boldsymbol{y^c} = [y_1^c, \ldots, y_N^c]$

Training

LTNet$_{true}$ Train LTNet $(\Theta, T^1, \ldots, T^C)$ on all observed labels $\boldsymbol{y^1}_{true}, \ldots, \boldsymbol{y^C}_{true}$

Basic$_{true}$ Train Basic model Θ on all observed labels $\boldsymbol{y^1}_{true}, \ldots, \boldsymbol{y^C}_{true}$

LTNet$_{pseudo}$ Train LTNet $(\Theta, T^1, \ldots, T^C)$ on all labels including pseudo labels $\boldsymbol{y^1}, \ldots, \boldsymbol{y^C}$

Basic$_{pseudo}$ Train Basic model Θ on all labels including pseudo labels $\boldsymbol{y^1}, \ldots, \boldsymbol{y^C}$

5.1.1 Experiment 1: One Matrix per Annotator

For the first experiment, datasets with distinguishable individual annotators are considered. A separate pretrained model is trained for each dataset and also the case of adding pseudo labels is considered.

Emotion Dataset

Introduced in chapter 4, the emotion dataset was annotated by 38 people and has 10 labels per sample. With pseudo labels, the number of labels per sample increases to 38. In total, the emotion dataset only has 100 samples. This means it is difficult to train a good neural network as they usually require much larger datasets. However, it might be possible since a very simple network that also works for relatively small datasets was chosen. Adding to the difficulty, a crowdsourced dataset with many annotations can introduce multiple types of noise. With 38 annotators, inspecting noisy labels due to bad annotators or labeling inconsistencies is too much manual labour as it would mean comparing all 38 bias matrices. Hence, this evaluation is constricted to only consider category noise by averaging over all bias matrices.
Considering pseudo labels, it is notable that the number of samples is increased by 28 labels each. This also increases the number of gradient updates. Given that the disagreement between labels stays constant or diminishes, better results of the models trained with all labels including pseudo labels could be expected.

Organic Dataset

The organic dataset has 10 annotators and is singly-labeled. Applying pseudo labels increases the number of labels per sample to 10. Therefore, models trained on all labels including pseudo labels could be optimized much faster due to 10 times as many gradient updates compared to the singly-labeled case. Albeit this is only true in case of low disagreement among labels.
With 4285 samples in total, the organic dataset is much better suited for training a neural network. Again, noisy labels caused by bad annotators and possible labeling inconsistencies are ignored and only category noise is considered. In case of high noise, one could expect better performance of LTNet compared to just the basic neural network.

5.1.2 Experiment 2: One Matrix per Group

As it would be beneficial to be able to perform crowdsourcing algorithms for reviewer annotated datasets, the crowdsourcing approaches relevant for this thesis are tested with the Tripadvisor dataset. As explained, each gender is represented by one bias matrix in LTNet, treating them as a single annotator each. As a result, a group refers to all annotators of one gender in the context of this experiment. Out of both parts of the Tripadvisor dataset, the hotel reviews part is used as it is smaller compared to the restaurants part. The organic dataset is discarded for this experiment as it only has 10 annotators in total. With so few annotators, no gains are to be expected when modeling whole groups of people. The exact same steps as in experiment 1 and training procedure 5.1 are performed. Again, the only difference is that one annotator actually denotes a group of people instead of just one person. Therefore, LTNet and the other approaches deal with C annotator groups instead of C annotators.
Now that there are only two annotator groups, both bias matrices are directly compared in the evaluation process for detection of noise. Thereby, category noise is considered as well as noisy labels due to labeling inconsistencies. Noisy labels due to a bad annotator is unlikely to have a major impact given that whole annotator groups are considered.

Experiments	Individual Training	Pretraining
Experiment 1		
Draws	10	20
Epochs	100	200
Experiment 2		
Draws	10	20
Epochs	100	200
Experiment 3		
Draws	8	15
Epochs	100	200

Table 5.1: Main hyperparameters of the initialization phase for all LTNet experiments. Draws refer to how many different learning rates were drawn from uniform log space.

Experiments	LTNet_{true}	LTNet_{pseudo}	Basic_{true}	Basic_{pseudo}
Experiment 1				
Draws	20	20	20	20
Epochs	300	300	300	300
Experiment 2				
Draws	20	20	20	20
Epochs	300	300	300	300
Experiment 3				
Draws	15	15	15	15
Epochs	300	300	300	300

Table 5.2: Main hyperparameters of the training phase for all LTNet experiments.

5.1.3 Experiment 3: One Matrix per Dataset

To answer the second research question the different parts of the TripAdvisor dataset are used. Both parts are reviews from the same time in Britain as established in chapter 4. There will be no differentiation among genders of annotators this time, but rather both hotel and restaurant reviews are represented via a bias matrix each. The intuition is that for restaurants and hotels, different words are important for the overall rating. Hence, inconsistent labeling could be expected to occur and the resulting noise is addressed by applying LTNet. For this purpose, both datasets' bias matrices are examined as this form of noise should be captured in addition to category noise.
At this point, the training procedure 5.1 is referred to again. Accordingly, one annotator represents one dataset part. As this experiment deals with 3 times the amount of samples than experiment 2, the number of different learning rates drawn from the uniform distribution in log space is limited to only 8 draws during individual training and 15 draws for all other training phases including pretraining.

5.2 Dawid-Skene

In order to compare the traditional Dawid-Skene crowdsourcing approach to LTNet, a complete set of labels by each annotator is required. This means that every sample needs an annotation from every annotator. The emotion dataset provides exactly this setting, although as mentioned before, it is not representative due to its small size. Therefore, Dawid-Skene is applied with pseudo labels for every other dataset. Naturally, this will lead to comparing its performance rather with the pseudo label version of LTNet than in case of only true labels.
The Fast-Dawid-Skene algorithm that was mentioned, provides an implementation, which was slightly adapted for this work. It takes as input all crowdsourced labels and gives an estimate for every sample's ground truth label. The likelihood in equation (3.1) is maximized and for higher precision, all calculations are computed in log space. The algorithm takes the tolerance for convergence of the log-likelihood as input. It acts as a lower bound for the possible change during one iteration. If the change is smaller than the tolerance, the iteration will be stopped. Furthermore, the user needs to provide a maximum number of iterations in case it doesn't converge. After finishing the iteration and obtaining the predicted labels, a basic classifier is trained with the same hyperparameters as used in a given experiment. The results of this approach will be compared to all the LTNet experiments with pseudo labels. However, since there is normally one backpropagation step per annotator for each sample and the number of labels per sample is reduced to one label, it makes sense to increase the training duration. Multiplying the number of epochs with the previous number of labels per sample counters this.

Experiment 1 Emotion	Basic$_{DS}$
Tolerance	1e-5
Draws	20
Epochs	3000
Organic	
Tolerance	1e-5
Draws	20
Epochs	3000

Experiment 2 TripAdvisor (gender)	Basic$_{DS}$
Tolerance	1e-5
Draws	20
Epochs	600
Experiment 3 TripAdvisor (datasets)	Basic$_{DS}$
Tolerance	1e-5
Draws	20
Epochs	600

Table 5.3: Hyperparameters for the Fast-Dawid-Skene algorithm.

5.3 MACE

The second traditional crowdsourcing approach is MACE, which works very similar to Dawid-Skene. During the optimization procedure of MACE, the logarithmic version of the likelihood in equation (3.14) is maximized. Although pseudo labels are not necessary for MACE, they are still included for all datasets other than the emotion dataset. This is done to retain the possibility for a comparison of both traditional crowdsourcing approaches.
In contrast to Dawid-Skene, the iteration is restarted multiple times and the best model is chosen according to the log-likelihood. A number of maximum iterations is specified instead of a tolerance signaling convergence. The optimization is continued until this number of iterations is reached. Since MACE also reduces the number of labels per sample to only one, increasing the training duration again compensates for this.

Experiment 1 Emotion	Basic_{MA}
Iterations	1000
Draws	20
Epochs	3000
Organic	
Iterations	1000
Draws	20
Epochs	3000

Experiment 2 TripAdvisor (gender)	Basic_{MA}
Iterations	1000
Draws	20
Epochs	600
Experiment 3 TripAdvisor (datasets)	Basic_{MA}
Iterations	1000
Draws	20
Epochs	600

Table 5.4: Hyperparameters for the MACE algorithm.

5.4 Experiment 4: Comparing Inferred Labels

Finally, the similarities of estimated ground truths generated by an end-to-end approach are examined and labels produced by traditional ground truth estimators. The goal of this experiment is to ascertain the ability of an end-to-end approach to act as a ground truth estimator by itself. As mentioned, the ground truth estimates of LTNet are devised by considering the highest prediction of the basic neural network for any sample. Since majority voting is a good comparison for ground truth estimators in general, the majority label is calculated for both the emotion dataset and the organic dataset utilizing pseudo labels. In case of a draw, the label is picked randomly among the highest occurring labels.
The evaluation of the different inferred labels is accomplished with two measures. Firstly, the overlap of samples with the same label for two or more approaches is counted. This is done mainly to give an intuitive measure that tells which approaches produce ground truth estimates in the same way. Secondly, the Krippendorff's alpha scores are calculated and compare inter-rater reliability in case of multiple labels per sample. It calculates a coincidence matrix consisting of weighted counts of occurences of different label configurations. Its values are then used to determine the observed disagreement and the disagreement expected by chance in order to calculate the alpha score. A score of 1 indicates perfect reliability, 0 entails that samples and their assigned labels are statistically unrelated and a negative score signifies systematic disagreement that exceeds the possible disagreement according to chance.
As a discriminative end-to-end approach, LTNet is fundamentally different in estimating ground truth labels. Instead of modeling the probability distribution of the estimates according to the observed labels, it only considers the input sample. The observed labels are only indirectly incorporated into the neural networks parameters. Therefore, the overlap with the other crowdsourcing approaches is not expected to be great. Nevertheless, the performance of this work's end-to-end approach is going to be evaluated in this regard.

6 Results

The overall goal in this thesis is to compare traditional crowdsourcing approaches to a state-of-the-art end-to-end approach. LTNet trains on all observable data directly whereas the traditional approaches first reduce it to ground truth estimates before training a classifier. The general use case of training a classifier in combination with a crowdsourcing approach is to leverage crowdsourcing as a cheap method of generating a dataset and then obtain a good classifier for the desired task. Notably, the goal is not to predict the label each annotator would provide. Usually obtaining a classifier that is able to predict one label only for each sample for the task at hand is desired. Therefore the ground truth predictions are much more important.
For this reason, the evaluation is done on the ground truth predictions of each approach's classifier. Unfortunately, none of the datasets in this thesis have an adequate test set with ground truth labels from expert annotations. Consequently, the labels of test and validation sets are reduced to ground truth estimates for the evaluation process. This works well in case of many annotators. As an alternative for datasets with few annotators, the performance of different approaches can be evaluated on each annotators labels separately.
To measure performance of classifiers, two major metrics are employed. Besides accuracy, the F1-score is calculated. It is the harmonic mean between a classifier's specificity and sensitivity. In order to prevent errors, the calculation of both metrics was delegated to a Scikit-learn function (Pedregosa et al. 2011).
Furthermore, the loss curves of all classifiers are inspected in order to detect overfitting. For each dataset a split into training, validation and test sets was introduced. Classifiers are trained with the training set and indirectly also with the validation set, since validation set performance is optimized by the hyperparameter search. Therefore, the test set performance becomes the most crucial measure. Although, the datasets in this thesis are quite small, oftentimes resulting in multiple models sharing the highest test set score. In that case, the validation set performance becomes decisive for choosing the best model.
The third research question of this thesis is to investigate the similarity of ground truth estimates generated by an end-to-end approach to the estimates produced by traditional ground truth estimators. For LTNet, the latent truth predictions can be interpreted as ground truth estimates. Thus, they are compared to the inferred labels

of the Dawid-Skene algorithm, MACE and majority voting. The overlap is measured by counting samples and by calculating the Krippendorff's alpha score. It is a measure for the inter-rater reliability and relies on a weighted count of occurences of different label configurations summed up in a coincidence matrix. To calculate this measure, an implementation by the NLTK library is used (Loper and Bird 2002). The overlap of inferred labels is evaluated in section 6.4.2.

6.1 Experiment 1

The first experiment is about testing LTNet's performance when respresenting one annotator with one bias matrix. This was done for the organic and the emotion dataset. Since this is a crowdsourcing problem, Dawid-Skene and MACE were also applied in order to produce ground truth estimates and then train a regular classifier. The performances of all approaches can be found in table 6.1. As mentioned, the test and validation set labels were reduced to one label per sample. For this purpose, majority voting is employed although the inferred labels greatly overlap with Dawid-Skene ground truth estimates.

6.1.1 Performance

Emotion Dataset

With this kind of evaluation, a clear advantage of the classifiers trained on Dawid-Skene and MACE labels is observed for the emotion dataset. This is very reasonable as the neural network only has 70 samples for training. On top of that, each sample has 10 annotations for the case of only true labels and 38 annotations when training with pseudo labels. 28 more instances of each sample also means 2.8 times more gradient updates per epoch. This might explain the better performance on the validation set. The test set scores are less representative with only 10 total samples and corresponding ground truth estimates. Thus, the better test set performance for true labels compared to pseudo labels might be an anomaly.
With so many annotations per sample a high noise ratio is to be expected for the emotion dataset. Therefore, a noise reduction technique such as confusion matrices to increase loss for noisy labels might yield better performance. As specified, only category noise is considered in this evaluation. Although it is only detected if individual classes have high noise. With high noise for every class, it would be unreasonable to expect any performance increases. The same is true in case of low noise for every class. For this reason the average confusion matrix of LTNet, as defined in section 3.2.1 for

one annotator, is examined both for true and pseudo labels,

$$\boldsymbol{T}^{avg}{}_{true} = \begin{pmatrix} 0.902 & 0.050 & 0.048 \\ 0.069 & 0.867 & 0.064 \\ 0.049 & 0.049 & 0.902 \end{pmatrix}, \quad \boldsymbol{T}^{avg}{}_{pseudo} = \begin{pmatrix} 0.905 & 0.048 & 0.047 \\ 0.115 & 0.787 & 0.098 \\ 0.044 & 0.044 & 0.922 \end{pmatrix},$$

signaling high consensus among annotators and slightly more noise for the neutral class for the true labels only. Still, close to 90% in a confusion matrix is not very noisy and thus no performance increase can be expected from applying LTNet on true labels. This is in line with the reported findings, the only exception being a better test set performance of LTNet for true labels compared to the basic classifier.
The results of training with pseudo labels are different though. As the diagonal element of the neutral class is 0.787, an increase of performance for LTNet could be expected. This is not the case, both LTNet and the basic network share exactly the same performance. Considering the loss curves in the following section, pseudo label LTNet models did not overfit. This might be a consequence of penalizing noisy labels with a higher loss. Although noisy labels are assigned higher loss, apparently this does not lead to an overall better classifier in this case. Naturally, the hypothesis about lower bias matrix parameters contributing to the network becoming more ambiguous when faced with noisy samples therefore increasing the networks decisive prowess, is drawn into questioning. Nevertheless, it is important to mention that the neutral category is the most prominent of all categories and it is considered in both test and validation sets. Maybe, since the influence of neutral samples on the network is decreased, it is possible to find better performance when only considering the negative and positive categories. This is an interesting hypothesis and worthy of further exploration in future work.

Organic Dataset

For the organic dataset, there is no clear advantage of using the Dawid-Skene algorithm or MACE for inferring labels and then training a neural network. With a more reliable test set, LTNet shows better performance in terms of training the basic classifier. Considering noisy labels for different classes in case of true and pseudo labels, the average confusion matrices are

$$\boldsymbol{T}^{avg}{}_{true} = \begin{pmatrix} 0.907 & 0.047 & 0.046 \\ 0.120 & 0.734 & 0.146 \\ 0.046 & 0.045 & 0.909 \end{pmatrix}, \quad \boldsymbol{T}^{avg}{}_{pseudo} = \begin{pmatrix} 0.919 & 0.040 & 0.041 \\ 0.185 & 0.688 & 0.127 \\ 0.044 & 0.043 & 0.913 \end{pmatrix}.$$

This means that the neutral class has significantly more noise compared to the positive or negative class. Therefore, higher performance of LTNet is expected compared to the basic network. Indeed, slightly better performance is found for LTNet with and without pseudo labels. This result is in contrast to the result for the emotion dataset. Overall training directly on the observed data yields greater performance, rather than reducing the labels with Dawid-Skene or MACE first. The classifier trained with the inferred labels by MACE actually has the best performance on the validation set. It does not generalize well to the test set though. For this reason, the classifier might have indirectly overfitted to the validation set. Since the organic dataset is a singly-labeled crowdsourced dataset, pseudo labels are required in order to meaningfully apply both label reduction techniques. This means that errors of the individual classifiers are propagated to the pseudo labels as is visible from the increased noise in the average pseudo label bias matrix. Naturally, an approach that incorporates the inference of a latent truth directly from the input data and the training of a classifier into one end-to-end method has an advantage since it does not need pseudo labels. Although, the performance of LTNet with pseudo labels is comparable to LTNet with true labels suggesting LTNet can deal with increased noise. Consequently, considering the input data during the inference part probably is the reason for the increased performance. Lastly, the basic classifier that is trained in this thesis does not match the F1 score of 0.60 by Burak and Restrepo 2020. This shows that a simple attention model is not the best approach for solving the sentiment analysis task for the organic dataset.

Experiment 1 Emotion	LTNet$_{true}$	LTNet$_{pseudo}$	Basic$_{true}$	Basic$_{pseudo}$	Basic$_{DS}$	Basic$_{MA}$
Test Set						
Accuracy	50.00%	25.00%	37.50%	25.00%	62.50%	**75.00**%
F1$_{macro}$	50.00%	14.81%	30.16%	14.81%	52.22%	**71.00**%
Validation Set						
Accuracy	37.50%	62.50%	37.50%	62.50%	**75.00**%	**75.00**%
F1$_{macro}$	20.00%	25.64%	20.00%	25.64%	**68.52**%	61.11%
Organic						
Test Set						
Accuracy	**47.63**%	**47.63**%	47.37%	46.58%	45.79%	43.68%
F1$_{macro}$	**42.52**%	40.33%	**42.52**%	38.07%	35.12%	29.37%
Validation Set						
Accuracy	39.23%	41.16%	39.23%	41.16%	41.48%	**42.12**%
F1$_{macro}$	35.23%	34.76%	35.05%	31.34%	34.18%	**37.17**%

Table 6.1: Performance of LTNet compared to competing crowdsourcing approaches for the organic and emotion dataset. In both cases, majority voting was used to reduce the crowdsourced labels of the validation and test set. Classifiers trained with inferred labels via the Dawid-Skene algorithm or MACE show significantly better performance on the emotion dataset due to its limited size and its multiple labels per sample. For the singly-labeled organic dataset, LTNet as well as the basic classifier yield better performance. Because the input texts are longer and more difficult compared to the emotion dataset, considering the input during the inference reasonably leads to increased performance.

6.1.2 Overfitting Analysis

Emotion Dataset

To recapitulate the training process for LTNet and the basic classifier, ahead of training the regular network a pretraining step is done. For the first 200 epochs only the basic classifier is trained on all of the observed data without pseudo labels. This is the reason for a drastic change in figure 6.1 at step 200. I decided to include the pretraining process in these loss curves because otherwise, a comparison with the loss curves of the basic classifiers trained on labels inferred by the Dawid-Skene algorithm or MACE would be useless. To be comparable and have the same amount of backpropagation steps in the basic network, Dawid-Skene and MACE classifiers were trained on the same amount of labels. With 10 labels per sample, a basic network inside of LTNet or on its own would have 10 backpropagation steps for each sample. On the contrary, Dawid-Skene and MACE classifiers only have one step per sample. In this context, the pretraining process is not viewed as part of the overall training process, but rather as an initialization for the basic network. Therefore, training the Dawid-Skene and MACE classifiers for 10 times as many epochs is reasonable. For pseudo labels, one would actually have 38 different labels and thus could choose to stop at 79 epochs to account for this. As a simplification LTNet and basic models were all trained for 300 epochs. Furthermore, for every method in figure 6.1 the loss curves of the three best models are depicted. The dashed lines are the validation losses of the respective models. The LTNet losses are averaged across annotators and all losses represent the average loss of one sample only.
Clearly, the best LTNet and basic models heavily overfit for true labels, whereas LTNet models with pseudo labels have the lowest loss overall. As mentioned before, this supports the hypothesis that the higher loss for noisy labels acts as a regularization preventing overfitting. Furthermore, the good training behavior of pseudo label LTNet models might also be explained by the availability of more labels per sample. Basic models with pseudo labels generally did not overfit with the exception of the $1.64e{-6}$ model. Dawid-Skene and MACE models also show good training behavior, although the Dawid-Skene validation losses do not change much over time. The training losses are overshadowed by the training loss of the 0.000967 model, which has a much better training behavior reaching a per sample loss of -0.8. The MACE models continuously improve, noticeably decreasing their validation losses with more epochs, albeit staying in the -0.4 loss range. Having a higher loss means that their predictions are not as certain compared to the 0.000967 Dawid-Skene model as an example. All of these training behaviors are consistent with performance metrics since the MACE model performs best, followed by the Dawid-Skene and the pseudo label models.

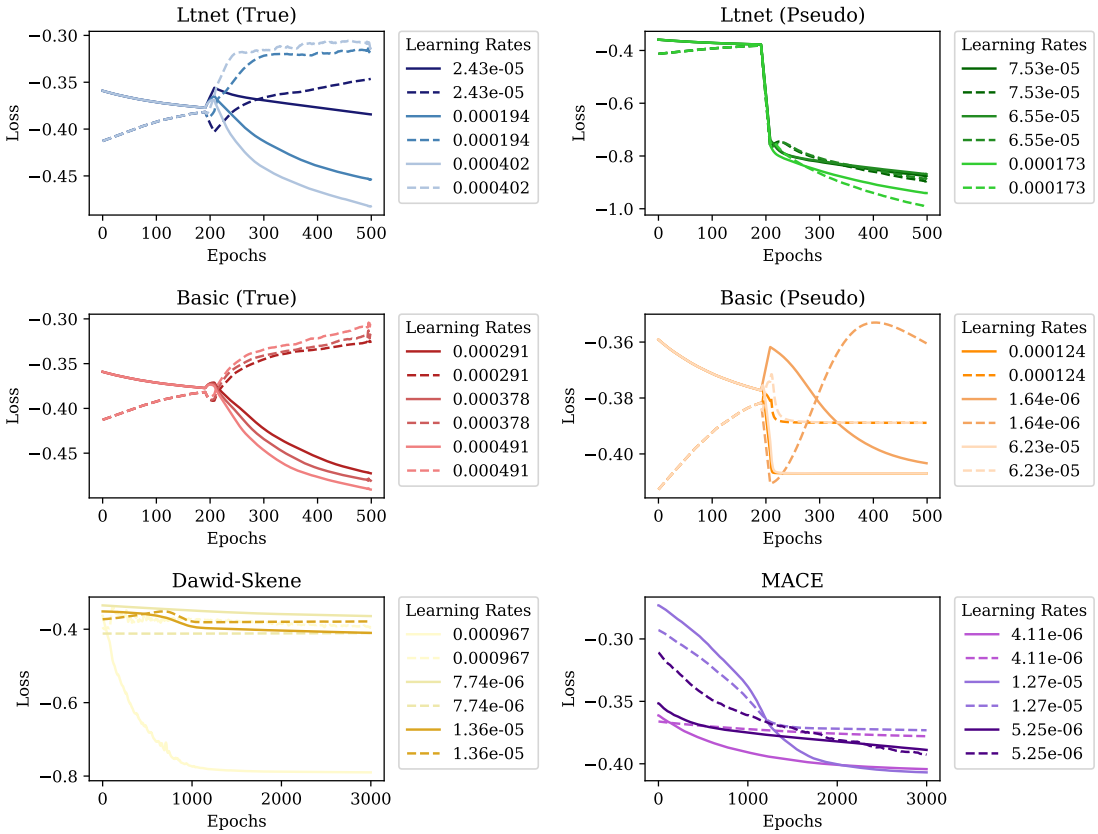


Figure 6.1: Training and validation (dashed) losses for the training process of all different approaches for the emotion dataset. The first 200 epochs of both LTNet and basic models actually originate from the same pretraining process and were added for comparability. The LTNet and basic classifiers overfit for true labels, whereas the training process of MACE, Dawid-Skene and pseudo label models is nominal.

Organic Dataset

For the organic dataset most models trained well, the only exception being the 0.000120 LTNet model with true labels. All loss curves are depicted in figure 6.2 in the same way as the loss curves of the emotion dataset in figure 6.1. Interestingly, only the LTNet model with true labels show a major increase in loss after the pretraining phase. This could be due to various reasons, but as it appears in every model it might be caused by an inopportune sequence of gradient updates weighted by the confusion matrix elements. In comparison to the pseudo label LTNet version, it does not quickly converge to a loss of -1.0, making the initial spike much more visible. Speaking of the Ltnet pseudo label models, again they show the lowest loss together with the Dawid-Skene models. As both LTNet versions exhibit the highest accuracy and F1 score, it is apparent that magnitude of loss does not correlate much with decisive prowess of a classifier. It rather represents the certainty of a prediction, i.e. the numeric value of a prediction is very close to its respective label. Still, pseudo labels are seemingly easier to predict for the LTNet model when modeling an annotator by one bias matrix. For the basic models this difference is not as large. None of the models overfit and they show reasonable behavior. The LTNet loss curves are furthermore matching the hypothesis of high loss for noisy labels preventing overfitting. Both true and pseudo label bias matrices showed high noise for neutral classes and with only one outlier, they all trained well.
Additionally, both the Dawid-Skene and MACE models converged very quickly rendering the long training period redundant. Overall, the basic classifier tends to converge fast due to its low complexity and the organic dataset being significantly bigger than the emotion dataset. This is consistent with the basic neural network performing worse compared to a more complex model as in Burak and Restrepo 2020.

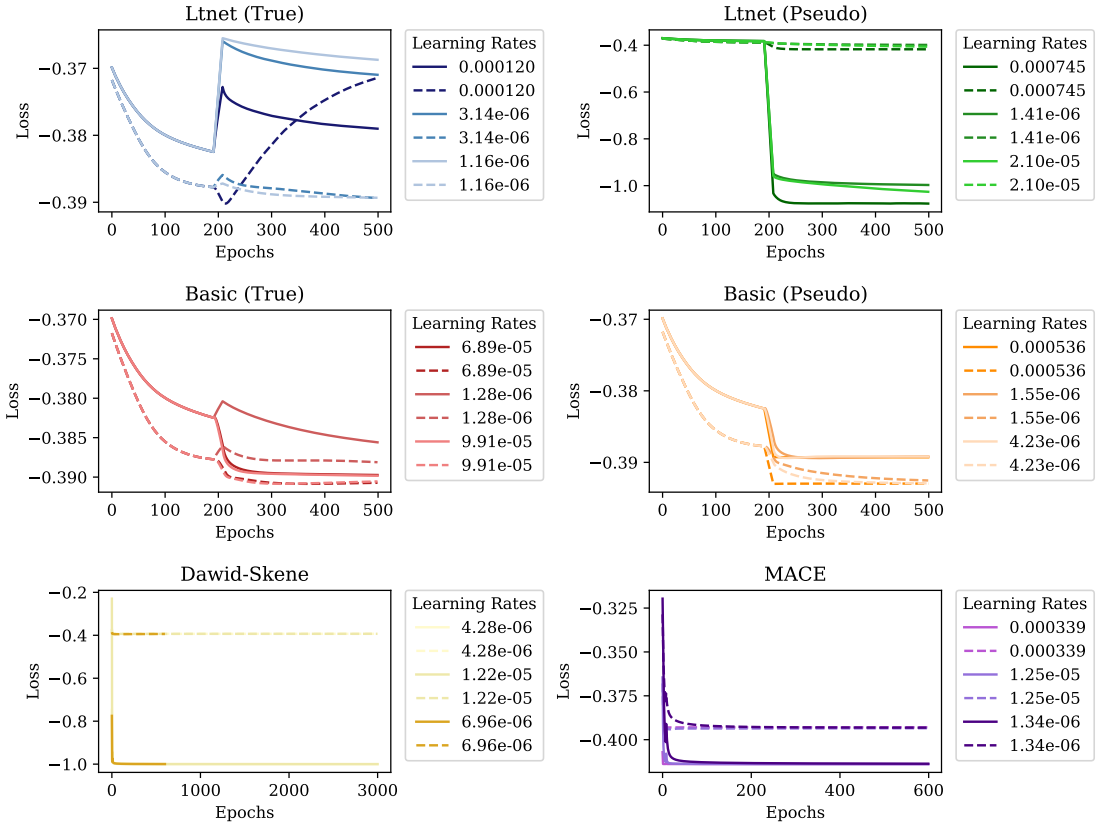


Figure 6.2: Loss curves for the organic dataset. Dashed lines represent a model's validation loss and each diagram displays 3 different models. Most classifiers show desired training behavior with only one case of overfitting for the first LTNet model with true labels. Both Dawid-Skene and MACE models converged quickly. The other approaches display similar progress demonstrating the limitations of using a simple attention model to solve this task.

6.2 Experiment 2

6.2.1 Performance

For the evaluation of classifiers, one label per sample is desirable. As the Tripadvisor dataset is a singly-labeled crowdsourced dataset providing only a binary sentiment analysis problem, applying the Dawid-Skene algorithm or MACE directly to it is not beneficial. In order to apply them in a meaningful way, pseudo labels are required to form a complete set of labels. If the labels were reduced by the Dawid-Skene algorithm for this evaluation, individual classifiers would be needed, which would introduce another source of error.
A better way of gaining an evaluation corpus is to simply evaluate on the data of each annotator group individually. This is convenient since there are merely two annotator groups in this experiment and each sample already has one label only. For these reasons table 6.2 presents the performance of all models on the test set samples annotated by men and women separately.
Generally speaking, the performance of the models trained on all observed data is superior to the models trained on inferred labels. As the performance for test samples is worse for the male annotator group, both Dawid-Skene and MACE might have chosen the female label more often during inference. In this case, the classifier would have learned the features of female data better.
To get an idea whether the classes display a lot of noise, let us take a look at the confusion matrices for true labels,

$$\boldsymbol{T}^{female}{}_{true} = \begin{pmatrix} 0.916 & 0.084 \\ 0.062 & 0.938 \end{pmatrix} , \quad \boldsymbol{T}^{male}{}_{true} = \begin{pmatrix} 0.922 & 0.078 \\ 0.075 & 0.925 \end{pmatrix} .$$

There is no major difference of bias for any of the two categories inside the confusion matrix indicating low noise. Therefore, major performance increases are not expected when accounting for noise by adding confusion matrices during the training process. Comparing with results, this hypothesis is consistent, the highest difference being 0.12% for the female validation set. Apart from that, no significant labeling inconsistencies are detected as the matrices of both annotator groups are quite similar. Accordingly, the samples of both groups have the same significance for training the basic network inside LTNet. In case of pseudo-labels, the bias matrices are very similar leading to the same interpretation as for true labels.

Experiment 2 TripAdvisor (female)	LTNet$_{true}$	LTNet$_{pseudo}$	Basic$_{true}$	Basic$_{pseudo}$	Basic$_{DS}$	Basic$_{MA}$
Test Set						
Accuracy	89.91%	89.91%	89.91%	90.01%	90.15%	**90.36**%
F1$_{macro}$	89.90%	89.90%	89.90%	90.01%	90.15%	**90.35**%
Validation Set						
Accuracy	90.93%	**91.04**%	90.93%	90.93%	88.71%	89.00%
F1$_{macro}$	90.92%	**91.04**%	90.92%	90.92%	88.71%	89.00%
TripAdvisor (male)						
Test Set						
Accuracy	**90.66**%	90.24%	**90.66**%	90.35%	87.16%	87.47%
F1$_{macro}$	**90.64**%	90.22%	**90.64**%	90.32%	87.15%	87.47%
Validation Set						
Accuracy	**89.94**%	89.77%	**89.94**%	89.77%	89.06%	89.11%
F1$_{macro}$	**89.94**%	89.77%	**89.94**%	89.77%	89.06%	89.11%

Table 6.2: Performance of LTNet compared to a basic classifier and classifiers trained on inferred labels via the Dawid-Skene algorithm and MACE for the hotels part of the TripAdvisor dataset. One bias matrix of LTNet represents one group of annotators, i.e. women or men. Performance of classifiers trained on labels inferred by Dawid-Skene or MACE is generally worse than for the other classifiers, which are trained directly on all observed labels. As this is a singly-labeled dataset, pseudo labels are computed before applying Dawid-Skene or MACE.

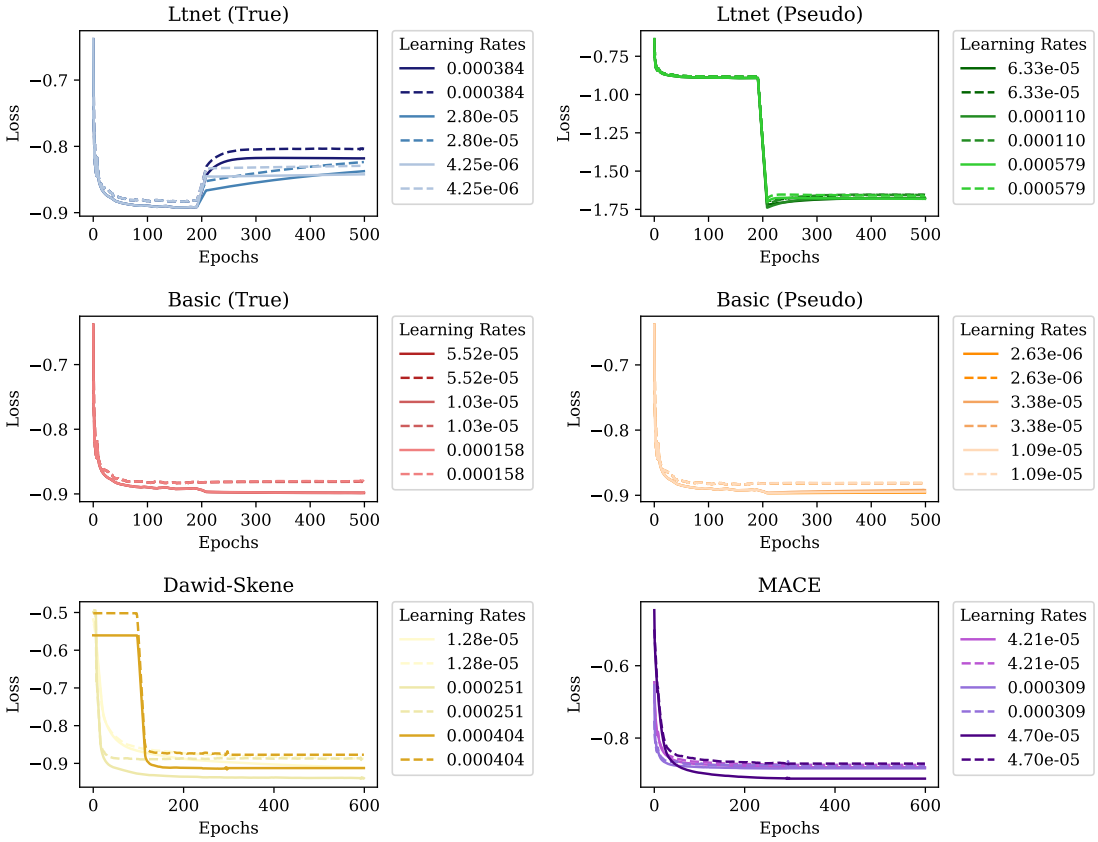


Figure 6.3: The different crowdsourcing approaches' training and validation loss curves for the Tripadvisor dataset, where one matrix represents one gender. Three models are depicted for each approach. The first 200 epochs are only part of the pretraining procedure and with the exception of the LTNet pseudo label models, all LTNet and basic models get worse with more training. This means that the pretrained network already converged. Similarly, both the Dawid-Skene and MACE models converged fast.

6.2.2 Overfitting Analysis

The most notable feature of the loss curves in figure 6.3 is that all LTNet and basic models except the LTNet pseudo label models actually get worse after pretraining. Consequently, the pretraining phase was enough for the basic network to converge and any additional training did not provide any benefit. Excluding the LTNet pseudo label model, all losses are in the -0.9 range. This means they have comparable certainty in their predictions. As in the previous experiment, the LTNet pseudo label models show the lowest loss. Their predictions are more certain, but their decision making is not consistently better as seen in table 6.2. The Dawid-Skeene and MACE models converge fast, which is in line with the basic network already converging in the pretraining procedure.

6.3 Experiment 3

6.3.1 Performance

The second research question of this thesis was to figure out whether the same crowdsourcing approaches that were investigated in the previous experiments can be applied to multiple datasets by treating them as a singly-labeled crowdsourced dataset and modeling their bias. The Tripadvisor dataset offers this opportunity with a separate dataset about restaurant reviews instead of hotel reviews. The task is still the same as for hotels, namely a binary sentiment analysis problem. However, the input data is expected to be very different. Therefore the same words can have a disparate meaning in the context of restaurants compared to hotels. This would count as inconsistent labeling and thus LTNet could provide benefits by compensating with a lower bias matrix weight for the respective class and dataset. The bias matrices are

$$\boldsymbol{T^{hot}}_{true} = \begin{pmatrix} 0.930 & 0.070 \\ 0.075 & 0.925 \end{pmatrix}, \quad \boldsymbol{T^{rest}}_{true} = \begin{pmatrix} 0.928 & 0.072 \\ 0.066 & 0.934 \end{pmatrix},$$

$$\boldsymbol{T^{hot}}_{pseudo} = \begin{pmatrix} 0.902 & 0.098 \\ 0.078 & 0.922 \end{pmatrix}, \quad \boldsymbol{T^{rest}}_{pseudo} = \begin{pmatrix} 0.922 & 0.078 \\ 0.072 & 0.928 \end{pmatrix}.$$

With no major differences, the labeling inconsistencies of both datasets are not as significant as assumed. Additionally, with only minor differences between diagonal values of categories, no discrepancy in noise distribution is detected. Thus the loss of noisy labels is not increased. The average bias matrices of LTNet with pseudo labels are also very similar, although with slightly more noise for negative hotel reviews. Since all bias matrix values are located in the 0.9 range, no performance increases are to be expected for LTNet compared to the basic classifier.

Considering the performance of approaches in table 6.3, the competing crowdsourcing approaches trained on a set of labels inferred from regular and pseudo labels are found to perform substantially worse than the methods trained directly on all observed labels. Comparing them with the pseudo label versions of LTNet and the basic classifier, the Dawid-Skene and MACE models apparently picked the pseudo labels as their own prediction rather often. Since the evaluation is done on the test sets of both dataset parts separately, the availability of both real and pseudo labels during training is certainly beneficial.
LTNet performs marginally better with true labels compared to pseudo labels. This might be due to 2% more noise in case of the positive category. But as all negative hotel reviews would be discounted more than positive hotel reviews or restaurant reviews, leading to better performance on restaurant reviews, noise as the reason for better performance is unlikely. No improvement of pseudo label LTNet models is measured for restaurant reviews. Presumably, in this experiment pseudo labels just introduce more irrelevant samples considering the evaluation in table 6.3 is done without pseudo labels. Overall, the noise differences are not as significant as to disprove any advantages of LTNet. The results confirm this with the highest difference being of a magnitude of 0.37% for the hotels test set pseudo label model.

6.3.2 Overfitting Analysis

The loss curves of figure 6.4 show much similarity to the training and validation losses of experiment 2. All models except the LTNet pseudo label models converged in the same loss range of -0.9 and only the basic pseudo label model displays minor overfitting tendencies. The LTNet models for true labels get worse compared to the final pretaining loss. This indicates that the basic classifier already converged during the pretraining phase. The fast convergence of the Dawid-Skene and MACE models confirms this hypothesis. Although, the validation losses of Dawid-Skene models seem to be higher than their training losses. All in all, no model overfits and the long pretraining phase made it possible for the network to converge before the actual training phase.

Experiment 3 TripAdvisor (hotels)	LTNet$_{true}$	LTNet$_{pseudo}$	Basic$_{true}$	Basic$_{pseudo}$	Basic$_{DS}$	Basic$_{MA}$
Test Set						
Accuracy	**90.24**%	89.98%	90.19%	89.61%	84.46%	85.15%
F1$_{macro}$	**90.24**%	89.98%	90.19%	89.61%	84.42%	85.14%
Validation Set						
Accuracy	**88.60**%	88.31%	88.49%	88.16%	84.85%	85.35%
F1$_{macro}$	**88.60**%	88.31%	88.49%	88.16%	84.84%	85.35%
TripAdvisor (restaurants)						
Test Set						
Accuracy	89.21%	89.12%	**89.24**%	89.00%	83.54%	85.94%
F1$_{macro}$	89.20%	89.11%	**89.23**%	88.99%	83.54%	85.92%
Validation Set						
Accuracy	88.49%	88.48%	88.57%	**88.59**%	83.96%	86.17%
F1$_{macro}$	88.49%	88.48%	88.57%	**88.59**%	83.95%	86.17%

Table 6.3: Performance metrics of different crowdsourcing approaches. Their performance is investigated for the case of treating multiple datasets as a singly-labeled crowdsourced dataset. As the Tripadvisor dataset contains only two separate datasets, hotel and restaurant reviews respectively, an evaluation by reducing multiple labels to one ground truth label is not reasonable. Accordingly, it is performed on the test set of each one independently. This leads to Dawid-Skene and MACE models having worse performance and shows their limitations. There is only slight variation of performance metrics for LTNet models and basic models.

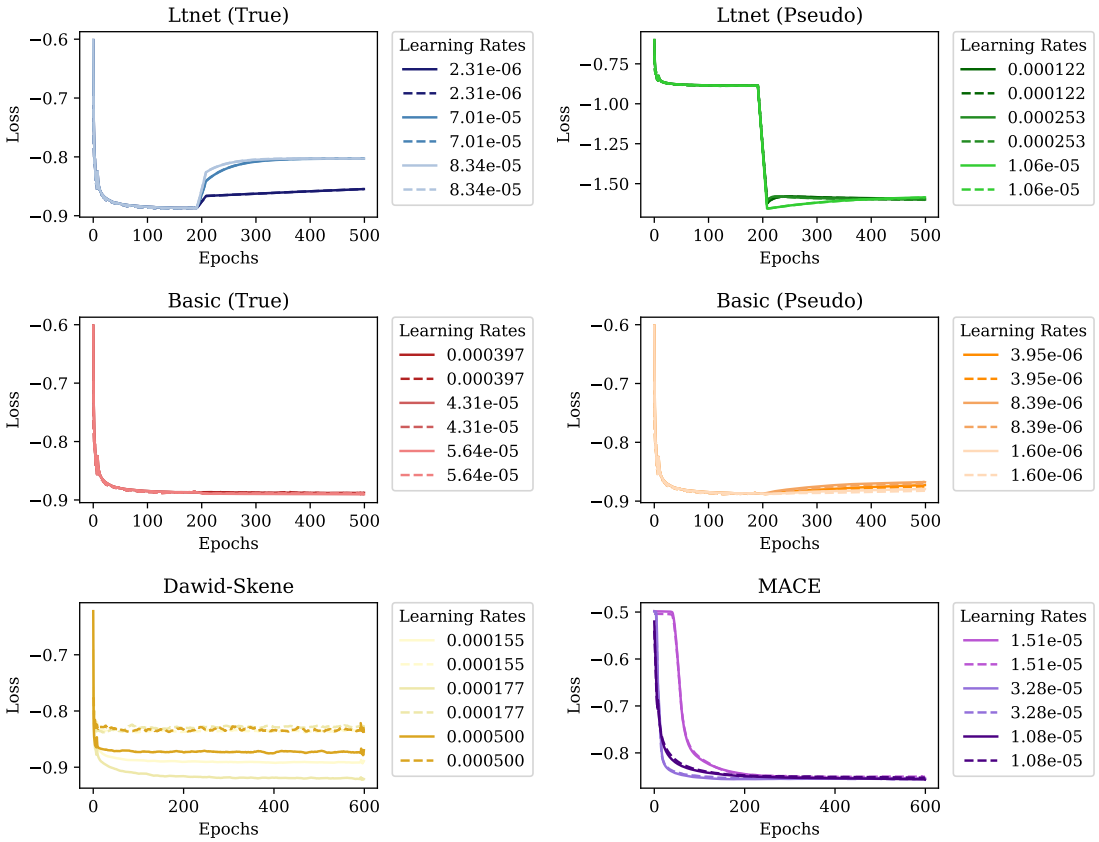


Figure 6.4: Training and validation losses for experiment 3 and the Tripadvisor dataset, where one bias matrix represents one dataset. No overfitting is detected for any model. Since the basic classifier converged fast for this task, an extensive pretraining phase of 200 epochs leads to the network already having converged before the actual training phase starts. Likewise, the Dawid-Skene and MACE models converged quickly.

6.4 Experiment 4

In this last experiment, the ground truth estimates of all crowdsourcing approaches are compared for the emotion and organic dataset. With the goal of ascertaining the ability of LTNet to act as a ground truth estimator by itself, the overlap of labels among crowdsourcing approaches is investigated. LTNet acts as a ground truth estimator by means of considering the highest prediction of its basic neural network as its ground truth estimate. As a baseline crowdsourcing solution, majority voting is also employed. In case of a draw of two or more labels, the ground truth estimate is decided randomly from among the most occurring labels.
For this evaluation, the overlapping samples for each combination of crowdsourcing approaches are counted. Additionally for each combination, the Krippendorff's alpha score indicating agreement among labels is calculated. If it is 1, there is only agreement, whereas for a value of 0, samples and their assigned labels are statistically unrelated. For negative values, disagreements are systematic and greater than what can be expected by chance.
The organic dataset requires pseudo labels being a singly-labeled crowdsourced dataset. Once again, it is noted here that LTNet does not rely on observed or pseudo labels but instead only considers the input sample. Therefore, it would be unaffected by the introduction of pseudo labels. In this case, differences between inferred labels of LTNet and the rest of approaches could be attributed to the pseudo label noise affecting all other approaches. Hence, utilizing pseudo labels is not ideal but there is no other option for the organic dataset. It will be seen whether this affects the comparison in any noticeable way. Of course, as a true crowdsourced dataset, the emotion dataset remains without pseudo labels.

6.4.1 Emotion Dataset

All training samples with their 10 annotations per sample are considered for this dataset. There are already enough labels per sample for the traditional crowdsourcing approaches, so no pseudo labels are required. This means no further noise will be introduced to majority voting, the Dawid-Skene algorithm or MACE.
Every possible combination of crowdsourcing approaches is depicted in figure 6.5. The grey bars indicate the relative amount of samples of the respective combination, whereas the red bars represent its alpha score. A combination is defined insofar it includes only samples with matching labels by all its approaches. One can quickly detect that the estimates by Dawid-Skene and majority voting completely match as their combination's normalized samples score is 1. This indicates that all samples are included in this combination. Now that there is a common understanding of how to

read figure 6.5, the correlation of high alpha scores for a combination including LTNet is observed. LTNet only matches with other approaches for a maximum of close to 40% of all samples. The other three approaches overlap for more than 90% of the dataset. Consequently, in case of a combination including LTNet nearly all samples in this combination have the same labels leading to very high agreement. On the other hand, with LTNet as the only approach to often disagree, large combinations including almost all samples have significantly lower agreement.
To conclude, LTNet does not compare equally to traditional ground truth estimators when inferring ground truth labels. Of course with the emotion dataset being a very small dataset and the basic neural network only performing average on it, this result is not indisputable.

6.4.2 Organic Dataset

Moving on to the organic dataset, pseudo labels are accumulated in order to provide a multi-labeled crowdsourced dataset. This introduces noise to the Dawid-Skene algorithm, MACE and majority voting. Comparing each approaches' ground truth estimates in figure 6.6, a completely opposite picture arises of the performance of the different approaches. Again, the agreement is high for combinations in which almost all approaches share the same labels. Although now, the estimates by Dawid-Skene, LTNet and majority voting seem to match most of the time. This leaves out MACE as the approach with many errors. Interestingly, despite MACE labels matching with the other approaches' labels for about 40% of samples, unlike LTNet in the last section, the disagreement for the full dataset is much higher here. The alpha score is actually negative indicating systematic disagreements being greater than expected by chance. Therefore, MACE learned to weight the labels of various annotators in a different way, filtering spammers during the process.
Looking back at the performance of LTNet on the emotion dataset, the alpha score was always positive when considering a combination with nearly all samples. This implies a random labeling behavior of LTNet indicating that the basic neural network did not manage to capture the task adequately.
LTNet could possibly still function as a ground truth estimator, but only when the network captured the underlying task. This could be explored further on larger crowdsourced datasets in future work.

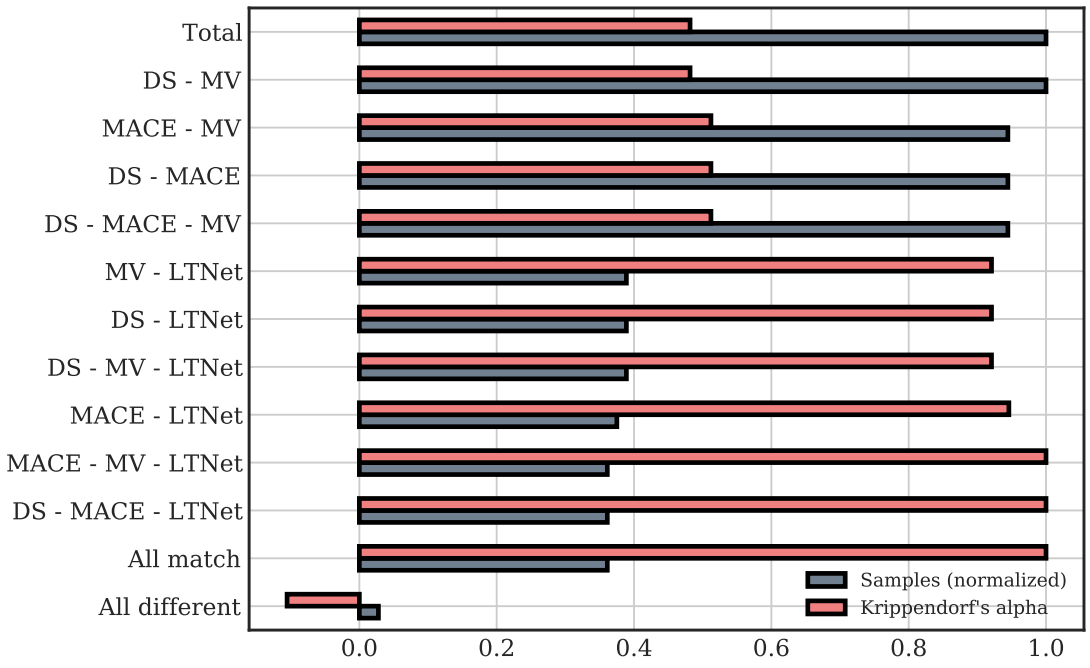


Figure 6.5: **Agreement of inferred labels for the emotion dataset**
With almost perfect agreement for all combinations of crowdsourcing approaches including LTNet, the end-to-end approach is the obvious weak ground truth estimator for the emotion dataset. However, this is not conclusive since the emotion dataset is too small and noisy for the basic neural network. The other approaches Dawid-Skene (DS), MACE and majority voting (MV) show an overlap of over 90% of all samples.

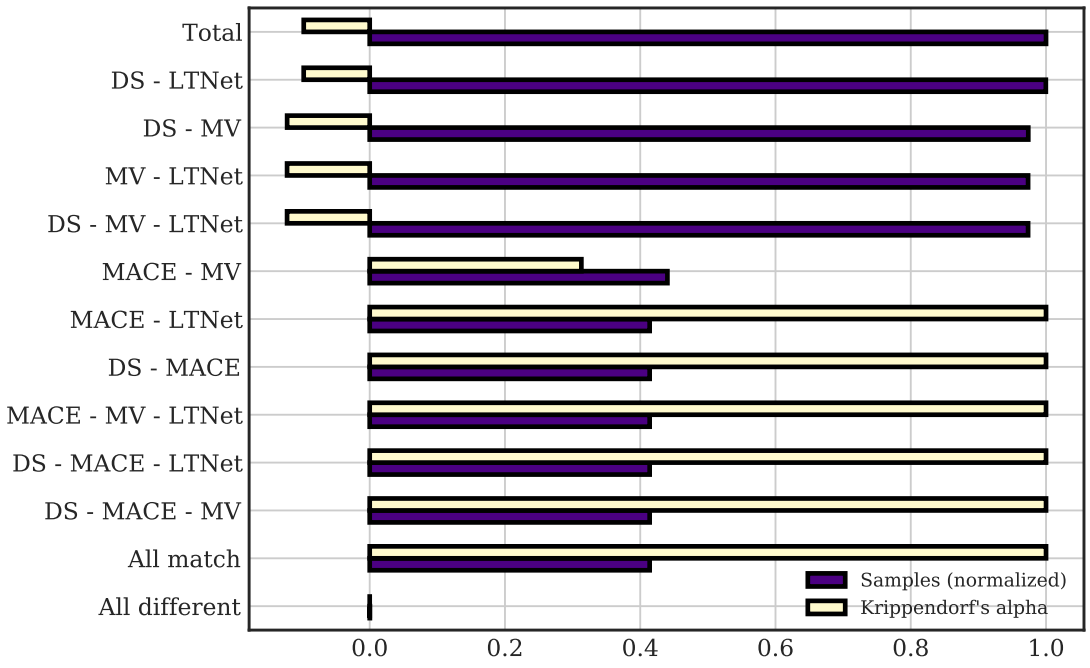


Figure 6.6: **Agreement of inferred labels for the organic dataset**
The estimates of Dawid-Skene (DS), LTNet and majority voting (MV) overlap on almost all of the data. MACE estimates disagree often, only matching with all other estimates in approximately 40% of all samples. In contrast to the differing LTNet estimates in figure 6.5, MACE seems to have learned a different mapping behavior since combinations with all samples show high disagreement via a negative alpha score.

7 Conclusion

This thesis was premised on three research questions. Overall, the goal was to examine a state-of-the-art end-to-end approach in many different ways. Mainly, its performance was compared to traditional crowdsourcing approaches in a crowdsourcing setting. Additionally, combining multiple datasets into one singly-labeled crowdsourced dataset in order to apply the end-to-end model was reviewed. Lastly, the similarities between the end-to-end model and traditional approaches were compared when using it as a ground truth estimator only.
The relevant end-to-end model LTNet was presented by Zeng et al. 2018. It was adapted for this thesis by using a different loss function as well as a neural network suitable for text data. By employing confusion matrices, it has the capability of capturing category noise, inconsistent labeling and noisy labels due to bad annotators.
Furthermore, LTNet has the ability to compensate for noisy labels by two hypothesized mechanisms. Firstly, a lower bias matrix parameter leads to higher loss. Therefore, noisy labels should have a higher loss that prevents the network to overfit on them. Secondly, a lower bias matrix parameter discounts the gradient updates, presumably reducing the influence of noisy samples. The findings of this thesis support the first hypothesis although the second hypothesis is drawn into questioning. As category noise discounts the influence of all samples of a noisy category, the classifier naturally will be worse at predicting this noisy class. Therefore, an evaluation of the classifier on all categories might not be ideal. In case of a noisy category, when discounting its influence on a classifier, the classifier would be expected to perform better on non-noisy classes compared to the case when not discounting the noisy category. This is a good topic for further research with LTNet or similar end-to-end approaches and could be explored in future work.

Moving on to the first research question investigated in experiment 1 and 2:

1. **How do traditional crowdsourcing approaches compare to a state-of-the-art end-to-end approach on sentiment analysis?**

As a crowdsourcing approach, the IPA2LT framework works well, directly providing a classifier that can potentially also work as a ground truth estimator. LTNet's limitations also became clear with its application on the emotion dataset. Since training a neural network, the crowdsourced dataset has to provide substantial amounts of data for LTNet to be applicable. In contrast, for the emotion dataset the traditional ground truth estimators Dawid-Skene and MACE in combination with a classifier perform significantly better. With more data as in the organic dataset, the IPA2LT end-to-end exhibited better performance as it considers the input samples and trains directly on all observed labels of annotators.
In experiments 2 and 3, the classifier already converged in the pretraining phase, rendering all further training almost useless. Investigating the performance of LTNet without a pretraining phase could be analyzed in future work as well. Both experiments yielded better results for the end-to-end approach mainly due to the ground truth estimators being required to use pseudo labels for the estimation, ending up with potentially wrong labels. Singly-labeled crowdsourced datasets in general are a good target for an end-to-end approach. For these datasets, LTNet was much simpler to use compared to ground truth estimators. Experiment 2 shows that even reviewer annotated datasets can be used with crowdsourcing algorithms when grouping annotators together.

On the other hand, experiment 3 displays the feasibility of combining multiple datasets together in a crowdsourcing problem and examines the second research question:

2. **Can the bias of different crowdsourcing sentiment datasets be modeled successfully by the given approaches?**

Unlike the results of Zeng et al. 2018, the results in this thesis did not demonstrate a significant increase in performance for LTNet. However, the bias matrices also showed no labeling inconsistencies. Therefore, no increases in performance should be expected, compared to a basic neural network trained on all datasets simultaneously. For this reason, the answer to the second research question is inconclusive.

An end-to-end model is likely to be capable of functioning as a ground truth estimator. This was explored in experiment 4, answering the third research question:

3. **How similar is the estimated ground truth generated by an end-to-end approach to one produced by traditional ground truth estimators?**

On the emotion dataset, LTNet performed worse compared to traditional ground truth estimators and majority voting. Although, it showed more random behavior instead of a labeling inconsistency signaling that the classifier did not fully capture the task of the emotion dataset. On the other hand, for the organic dataset MACE learned a different weighting of annotators according to its model, whereas LTNet estimates are aligned with Dawid-Skene and majority voting estimates. This supports the capability of LTNet to function as a ground truth estimator.

To conclude, this thesis demonstrated the use cases and limitations of a state-of-the-art end-to-end approach while thoroughly exploring its mechanisms. Furthermore, the end-to-end approach was contrasted with traditional crowdsourcing approaches, all the while keeping the practical focus on training a classifier for the respective underlying task. End-to-end approaches are found to be useful for most crowdsourcing scenarios and should be a topic worth studying in future work.

List of Figures

List of Tables

Bibliography

Aroyo, L. and C. Welty (2013). "Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard." In: *WebSci2013. ACM* 2013.2013.

Bengio, Y., R. Ducharme, and P. Vincent (2000). "A neural probabilistic language model." In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*, pp. 893–899.

Burak, I. and P. Restrepo (2020). "Sentiment Analysis Using BERT and Multi-Instance Learning." NLP Lab Course at Department of Informatics, Technical University of Munich (TUM); https://gitlab.lrz.de/nlp-lab-course-ss2020/opinion-mining/opinion-lab-group-1.4.

Camilleri, M. P. J. and C. K. I. Williams (2019). "The Extended Dawid-Skene Model: Fusing Information from Multiple Data Schemas." In: *CoRR* abs/1906.01251.

Carpenter, B. (2008). *Multilevel bayesian models of categorical data annotation*.

Collobert, R., K. Kavukcuoglu, and C. Farabet (2011). "Torch7: A Matlab-like Environment for Machine Learning." In: *BigLearn, NIPS Workshop*.

Danner, H. and L. Menapace (2020). "Using online comments to explore consumer beliefs regarding organic food in German-speaking countries and the United States." In: *Food Quality and Preference* 83, p. 103912.

Dawid, A. P. and A. M. Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm." In: *Applied Statistics* 28.1, pp. 20–28.

Demszky, D., D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi (2020). "GoEmotions: A Dataset of Fine-Grained Emotions." In: *arXiv preprint arXiv:2005.00547*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: `1810.04805 [cs.CL]`.

Dingwall, N. and C. Potts (2018). "Mittens: an extension of glove for learning domain-specialized representations." In: *arXiv preprint arXiv:1803.09901*.

Garg, S., G. Ramakrishnan, and V. Thumbe (2021). "Towards Robustness to Label Noise in Text Classification via Noise Modeling." In: *arXiv preprint arXiv:2101.11214*.

Ghosh, A., H. Kumar, and P. Sastry (2017). "Robust loss functions under label noise for deep neural networks." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1.

Hagerer, G., D. Szabo, A. Koch, M. Ripoll, H. Danner, and G. Groh (2021). "End-to-End Annotator Bias Approximation on Crowdsourced Single-Label Sentiment

Analysis." Social Computing Research Group, Department of Informatics and Chair of Marketing and Consumer Research, TUM School of Management, Technical University of Munich (TUM).

Harinarayan, V., A. Rajarama, and A. Ranganatha (2007). "Hybrid machine/human computing arrangement." US7197459B1.

Hendrycks, D., M. Mazeika, D. Wilson, and K. Gimpel (2018). "Using trusted data to train deep networks on labels corrupted by severe noise." In: *arXiv preprint arXiv:1802.05300*.

Hovy, D., T. Berg-Kirkpatrick, A. Vaswani, and E. H. Hovy (2013). "Learning Whom to Trust with MACE." In: *HLT-NAACL*. Ed. by L. Vanderwende, H. D. III, and K. Kirchhoff. The Association for Computational Linguistics, pp. 1120–1130. ISBN: 978-1-937284-47-3.

Jiang, L., Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei (2018). "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels." In: *International Conference on Machine Learning*. PMLR, pp. 2304–2313.

Karger, D. R., S. Oh, and D. Shah (2011). "Iterative learning for reliable crowdsourcing systems." In: Neural Information Processing Systems.

– (2014). "Budget-optimal task allocation for reliable crowdsourcing systems." In: *Operations Research* 62.1, pp. 1–24.

Khetan, A., Z. C. Lipton, and A. Anandkumar (2017). *Learning From Noisy Singly-labeled Data*.

Kirange, D. and R. R. Deshmukh (2014). "Aspect Based Sentiment analysis SemEval-2014 Task 4." In: *Asian Journal of Computer Science And Information Technology*, pp. 72–75.

Kossaifi, J., R. Walecki, Y. Panagakis, J. Shen, M. Schmitt, F. Ringeval, J. Han, V. Pandit, A. Toisoul, B. Schuller, K. Star, E. Hajiyev, and M. Pantic (2021). "SEWA DB: A Rich Database for Audio-Visual Emotion and Sentiment Research in the Wild." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.3, pp. 1022–1040. DOI: 10.1109/TPAMI.2019.2944808.

Lavelli, A., F. Sebastiani, and R. Zanoli (2004). "Distributional Term Representations: An Experimental Comparison." In: *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*. CIKM '04. Washington, D.C., USA: Association for Computing Machinery, pp. 615–624. ISBN: 1581138741. DOI: 10.1145/1031171.1031284.

Lee, K., S. Yun, K. Lee, H. Lee, B. Li, and J. Shin (2019). "Robust inference via generative classifiers for handling noisy labels." In: *International Conference on Machine Learning*. PMLR, pp. 3763–3772.

Li, J., R. Socher, and S. C. Hoi (2020). "Dividemix: Learning with noisy labels as semi-supervised learning." In: *arXiv preprint arXiv:2002.07394*.

Li, Q., Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han (2014). "A confidence-aware approach for truth discovery on long-tail data." In: *Proceedings of the VLDB Endowment* 8.4, pp. 425–436.

Loper, E. and S. Bird (2002). "NLTK: The Natural Language Toolkit." In: *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Ma, X., H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey (2020). "Normalized loss functions for deep learning with noisy labels." In: *International Conference on Machine Learning*. PMLR, pp. 6543–6553.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Peldszus, A. and M. Stede (2013). "Ranking the annotators: An agreement study on argumentation structure." In: *LAW@ACL*. Ed. by S. Dipper, M. Liakata, and A. Pareja-Lora. The Association for Computer Linguistics, pp. 196–204. ISBN: 978-1-937284-58-9.

Pennington, J., R. Socher, and C. D. Manning (2014). "Glove: Global vectors for word representation." In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). *Deep contextualized word representations*. arXiv: 1802.05365 [cs.CL].

Pontiki, M., D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos (2015). "Semeval-2015 task 12: Aspect based sentiment analysis." In: *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 486–495.

Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. D. Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jimenez-Zafra, and G. Eryigit (2016). "SemEval-2016 task 5 : aspect based sentiment analysis." In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 19–30.

Raykar, V. C. and S. Yu (2012). "Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks." In: *J. Mach. Learn. Res.* 13, pp. 491–518.

Raykar, V. C., S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy (2009). "Supervised learning from multiple experts: whom to trust when everyone lies a bit." In: *Proceedings of the 26th Annual international conference on machine learning*, pp. 889–896.

Rodrigues, F. and F. Pereira (2018). "Deep learning from crowds." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Salton, G., A. Wong, and C. S. Yang (1975). "A Vector Space Model for Automatic Indexing." In: *Commun. ACM* 18.11, pp. 613–620. ISSN: 0001-0782. DOI: 10.1145/361219.361220.

Sinha, V. B., S. Rao, and V. N. Balasubramanian (2018). "Fast Dawid-Skene: A Fast Vote Aggregation Scheme for Sentiment Classification." In: *arXiv preprint arXiv:1803.02781*.

Smyth, P., U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi (1994). "Inferring Ground Truth from Subjective Labelling of Venus Images." In: *NIPS*. Ed. by G. Tesauro, D. S. Touretzky, and T. K. Leen. MIT Press, pp. 1085–1092.

Snow, R., B. O'Connor, D. Jurafsky, and A. Y. Ng (2008). "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks." In: *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 254–263.

Takamatsu, S., I. Sato, and H. Nakagawa (2012). "Reducing Wrong Labels in Distant Supervision for Relation Extraction." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, pp. 721–729.

team, T. pandas development (2020). *pandas-dev/pandas: Pandas*. Version latest. DOI: 10.5281/zenodo.3509134.

Thelwall, M. (2018). "Gender bias in sentiment analysis." In: *Online Information Review*.

Tian, T. and J. Zhu (2015). "Max-Margin Majority Voting for Learning from Crowds." In: *NIPS*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, pp. 1621–1629.

Wauthier, F. L. and M. I. Jordan (2011). "Bayesian Bias Mitigation for Crowdsourcing." In: *NIPS*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, pp. 1800–1808.

Weizenbaum, J. (1966). "ELIZA—a computer program for the study of natural language communication between man and machine." In: *Communications of the ACM* 9.1, pp. 36–45.

Welinder, P., S. Branson, S. Belongie, and P. Perona (2010). "The Multidimensional Wisdom of Crowds." In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., pp. 2424–2432.

Whitehill, J., P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise." In: *NIPS*. Ed. by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta. Curran Associates, Inc., pp. 2035–2043. ISBN: 9781615679119.

Wulczyn, E., N. Thain, and L. Dixon (2017). *Ex Machina: Personal Attacks Seen at Scale*. arXiv: 1610.08914 [cs.CL].

Yan, Y., R. Rosales, G. Fung, M. W. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy (2010). "Modeling annotator expertise: Learning when everybody knows a bit of something." In: *AISTATS*. Ed. by Y. W. Teh and D. M. Titterington. Vol. 9. JMLR Proceedings. JMLR.org, pp. 932–939.

Yi, K. and J. Wu (2019). "Probabilistic end-to-end noise correction for learning with noisy labels." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7017–7025.

Zeng, J., S. Shan, and X. Chen (2018). "Facial Expression Recognition with Inconsistently Annotated Datasets." In: *ECCV (13)*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Vol. 11217. Lecture Notes in Computer Science. Springer, pp. 227–243. ISBN: 978-3-030-01261-8.

Zhang, Z. and M. R. Sabuncu (2018). "Generalized cross entropy loss for training deep neural networks with noisy labels." In: *arXiv preprint arXiv:1805.07836*.

Zhuang, H., A. Parameswaran, D. Roth, and J. Han (2015). "Debiasing crowdsourced batches." In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1593–1602.