UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Andreas Pung

Weakly-Supervised Text Classification for Estonian Sentiment Analysis

Master's Thesis (30 ECTS)

Supervisor: Kairit Sirts, PhD

Tartu 2022

Weakly-Supervised Text Classification for Estonian Sentiment Analysis

Abstract:
Text Classification is one of the most fundamental tasks in Natural Language Processing. Hand-labelling texts is costly and might need specialised domain knowledge – this is where unsupervised and weakly-supervised approaches could be useful. In this Master's Thesis, the weakly-supervised text classification paradigm is used to classify the sentiment of Estonian texts. In this paradigm, the weak labels are created using labelling functions (Ratner et al., 2016). The aim of this thesis is to assess the applicability of weakly-supervised models trained with around 40× larger dataset in contrast to hand-labelling a smaller amount of texts to train a fully-supervised classifier. The compared models are fully and weakly-supervised `BERT` (Devlin et al., 2019); weakly-supervised `COSINE` (Yu et al., 2021) and `WeaSEL` (Cachay et al., 2021). Human evaluation is performed on texts where the models disagreed the most. As a result, we find that the fully-supervised models have the best performance. The best-performing weakly-supervised model trained on the larger dataset had an average classification accuracy of 7.29% worse (7.05% worse weighted F1-score) than the fully-supervised `BERT` model. The lower performance of weakly-supervised models might be caused by the low quality of labelling functions – developing them further might lead to better results.

Keywords:
Text classification, weakly-supervised text classification, weak supervision, labelling functions, unsupervised text classification, natural language processing, sentiment analysis, Estonian dataset.

CERCS: P176 Artificial intelligence

Nõrgalt juhendatud teksti klassifitseerimine eestikeelse meelsusanalüüsi jaoks

Lühikokkuvõte:
Teksti klassifitseerimine on üks kõige fundamentaalsem ülesanne loomuliku keele töötluses. Käsitsi tekstide märgendamine on kulukas ja võib vajada spetsialiseeritud domeeniteadmisi – sellisel juhul võivad juhendamata ja nõrgalt juhendatud lähenemised olla kasulikud. Käesolevas magistritöös klassifitseeritakse eestikeelsete tekstide meelsust nõrga juhendamise paradigmaga. Selles paradigmas luuakse nõrgad märgendid märgendusfunktsioonidega (Ratner et al., 2016). Käesoleva töö eesmärk on hinnata nõrgalt juhendatud umbes 40× suurema andmestikuga treenitud mudelite rakendatavust, võrreldes väiksema arvu tekstide käsitsi märgendamisega, et treenida täielikult juhendatud klassifitseerija. Võrreldud mudelid on täielikult ja nõrgalt juhendatud `BERT` (Devlin et al., 2019); nõrgalt juhendatud `COSINE` (Yu et al., 2021) ja `WeaSEL` (Cachay et al., 2021). Inimhindamine viidi läbi tekstidel, kus mudelite ennustused olid kõige vastukäivamad. Leitakse, et täielikult juhendatud mudelid töötavad kõige paremini. Kõige paremini toimival suuremal andmestikul treenitud nõrgalt juhendatud mudelil oli keskmine klassifitseerimistäpsus 7.29% halvem (7.05% halvem F1-skoor) kui täielikult juhendatud `BERT`i mudelil. Nõrgalt juhendatud mudelite kehvem tulemus võib tuleneda märgendusfunktsioonide madalast kvaliteedist – nende edasiarendamine võib anda paremaid tulemusi.

Võtmesõnad:
Teksti klassifitseerimine, nõrgalt juhendatud teksti klassifitseerimine, nõrk juhendamine, märgendusfunktsioonid, juhendamata teksti klassifitseerimine, loomuliku keele töötlus, meelsusanalüüs, eestikeelne andmestik.

CERCS: P176 Tehisintellekt

Contents

1 Introduction

Text Classification is one of the most fundamental tasks in Natural Language Processing (NLP). Text classification aims to classify a text into one class from a predefined set of classes. Text classification has been chiefly performed using various supervised text classification algorithms, which have been researched extensively. A fully-supervised approach requires a labelled dataset, but gathering large amounts of training data might be troublesome because of the need for many experts with domain knowledge. Hand-labelling takes a significant amount of time and is very costly and mundane. Various unsupervised methods can be used to mitigate the problems of labelling texts. These approaches have been studied to a much smaller extent when compared to supervised models.

In this Master's Thesis, the main emphasis is on a perspective text classification approach that does not require a labelled dataset called the weakly-supervised text classification paradigm. Weak supervision aims to generate weak labels using some heuristic hand-programmed labelling functions (Ratner et al., 2016) and then use these weak labels to train a discriminative end model that might have some similarities to a fully-supervised classification model but takes into account the fact that the weak labels introduce additional noise and that they might be incorrect. The advantage of using weakly-supervised text classification is the ability to use much more training data. When comparing fully-supervised and weakly-supervised models, there is a tradeoff between model performance and the work needed to either hand-label texts or develop labelling functions.

An example of a language with a lack of annotated training data is Estonian. More concretely, one example of a text classification task is Estonian sentiment analysis, where there is only one major public labelled dataset (Pajupuu et al., 2016). This work is novel in the sense that, to the best of our knowledge, nobody has written about applying weakly-supervised text classification models to a textual dataset in Estonian before. The annotated Estonian sentiment analysis dataset (Pajupuu et al., 2016) is relatively small, and therefore supervised models tend to overfit the dataset. The aim of this Master's Thesis is to assess the applicability of weakly-supervised methods in Estonian sentiment analysis as an alternative to annotating more data.

The Estonian Valence Corpus sentiment analysis dataset (Pajupuu et al., 2016) is used in conjunction with the Postimees (Kaalep et al., 2010; Muischnek, 2011) dataset to create around 40× larger unannotated training dataset (the Combined dataset), which can be used to train weakly-supervised text classifiers. The state-of-the-art weakly-supervised text classification transformer-based model `COSINE` (Yu et al., 2021) and `WeaSEL` (Cachay et al., 2021) are applied. These models are implemented in the weak supervision benchmark WRENCH (Zhang et al., 2021)

that is also used in this work. To compare the models, the regular BERT (Devlin et al., 2019) (EstBERT (Tanvir et al., 2021) implementation) model is also trained and evaluated as a baseline model. Four models are trained on an annotated Valence dataset for comparison purposes, and three weakly-supervised models are trained on the larger Combined dataset. All seven models are first evaluated extensively on the labelled Valence dataset to get a good baseline understanding of their performance. After that, the Postimees test set texts are classified. The top-100 most difficult to classify texts with the most disagreements are selected for further analysis. Human labelling is performed to create ground truth labels using majority voting. The models are analysed further to assess whether the many times larger training dataset helps to gain performance.

The fully-supervised EstBERT model had the best performance in all experiments. On the Combined dataset, the COSINE model had the best average test classification accuracy of 67.15% (66.77% weighted F1-score). Compared to the COSINE model trained on the Valence dataset, the accuracy was, on average, 2.93% (3.35% higher weighted F1-score) higher, so a larger training dataset did help the model get better performance on the annotated Valence test set. For the Postimees test subset for which the labels were set by a majority vote by human labellers without ambiguous texts, the best-performing model, COSINE trained on the Combined dataset had an average classification accuracy of 11.43% (8.79% worse weighted F1-score) worse than the fully-supervised EstBERT model. The lower performance of weakly-supervised models might be caused by the low quality of labelling functions – developing them further might lead to better results.

Firstly, necessary background knowledge will be provided in section 2. Section 3 will give an overview of different text classification methods. Methodology and the setup of the experiment will be explained in section 4. Experiment results will be provided in section 5. Finally, the work will be concluded in section 6.

I would like to thank my supervisor Kairit Sirts for providing invaluable feedback throughout the writing of this thesis. Discussing the topics of the thesis has been one of the most enriching experiences during my Master's studies. I would also like to thank the annotators for labelling the texts.

2 Background Knowledge

In this section, some definitions and general knowledge will be provided. The natural language processing problem of text classification will be introduced along with the weak supervision paradigm, labelling functions and the task of sentiment analysis. To understand the models used in the experiment of this thesis, a concise overview will be given of transformer neural networks and the `BERT` model. Finally, a summary will be given of the `MeTaL` label model and the `COSINE` and `WeaSEL` models.

2.1 Text Classification

Text Classification can be formally defined as the following (Meng et al., 2018). Given a text collection $\mathcal{D} = \{D_1, \ldots, D_n\}$ and m target classes $\mathcal{C} = \{C_1, \ldots, C_m\}$, text classification aims to assign a class label $C_j \in \mathcal{C}$ to each document $D_i \in \mathcal{D}$.

There can be four different scopes of text classification (Kowsari et al., 2019):

1. Document-level – a whole document is classified.
2. Paragraph-level – a single paragraph is classified.
3. Sentence-level – a single sentence (a portion of a paragraph) is classified.
4. Sub-sentence level – sub-expressions within a sentence (a portion of a sentence) are classified.

There are different approaches to text classification in terms of the strength of the supervision – supervised, weakly-supervised, and unsupervised methods.

Supervised Text Classification is defined as the following (Jurafsky and Martin, 2021). There is a training set of N documents $(d_1, \ldots d_N)$ and M predefined classes that have each been hand-labelled with a class $C = \{c_1, \ldots, c_M\}$ as pairs $(d_1, c_1), \ldots, (d_N, c_N)$. The goal is to learn a classifier that is capable of mapping from a new document d to its correct class $c \in C$.

As the work mostly pays attention to methods that do not require an annotated dataset, unsupervised and weakly-supervised methods will be explained in the following subsection.

2.2 Unsupervised & Weakly-Supervised Text Classification

Unsupervised Text Classification is defined as performing text classification using no annotated dataset. It is not straightforward to classify weak supervision into either a supervised or an unsupervised method. Weakly-supervised text

classification could be seen as one paradigm of unsupervised text classification, in some sense.

Weak Supervision (WS) is defined as the following in the WRENCH paper (Zhang et al., 2021), originally by Ratner et al. (2016). We are given a dataset containing n data points $\mathcal{X} = [X_1, X_2, \ldots, X_n]$ with the i-th data point denoted by $X_i \in \mathcal{X}$. For each X_i, there is an unobserved true (gold, ground truth) label denoted by $Y_i \in \mathcal{Y}$. Let m be the number of WS sources $\{S_j\}_{j=1}^{m}$, each assigning a weak label $\lambda_j \in \mathcal{Y}$ to X_i to vote on its respective golden Y_i or abstaining ($\lambda_j = -1$).

Weakly-Supervised Text Classification. A label matrix $L \in \mathbb{R}^{n \times m}$ is obtained via applying m labelling functions to the dataset $\mathcal{X} = [X_1, X_2, \ldots, X_n]$. The aim of the classification is to build an end model $f_w : \mathcal{X} \to \mathcal{Y}$ to infer the predicted labels $\hat{Y}$ for each $X_i \in \mathbf{X}$ using L. This is the definition by Zhang et al. (2021), that is used in this thesis.

Label Model is used to combine the weak label matrix $L \in \mathbb{R}^{n \times m}$ into either probabilistic soft labels or one-hot hard labels $L \in \mathbb{R}^{n}$ (Zhang et al., 2021).

End Model is a discriminative model that is trained using the weak labels as input. The model can generalise beyond the weak labels for downstream tasks (Zhang et al., 2021).

Joint Model is a coupling of a label model and an end model in an end-to-end manner (Zhang et al., 2021).

Weakly-Supervised Text Classification (alternative definition) has been defined quite differently in various other works. For example, in Meng et al. (2018) weak supervision is defined (as an extension to the text classification definition) as the following. Let there be a text collection $\mathcal{D} = \{D_1, \ldots, D_n\}$ and target classes $\mathcal{C} = \{C_1, \ldots, C_m\}$. Weak supervision for text classification could come from one of the following sources:

- *label surface names*: $\mathcal{L} = \{L_j\}|_{j=1}^{m}$, where L_j is the surface name for class C_j. This is also called as dataless text classification in some other works,
- *class-related keywords*: $\mathcal{S} = \{S_j\}|_{j=1}^{m}$, where $S_j = \{w_{j,1}, \ldots, w_{j,k}\}$ represents a set of k keywords in class C_j. Some authors call this also dataless text classification, and
- *labelled documents*: $\mathcal{D}^L = \{\mathcal{D}_j^L\}|_{j=1}^{m}$, where $\mathcal{D}_j^L = \{D_{j,1}, \ldots, D_{j,l}\}$ denotes a set of l labelled documents in class C_j. This is also called as zero-shot text classification in some works.

Given a text collection $\mathcal{D}$, target classes $\mathcal{C}$, and weak supervision from either $\mathcal{L}$, $\mathcal{S}$ or $\mathcal{D}^L$, the weakly-supervised text classification task aims to assign a label $C_j \in \mathcal{C}$

to each $D_i \in \mathcal{D}$. This shows that the terminology is fuzzy and that there is no unanimous agreement among researchers.

Zero-Shot Text Classification is defined as the following by Yin et al. (2019) (the *restrictive* definition). Given labelled instances belonging to a set of seen classes S, zero-shot text classification aims at learning a classifier $f : X \to Y$, where $Y = S \cup U$. Set U is a set of unseen classes and belongs to the same aspect as S.

Dataless Text Classification is defined as a text classification task that does not need annotated training data and where the source of supervision comes from label names (Chang et al., 2008) or label names with seed words (Li et al., 2018b).

Semi-Supervised Text Classification is a text classification paradigm where a classifier is learned using a limited labelled and a large unlabelled text set (Li et al., 2021).

2.3 Labelling Functions

The basis of weak supervision is formed by using different labelling functions (LFs), as they were initially introduced in the data programming paradigm by Ratner et al. (2016).

Informally, LFs are some heuristic functions that, in the case of a text, check whether it contains some predefined class-indicative words, regex patterns, or any other heuristics that might predict the class of the text as well as possible. If some LF does not detect a sentiment or the heuristic signal is too weak, it can also defer from making a decision and output `ABSTAIN` (Ratner et al., 2018).

Labelling Function (LF) $\lambda_i : \mathcal{X} \to \{-1, 0, 1\}$ is a user-defined function that encodes some domain heuristic, which provides a (non-zero) label for some subset of the objects (Ratner et al., 2016). The definition is provided for a binary classification task. In this work $\lambda_i : \mathcal{X} \to \{-1, 0, 1, 2\}$, where -1 denotes abstaining, 0 the negative label, 1 the positive label, and 2 the neutral label.

Labelling functions are created using different weak supervision sources. According to Ratner et al. (2017), these sources include distant supervision where the records of an external knowledge base are heuristically aligned with data points to produce noisy labels. They add that crowdsourced labels, rules, and heuristics could be used for generating LFs. They conclude by saying that the labels should be combined from many weak supervision sources to increase the accuracy and coverage of the training set.

The `Snorkel` (Ratner et al., 2017) library calculates multiple metrics related to the LFs[1], which can be used to evaluate them:

[1] `https://snorkel.readthedocs.io/en/v0.9.7/packages/_autosummary/labeling/snorkel.labeling.LFAnalysis.html`

- Polarity – a list of unique output labels.
- Coverage – a fraction of texts each LF labels.
- Overlaps – a fraction of texts each LF labels that are labelled by another LF.
- Conflicts – a fraction of examples each LF labels and that are labelled differently by another LF.
- Correct – a number of texts classified correctly by an LF.
- Incorrect – a number of texts classified incorrectly by an LF.
- Empirical Accuracy – an empirical accuracy against a set of ground truth labels for each LF.

In the following subsection, the task of sentiment analysis will be explained along with some related work regarding Estonian sentiment analysis.

2.4 Sentiment Analysis

Various textual aspects can be classified. According to Jurafsky and Martin (2021), a common one is called sentiment analysis, the extraction of sentiment – the orientation that a writer expresses toward some object. In this work, the experiments are done on Estonian sentiment analysis. Other aspects for which a text could be classified include spam detection, language identification, authorship attribution, and (library) topic classification.

The following text is based on a survey paper by Zhang et al. (2018). **Sentiment Analysis**, or opinion mining, is a computational study of opinions, sentiments, emotions, appraisals, and attitudes towards products, services, organisations, individuals, issues, events, topics, and their attributes. There are three levels of sentiment analysis granularity – document-level, sentence-level, and aspect-level. This work focuses on document-level (paragraph-level) sentiment analysis. Document-level sentiment classification categorises an opinionated document as expressing an overall positive or negative opinion. The neutral class is also included in this work because not every text might be opinionated in the datasets used.

Estonian sentiment analysis has not been researched very extensively. Pajupuu et al. (2016) proposed a lexicon-based approach and also trained machine learning models. Ojamaa et al. (2015) describe ongoing work related to estimating the speaker's attitude based on their opinions expressed by utterances. Uustalu (2019) explored Estonian entity-level sentiment analysis.

2.5 Transformer Neural Network

The Transformer is a sequence-to-sequence artificial neural network architecture, which relies entirely on an attention mechanism to draw global dependencies between input and output (Vaswani et al., 2017).

This subsection is based on Jurafsky and Martin (2021). Transformers map sequences of input vectors $(x_1, \ldots, x_n)$ to sequences of output vectors $(y_1, \ldots, y_n)$ of the same length. The network consists of transformer layers which consist of linear layers and self-attention layers. They further explain that self-attention allows a network to directly extract and use information from arbitrarily large contexts without passing it through intermediate recurrent connections. The self-attention layer is one of the most important components of the transformer block.

A Transformer consists of three input embeddings which contain different roles during the attention process:

- Query – a current focus of attention compared to all of the other preceding inputs (weight matrix W_Q introduced).
- Key – a preceding input compared to the current focus of attention (weight matrix W_K, dimensions d_k).
- Value – to compute the output for the current focus of attention (weight matrix W_V).

In a general form, self-attention could be described as the following. Let there be an input matrix X of the word embeddings of the input tokens. The input matrix X is multiplied by the key, query and value matrices, $Q = XW_Q, K = XW_K, V = XW_V$. The self-attention layer values can be computed with the following computation: $SelfAttention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$. In addition to self-attention layers, transformers consist of feedforward layers, residual connections and normalisation layers. A schematic overview of the encoder block of the transformer network can be seen in Figure 1.

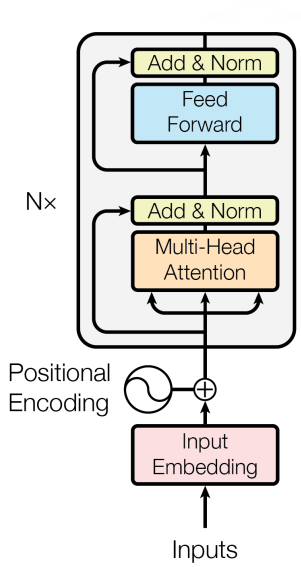


Figure 1. A schematic overview of the encoder block of the Transformer neural network. Figure adapted from Vaswani et al. (2017).

Jurafsky and Martin (2021) explain that Transformers use multi-head self-attention layers. These are sets of heads in the multi-head self-attention layers' parallel layers at the same depth in a model, each with its own set of parameters. Each head has its learnable key, query, and value matrices.

To model positions of tokens in a sequence, positional embeddings (encodings) are used. These embeddings can be learned or be set to a fixed function, for example, the sine or cosine functions (Vaswani et al., 2017).

2.6 Bidirectional Encoder Representations from Transformers (BERT)

One of the most useful applications of the Transformer neural networks in NLP is the pre-trained **B**idirectional **E**ncoder **R**epresentations from **T**ransformers model, better known as the BERT model (Devlin et al., 2019).

The authors (Devlin et al., 2019) explain that the key contribution is that BERT is designed to pre-train deep bidirectional representations from unlabelled texts by jointly conditioning on the left and right contexts. They add that pre-trained BERT models can be fine-tuned with just one additional output layer to create state-of-the-art models. An overview of the pre-training and fine-tuning steps can be seen in Figure 2.

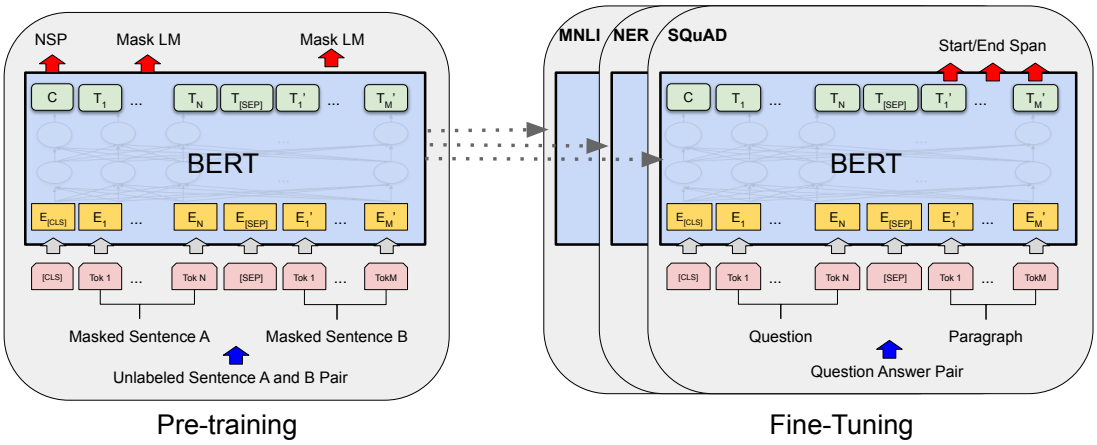


Figure 2. Two main steps in the BERT framework – pre-training and fine-tuning (Devlin et al., 2019).

The multi-layer bidirectional Transformer encoder (Vaswani et al., 2017) was used as the model architecture for the BERT model. For example, the $\mathbf{BERT}_{BASE}$ model had 12 Transformer layers, 768-dimension hidden layers, and 12 self-attention heads resulting in 110 million parameters.

WordPiece (Wu et al., 2016) embeddings were used with a 30 000 token vocabulary. A special [CLS] token is in place at the beginning of every sequence. The final hidden state corresponding to the [CLS] token is used as the sequence representation for various classification tasks. A special [SEP] token is used to differentiate different sentence pairs. A learned embedding is added to every token indicating whether it belongs to the first or the following sentence. A token's input representation is constructed by summing the corresponding token, segment, and position embeddings.

There are two tasks on which the BERT model was trained on (Devlin et al., 2019):

- Masked Language Modelling – some percentage of the input tokens are masked at random, and the objective of the model is to predict the masked tokens. This is done by feeding the final hidden vectors corresponding to the mask tokens into an output softmax over the model's vocabulary.

- Next Sentence Prediction – the first and its following sentence for each pre-training example is picked to do binary classification, whether the following sentence comes after the first one in the training corpus. Half of the time, the following sentence is the actual next sentence that follows the first sentence, and the other half of it is a random sentence from the corpus.

The original BERT model was trained on BooksCorpus (Zhu et al., 2015), which consists of 800 million words and the English Wikipedia, which consisted of 2500 million words at that time.

Many different autoencoding based language models have been trained. A large part of them have been inspired from the original BERT (Devlin et al., 2019) model. Examples of such models include the RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019) models. Many language-specific BERT implementations have been pre-trained. The main BERT implementation used in this thesis is the EstBERT (Tanvir et al., 2021) model. EstBERT was trained on the Estonian National Corpus 2017 (Kallas and Koppel, 2018) dataset – after preprocessing, 1154 million words remained in the corpus.

2.7 Overview of the Models Used

In this section, an overview will be given explaining the models used in the experimental section of this work. The model summaries will be referenced from their original papers.

2.7.1 MeTaL Framework

The MeTaL framework was used in this work as a label model. This section will be based on the original article by Ratner et al. (2018). In the experiments of this work, the implementation of the Snorkel (Ratner et al., 2017) library's LabelModel[2] class will be used.

A high-level schematic overview of the MeTaL framework can be seen in Figure 3. In its original setting, the first step in the framework is inputting a task graph G_{task}, which defines the relationships between task labels $Y_1, \ldots, Y_t$; a set of unlabeled data points X; a set of multi-task weak supervision sources s_i, which each output a vector $\boldsymbol{\lambda}_i$ of task labels for X; and the dependency structure between these sources, G_{source}. A label model to learn the accuracies of the sources is trained. A vector of probabilistic training labels $\tilde{\boldsymbol{Y}}$ is outputted for training the end model.

2.7.1.1 Problem Definition

More concretely, the problem is defined as the following (Ratner et al., 2018). Let $X \in \mathcal{X}$ be a data point and $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_t]^T$ be a vector of categorical *task labels*, $Y_i \in \{1, \ldots, k_i\}$, corresponding to t tasks, where $(X, \mathbf{Y})$ is drawn (independent and identically distributed) from a distribution $\mathcal{D}$. The user provides a specification of how these tasks relate to each other, and this schema is denoted as

[2] https://snorkel.readthedocs.io/en/v0.9.3/packages/_autosummary/labeling/snorkel.labeling.LabelModel.html#snorkel.labeling.LabelModel

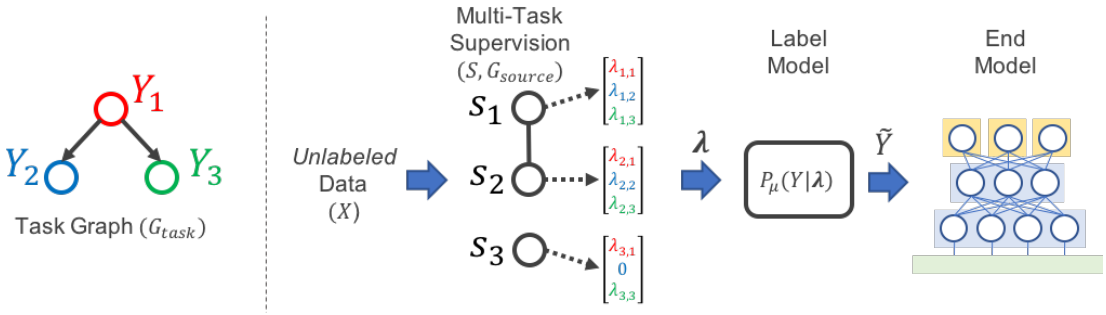


Figure 3. A schematic overview of the MeTaL framework (Ratner et al., 2018).

the *task structure* G_{task}. The task structure expresses logical relationships between tasks, defining a *feasible set* of label vectors $\mathcal{Y}$, such that $\mathbf{Y} \in \mathcal{Y}$. In MeTaL, instead of observing the true label $\mathbf{Y}$, there is access to m *multi-task weak supervision* sources $s_i \in S$. They emit label vectors $\boldsymbol{\lambda}_i$ that contain labels for some subset of the t tasks. Let 0 denote an abstaining label, and let the *coverage set* $\tau_i \subseteq \{1, \ldots, t\}$ be the fixed set of tasks for which the ith source emits non-zero labels, such that $\boldsymbol{\lambda}_i \in \mathcal{Y}_{\tau_i}$.

The overall goal of the task is to apply the set of weak supervision sources $S = \{s_1, \ldots, s_m\}$ to an unlabelled dataset $\mathcal{X}_U$ consisting of n data points. After that, the resulting weakly-labelled training set can be used to supervise an *end model* $f_w : \mathcal{X} \mapsto \mathcal{Y}$. The weakly-labelled training set might contain overlapping and conflicting labels because the sources may have unknown accuracies and correlations. The authors propose to learn a *label model* $P_\mu(\mathbf{Y}|\boldsymbol{\lambda})$, parameterized by a vector of source correlations and accuracies μ. For each data point X, the label model takes as input the noisy labels $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_m\}$ and outputs a single probabilistic label vector $\tilde{\mathbf{Y}}$.

Learning a Label Model. The authors (Ratner et al., 2018) introduce some statistics over the random variables in G_{source}. Variable $\mathcal{C}$ is defined as the set of cliques in G_{source}, and an indicator random variable is defined for the event of a clique $C \in \mathcal{C}$ taking on a set of values y_C:

$$\psi(C, y_C) = \mathbb{1}\left\{\cap_{i \in C} V_i = (y_C)_i\right\},$$

where $(y_C)_i \in \mathcal{Y}_{\tau_i}$. Vector $\psi(C) \in \{0,1\}^{\prod_{i \in C}(|\mathcal{Y}_{\tau_i}|-1)}$ is defined as the vector of indicator random variables for all combinations of all but one of the labels emitted by each variable in clique C, this defines a minimal set of statistics and defines $\psi(\mathbf{C})$ accordingly for any set of cliques $\mathbf{C} \subseteq \mathcal{C}$. Vector $\mu = \mathbb{E}\left[\psi(\mathcal{C})\right]$ is the vector of sufficient statistics for the label model.

2.7.1.2 Computational Approach

The authors (Ratner et al., 2018) begin by explaining that the critical problem is that $\mathbf{Y}$ cannot be observed. This problem is overcome by analysing the covariance matrix of an observable subset of the cliques in G_{source}. That leads to a matrix completion-style approach for recovering μ.

The authors start by considering two disjoint subsets of $\mathcal{C}$: the set of observable cliques, $O \subseteq \mathcal{C}$, these are the cliques not containing $\mathbf{Y}$, and the separator set cliques of the junction tree, $\mathcal{S} \subseteq \mathcal{C}$. The covariance matrix of the indicator variables for $O \cup \mathcal{S}$, $\mathbf{Cov}\left[\psi(O \cup \mathcal{S})\right]$, could be written in block form:

$$\mathbf{Cov}\left[\psi(O \cup \mathcal{S})\right] \equiv \Sigma = \begin{bmatrix} \Sigma_O & \Sigma_{O\mathcal{S}} \\ \Sigma_{O\mathcal{S}}^T & \Sigma_{\mathcal{S}} \end{bmatrix} \tag{1}$$

and similarly define its inverse:

$$K = \Sigma^{-1} = \begin{bmatrix} K_O & K_{O\mathcal{S}} \\ K_{O\mathcal{S}}^T & K_{\mathcal{S}} \end{bmatrix} \tag{2}$$

In the previous formula, $\Sigma_O = \mathbf{Cov}\left[\psi(O)\right] \in \mathbb{R}^{d_O \times d_O}$ is the observable block of Σ, where $d_O = \sum_{C \in O} \prod_{i \in C} (|\mathcal{Y}_{\tau_i}| - 1)$. Formula $\Sigma_{O\mathcal{S}} = \mathbf{Cov}\left[\psi(O), \psi(\mathcal{S})\right]$ is the unobserved block, which is a function of μ. The sum $\Sigma_{\mathcal{S}} = \mathbf{Cov}\left[\psi(\mathcal{S})\right] = \mathbf{Cov}\left[\psi(\mathbf{Y})\right]$ is a function of the class balance $P(\mathbf{Y})$.

The authors apply the block matrix inversion lemma:

$$K_O = \Sigma_O^{-1} + c\Sigma_O^{-1}\Sigma_{O\mathcal{S}}\Sigma_{O\mathcal{S}}^T\Sigma_O^{-1}, \tag{3}$$

where $c = \left(\Sigma_{\mathcal{S}} - \Sigma_{O\mathcal{S}}^T\Sigma_O^{-1}\Sigma_{O\mathcal{S}}\right)^{-1} \in \mathbb{R}^+$. Let $z = \sqrt{c}\Sigma_O^{-1}\Sigma_{O\mathcal{S}}$; then Equation (3) could be expressed as:

$$K_O = \Sigma_O^{-1} + zz^T \tag{4}$$

Finally, let Ω be the set of indices (i, j) where $(K_O)_{i,j} = 0$, determined by G_{source}. This yields a system of equations:

$$0 = (\Sigma_O^{-1})_{i,j} + \left(zz^T\right)_{i,j} \text{ for } (i, j) \in \Omega, \tag{5}$$

which is now a matrix completion problem. The authors define $||A||_\Omega$ as the Frobenius norm of A with entries not in Ω set to zero. Equation (5) could be rewritten as $\left|\left|\Sigma_O^{-1} + zz^T\right|\right|_\Omega = 0$. This equation can be solved to estimate z, and thereby recover $\Sigma_{O\mathcal{S}}$, from which it is possible to recover the label model parameters μ algebraically directly.

2.7.2 Contrastive Self-Training for Fine-Tuning Pre-Trained Language Model (COSINE)

Contrastive **S**elf-Training for F**ine**-Tuning Pre-Trained Language Model (COSINE) is a model introduced by Yu et al. (2021). This section will be referenced from their original article.

COSINE is a contrastive self-training framework to enable fine-tuning Language Models (LMs) with weak supervision. The essence of COSINE is to use contrastive regularisation and confidence-based reweighting. A high-level schematic overview of the COSINE model can be seen in Figure 4.

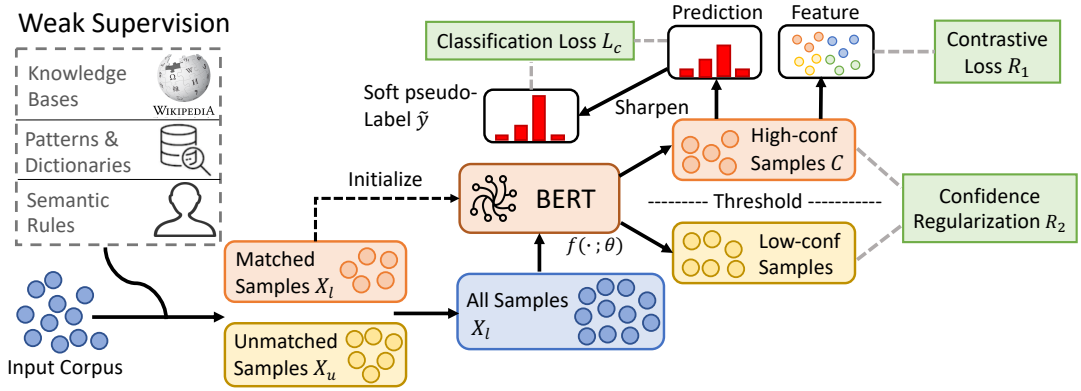


Figure 4. High-level overview of the COSINE framework (Yu et al., 2021).

The authors (Yu et al., 2021) explain that in COSINE, there is a classifier $f = g \circ \text{BERT}$ that consists of two parts – BERT is any pre-trained language model that outputs hidden representations of input samples, the function g is a task-specific classification head that outputs a C-dimensional vector, where each dimension corresponds to the prediction confidence of a specific class. In this thesis, EstBERT (Tanvir et al., 2021) is used as the realisation of BERT. Firstly, the LM is initialised in COSINE with weak labels. This step is necessary because the semantic and syntactic knowledge of the pre-trained LM are transferred to the model. The main idea of COSINE is to use contrastive self-training to suppress label noise propagation.

2.7.2.1 Training Procedure

In this section, a brief overview will be given of how COSINE is trained by referencing the original source (Yu et al., 2021).

Initialization with weakly-labelled data. Function $f(\cdot;\theta)$ is fine-tuned with weakly-labelled data $\mathcal{X}_l$ by solving the optimisation problem

$$\min_{\theta} \frac{1}{|\mathcal{X}_l|} \sum_{(x_i,y_i)\in\mathcal{X}_l} \mathrm{CE}\left(f(x_i;\theta), y_i\right), \tag{6}$$

where $\mathrm{CE}(\cdot;\cdot)$ is defined as the cross-entropy loss. The authors also use early stopping (Dodge et al., 2020) to prevent the model from overfitting to the label noise. Early stopping might cause underfitting, and this is resolved by contrastive self-training.

Contrastive self-training with all data. Contrastive self-training leverages all available data, both labelled and unlabelled. They are used for fine-tuning and to reduce the error propagation of wrongly labelled data. Pseudo-labels are generated for the unlabelled data and incorporated into the training set. Contrastive representation learning and confidence-based sample reweighting and regularisation are introduced to reduce error propagation. The pseudo-labels $\tilde{\boldsymbol{y}}$ and the model are updated iteratively.

Updating $\tilde{y}$ with the current θ. Soft pseudo-labels $\tilde{\boldsymbol{y}} \in \mathbb{R}^C$ for each sample x in a batch $\mathcal{B}$ are generated based on the current model

$$\tilde{\boldsymbol{y}}_j = \frac{[f(x;\theta)]_j^2/f_j}{\sum_{j'\in\mathcal{Y}}[f(x;\theta)]_{j'}^2/f_{j'}}, \tag{7}$$

where $f_j = \sum_{x'\in\mathcal{B}}[f(x';\theta)]_j^2$ is defined as the sum over soft frequencies of class j.

Updating θ with the current $\tilde{y}$. The model parameters θ are updated by minimising

$$\mathcal{L}(\theta;\tilde{\boldsymbol{y}}) = \mathcal{L}_c(\theta;\tilde{\boldsymbol{y}}) + \mathcal{R}_1(\theta;\tilde{\boldsymbol{y}}) + \lambda\mathcal{R}_2(\theta), \tag{8}$$

where $\mathcal{L}_c$ is the classification loss, $\mathcal{R}_1(\theta;\tilde{\boldsymbol{y}})$ is the contrastive regulariser, $\mathcal{R}_2(\theta)$ is the confidence regulariser, and λ is the hyper-parameter for the regularisation.

2.7.2.2 Contrastive Learning

The authors of `COSINE` (Yu et al., 2021) explain that a key ingredient of the contrastive self-training method is to learn representations that encourage data within the same class to have similar representations and keep data in different

classes separated. High-confidence samples $\mathcal{C}$ are selected first from $\mathcal{X}$. Then for each pair $x_i, x_j \in \mathcal{C}$, their similarity is defined as

$$W_{ij} = \begin{cases} 1, & if \underset{k \in \mathcal{Y}}{\operatorname{argmax}} [\tilde{\boldsymbol{y}}_i]_k = \underset{k \in \mathcal{Y}}{\operatorname{argmax}} [\tilde{\boldsymbol{y}}_j]_k \\ 0, & otherwise \end{cases}, \tag{9}$$

where $\tilde{\boldsymbol{y}}_i$, $\tilde{\boldsymbol{y}}_j$ are the soft pseudo-labels for x_i, x_j, respectively. For each $x \in \mathcal{C}$, its representation $\boldsymbol{v} = \text{BERT}(x) \in \mathbb{R}^d$ is calculated, then the contrastive regulariser is defined as

$$\mathcal{R}_1(\theta; \tilde{\boldsymbol{y}}) = \sum_{(x_i, x_j) \in \mathcal{C} \times \mathcal{C}} \ell(\boldsymbol{v}_i, \boldsymbol{v}_j, W_{ij}), \tag{10}$$

where

$$\ell = W_{ij} d_{ij}^2 + (1 - W_{ij})[\max(0, \gamma - d_{ij})]^2. \tag{11}$$

In the previous formula, $\ell(\cdot, \cdot, \cdot)$ is the contrastive loss, d_{ij} is the distance between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, and γ is a pre-defined margin.

2.7.2.3 Confidence-Based Sample Reweighting and Regularisation

The authors (Yu et al., 2021) explain that while contrastive representations yield better decision boundaries, they require samples with high-quality pseudo-labels.

Sample reweighting. Samples with high prediction confidence are more likely to be classified correctly than those with low confidence. Label noise propagation is further reduced by a confidence-based sample reweighting scheme. For each sample x with the soft pseudo-label $\tilde{\boldsymbol{y}}$, x is assigned with a weight $\omega(x)$ defined by

$$\omega = 1 - \frac{H(\tilde{\boldsymbol{y}})}{\log(C)}, \; H(\tilde{\boldsymbol{y}}) = -\sum_{i=1}^{C} \tilde{\boldsymbol{y}}_i \log \tilde{\boldsymbol{y}}_i, \tag{12}$$

where $0 \leq H(\tilde{\boldsymbol{y}}) \leq \log(C)$ is the entropy of $\tilde{\boldsymbol{y}}$. A pre-defined threshold ξ is used to select high confidence samples $\mathcal{C}$ from each batch $\mathcal{B}$ as

$$\mathcal{C} = \{x \in \mathcal{B} \mid \omega(x) \geq \xi\}. \tag{13}$$

The loss function is defined as

$$\mathcal{L}_c(\theta, \tilde{\boldsymbol{y}}) = \frac{1}{|\mathcal{C}|} \sum_{x \in \mathcal{C}} \omega(x) \mathcal{D}_{\text{KL}} \left(\tilde{\boldsymbol{y}} \| f(x; \theta)\right), \tag{14}$$

where $\mathcal{D}_{\text{KL}}(P \| Q)$ is the Kullback–Leibler divergence.

Confidence regularisation. The authors propose a confidence-based regulariser that encourages smoothness over predictions, and it is defined as

$$\mathcal{R}_2(\theta) = \frac{1}{|\mathcal{C}|} \sum_{x \in \mathcal{C}} \mathcal{D}_{\mathrm{KL}} \left(\mathbf{u} \| f(x; \theta) \right), \tag{15}$$

where $\mathbf{u}_i = 1/C$ for $i = 1, 2, \ldots, C$. This term is needed to prevent over-confident predictions and leads to better generalisation (Pereyra et al., 2017).

2.7.3 Weakly-Supervised End-To-End Learner Model (`WeaSEL`)

In this section, the **Wea**kly **S**upervised **E**nd-To-End **L**earner model (`WeaSEL`) will be introduced. The model was proposed by Cachay et al. (2021), and this section will be referenced from the original article. The authors of the `WeaSEL` paper begin by criticising that the approaches that require two modelling steps – learning a probabilistic latent variable model and training a separate downstream model may not work as well as they could. A high-level overview of the `WeaSEL` model can be seen in Figure 5.

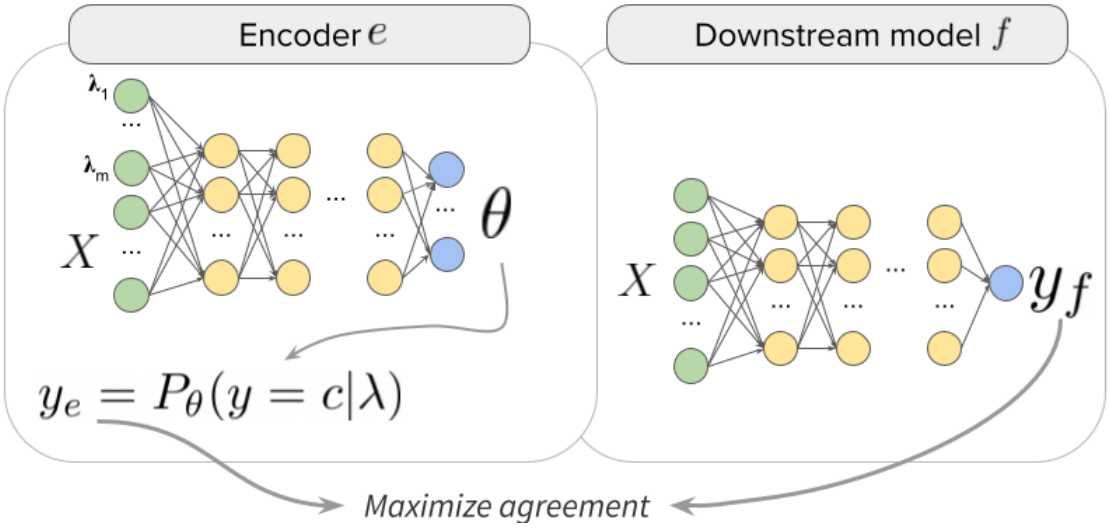


Figure 5. High-level overview of the `WeaSEL` model (Cachay et al., 2021).

2.7.3.1 Problem Setup

The problem is defined by the authors (Cachay et al., 2021) as the following. The data generating distribution is defined as $(\mathbf{x}, y) \sim \mathcal{D}$, where the unknown labels belong to one of C classes: $y \in \mathcal{Y} = \{1, ..., C\}$. An unlabelled training set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, and m labelling functions (LFs) $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{x}) \in \{0, 1, ..., C\}^m$ are

provided by the user, where 0 means that the LF abstained from labelling for any class. The one-hot representation of the LF votes provided by the m LFs for C classes is written as $\bar{\boldsymbol{\lambda}} = (\mathbb{1}\{\boldsymbol{\lambda} = 1\}, \ldots, \mathbb{1}\{\boldsymbol{\lambda} = C\}) \in \{0,1\}^{m \times C}$. The goal set by the authors is to train a downstream model $f : \mathcal{X} \to \mathcal{Y}$ on a *noise-aware* loss $L(y_f, y_e)$. The loss operates on the model's predictions $y_f = f(\mathbf{x})$ and *probabilistic labels* y_e generated by an encoder model e. The encoder model has access to LF votes, $\boldsymbol{\lambda}$, and features, $\mathbf{x}$.

2.7.3.2 Posterior Reparameterisation

The latent label is viewed by the authors (Cachay et al., 2021) as an aggregation of the LF votes that is a function of the entire set of LF votes and features on a sample-by-sample basis. The probability of a particular sample $\mathbf{x}$ having the class label $c \in \mathcal{Y}$ is modelled as

$$P_\theta(y = c|\, \boldsymbol{\lambda}) = \text{softmax}\,(\mathbf{s})_c\, P(y = c), \tag{16}$$

$$\mathbf{s} = \theta(\boldsymbol{\lambda}, \mathbf{x})^T \bar{\boldsymbol{\lambda}} \in \mathbb{R}^C. \tag{17}$$

where $\theta(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^m$ is weighted by the LF votes on a sample-by-sample basis and the softmax for class c on s is defined as

$$\text{softmax}\,(\mathbf{s})_c = \exp\left(\theta(\boldsymbol{\lambda}, \mathbf{x})^T \mathbb{1}\{\boldsymbol{\lambda} = c\}\right) \sum_{j=1}^{C} \exp\left(\theta(\boldsymbol{\lambda}, \mathbf{x})^T \mathbb{1}\{\boldsymbol{\lambda} = j\}\right).$$

The formulation by the authors can be seen as a reparameterisation of the posterior of the pairwise Markov Random Fields, where θ corresponds to the LF accuracies that are fixed across the dataset and are solely learned via LF agreement and disagreement signals, ignoring the informative features.

2.7.3.3 Neural Encoder

The goal of the authors (Cachay et al., 2021) is to estimate latent labels by means of learning sample-dependent accuracy scores $\theta(\boldsymbol{\lambda}, \mathbf{x})$, which it is proposed to parameterise by a neural encoder e. The network takes as input the features $\mathbf{x}$ and the corresponding LF outputs $\boldsymbol{\lambda}(\mathbf{x})$ for a data point, and outputs unnormalized scores $e(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^m$. It is defined

$$\theta(\boldsymbol{\lambda}, \mathbf{x}) = \tau_2 \cdot \text{softmax}\,(e(\boldsymbol{\lambda}, \mathbf{x})\tau_1)\,, \tag{18}$$

where τ_2 is defined as a constant factor that scales the final softmax transformation in relation to the number of LFs m. Hyperparameter τ_1 is defined as an inverse temperature that controls the smoothness of the predicted accuracy scores.

3 Related Work about Unsupervised & Weakly-Supervised Text Classification Methods

This section will present some historical approaches to unsupervised text classification, dataless text classification, topic modelling-based approaches, and weakly-supervised approaches.

3.1 Historical Approaches

Lack of sufficient training data has been a problem since the task of text classification was formally defined in the research literature.

One of the first works that discussed and proposed computational approaches for unsupervised text classification is by McCallum and Nigam (1999). In the approach they propose, no labelled documents are required. The approach uses a small set of keywords per class, a class hierarchy, and many unlabelled documents. The main idea is to assign approximate labels by term matching the keywords. These labels can be used for a bootstrapping process where a Naive Bayes classifier is learned using Expectation-Maximisation (Dempster et al., 1977) and hierarchical shrinkage.

For example, in another work by Nigam et al. (2000), they also augmented a small number of labelled texts with a much larger amount of unlabelled documents. They developed an algorithm that uses Naive Bayes with Expectation-Maximisation further.

3.2 Dataless Text Classification

Chang et al. (2008) proposed a method of using Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) to perform dataless text classification. The model uses Wikipedia as its source of knowledge. The texts and labels are mapped to a common semantic space, and the label with the highest matching score is returned.

Dataless text classification has also been applied to hierarchical textual data. This classification setting was studied by Song and Roth (2014).

Li et al. (2018a) proposed an unsupervised representation learning model that directly classifies documents without the need for labelled training data. The method takes class names as input and applies a cascade embedding approach. They first embed the seeded category names and other phrases into vectors to capture concept semantics. Next, the concepts are embedded into a hidden category space to make the category information explicit.

Meng et al. (2018) proposed a weakly-supervised neural text classification method, WeSTClass. The method consists of two main parts, a pseudo-document

generator that leverages seed information to generate pseudo-labelled documents for model pre-training and a self-training module that bootstraps on actual unlabelled data for model refinement.

Another dataless text classification method, the Label-Name-Only Text Classification (LOTClass), was proposed by Meng et al. (2020). The authors propose a language model self-training approach wherein a pre-trained neural language model is used as the general knowledge source for category understanding and a feature representation learning model for classification. The language model creates contextualised word-level category supervision from unlabelled data to train itself and then generalises to document-level classification via a self-training objective.

Chu et al. (2020) proposed an unsupervised label refinement approach which reduces the sensitivity to the choice of label descriptions by refining a dataless classifier's set of predictions using k-means (Lloyd, 1982) clustering.

Mekala and Shang (2020) proposed a novel framework ConWea. It provides contextualised supervision for text classification. Specifically, the framework leverages contextualised representations of word occurrences and seed word information to automatically differentiate multiple interpretations of the same word and create a contextualised corpus.

Yang et al. (2020) proposed a pseudo-label based dataless Naive Bayes classifier with seed words.

The DocSCAN method was proposed by Stammbach and Ash (2021). The method leverages large pre-trained language models, and it uses Semantic Clustering by Adopting Nearest-Neighbors.

3.3 Topic Models Based Methods

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) can be used to classify texts. One of the first papers that proposes an LDA-based document classification algorithm which does not require any labelled dataset is published by Hingmire et al. (2013). In their proposed approach, they construct a topic model using LDA, after which they assign one topic to one of the class labels. After that, they aggregate all the same class label topics into a single topic using the aggregation property of the Dirichlet distribution. Finally, they automatically assign a class label to each unlabelled document depending on its closeness to one of the aggregated topics. The same author has published the Topic Labeled Classification approach (Hingmire and Chakraborti, 2014b) and the Sprinkling Topics (Hingmire and Chakraborti, 2014a) approach.

Other authors have also developed methods that use LDA. Chen et al. (2015) propose a Descriptive LDA approach. Li et al. (2016) propose a Seed-Guided Topic Model approach where for each category, a small set of seedwords that are relevant

to the semantic meaning of the category are needed. Li et al. (2018b) propose a novel Laplacian seed word topic model.

3.4 Weakly-Supervised Text Classification

Ratner et al. (2016) in their seminal work propose a paradigm for the programmatic creation of training sets called data programming. This is one of the backbones of the models used in this work. In data programming, users express weak supervision strategies or domain heuristics as labelling functions, which are programs that label subsets of the data, but that are noisy and may conflict.

Ratner et al. (2017) proposed the `Snorkel` framework, a system that enables users to train state-of-the-art models without hand labelling any training data. `Snorkel` denoises their outputs without access to ground truth by incorporating the first end-to-end implementation of the data programming paradigm.

Ratner et al. (2018) propose a framework for integrating and modelling weak supervision sources by viewing them as labelling different related sub-tasks of a problem, called the `MeTaL` framework. More information on this was provided in section 2.7.1.

Both of the models used in this work, `COSINE` (Yu et al., 2021) and `WeaSEL` (Cachay et al., 2021) take into account that their input is a weak label generated from the output of a label model.

4 Methodology and Experimental Setup

In this section, an overview of the used datasets will be given, and the setup of the models along with implementation details will be provided.

4.1 Overview of the Datasets

There are two primary datasets used in this thesis, the Estonian Valence Corpus (Pajupuu et al., 2016), which is a small annotated sentiment analysis dataset in Estonian and the Postimees Corpus (a subcorpus from Kaalep et al. (2010); Muischnek (2011)), which is a large unannotated dataset. The training sets of the Valence and Postimees datasets will be joined to create a new large dataset – the Combined dataset.

4.1.1 Estonian Valence Corpus Dataset

The first annotated dataset used in this work is the Estonian Valence Corpus (Pajupuu et al., 2016) dataset. The corpus originally consisted of 4088 online daily and weekly articles' and reader comments' paragraph-level extracts. The dataset contains labels for both sentiment and rubric, but only the sentiment analysis is included in this thesis. Sentiment labels are Positive (class 1), Negative (class 0), Neutral (class 2), and Ambiguous. Training data is split into the same 70/10/20 per cent training, development, and test sets as were done by Tanvir et al. (2021); Kittask et al. (2020). Duplicate paragraphs were also removed by Kittask et al. (2020), resulting in 4068 remaining paragraphs. The text counts can be seen in Table 1. In this work, the Ambiguous class will be left out following the machine learning approach of Pajupuu et al. (2016). This way, it would be possible to compare the results to Kittask et al. (2020) as well.

Table 1. Sentiment classes text (paragraphs) counts of the Estonian Valence Corpus dataset. Table from Tanvir et al. (2021).

Label	**Train**	**Development**	**Test**	**Total**
Positive	612	87	175	874
Negative	1347	191	385	1923
Neutral	505	74	142	721
Ambiguous	385	55	110	550
Total	2849	407	812	4068

To understand the content of the Estonian Valence Corpus dataset better, examples of some texts and their English translations will be provided from the original article where the dataset was proposed (Pajupuu et al., 2016).

Positive text example: *Koht, mis varem ei olnud püha, võib selleks saada. Kui istutame tammikud, muudame need kohad pühaks. Hoolitseme ka selle eest, et tammikutes kasvaks kaunis kask ja püha pihlakas, et kaugete esivanemate vaimud end seal hästi tunneksid.*: A place that previously was not holy can become like that. We can make it holy ourselves by planting an oak forest. Moreover, let us take care that the oak forest also features the beautiful birch and the protective rowan, just to make the distant ancestral spirits feel good.

Negative text example: *Tabati ka üks kriminaalses joobes sõidukijuht. See juhtus pühapäeva öösel kella 4 ajal, kui Viljandis Lääne tänaval peeti kinni sõiduauto BMW, mille roolis oli 21-aastane noormees. Tema suhtes alustati kriminaalmenetlust.*: Also, a criminally intoxicated driver was apprehended. It happened at 4 o'clock Sunday morning that a BMW driven by a 21-year old was stopped on Lääne St. in Viljandi. Criminal charges were filed.

Neutral text example: *Peaaegu samasugune nägi pööning välja märtsis, kui kunstnik oli sinna üles seadnud "Asjade" esimese osa. Vahepealse kuue kuu jooksul on katusealune ja seda külastanud vaatajad osa saanud suurtest muudatustest.*: The attic looked almost the same in March, just after the artist had set up the first part of the "Things". During the six months passed, the attic and its visitors have been exposed to some considerable changes.

Before applying the labelling functions to the dataset, all texts were preprocessed. Texts were tokenised and lemmatised using EstNLTK 1.6 (Laur et al., 2020). No preprocessing was applied to the texts for training the end or joint models.

4.1.2 Examples of Texts Difficult to Label

The Estonian Valence Corpus dataset contains some texts for which it is difficult to assign a gold label from the perspective of a human annotator. In addition to that, some of the gold labels might be incorrect to some extent. The author of this thesis will provide some subjective examples and provide a different (one possible) interpretation of the labels assigned to some texts.

Gold label set as positive: *Elavaid pilte on pärast rahvusvahelist esilinastust A-klassi festivalil Shanghais näidatud festivalidel Leedus, Venemaal, Gruusias, Soomes, Kanadas ja Indias.*: After its international premiere at a top-class film festival in Shanghai *Living Images* (movie festival) has been shown in Lithuania, Russia, Georgia, Finland, Canada and India. – It is considered positive that a movie was shown in many film festivals all across the world. The paragraph could also be

considered neutral because although it talks about the movie's success, it is done neutrally, in the opinion of the author of this thesis.

Gold label set as negative: *Esimene pruut võttis unerohtu.*: The first bride took sleeping pills. – Taking sleeping pills is considered negative in this paragraph, which has no further context. This text could also be classified as neutral because the sentiment of taking sleeping pills is neither positive nor negative, in the opinion of the author of this thesis.

Gold label set as neutral: *Ehh, ikka ja alati taandub kõik sellele, kuidas keegi käitub, mõni on kainenagi täiesti talumatu käitumisega.*: Oh, now and always, everything boils down to how somebody behaves; some people have unbearable behaviour when being sober. – This text is considered neutral but might also be classified as negative because the text might reflect some disappointment towards somebody. Unbearable behaviour might be considered to be a negative trait in the opinion of the author of this thesis.

Having seen these examples, it could be concluded that some gold labels could be different and that it might be challenging to set a single label as the correct gold label. In some cases, a text could be assigned different labels, given different interpretations.

4.1.3 Postimees Corpus Dataset

The other unannotated dataset used in this thesis is the Postimees Corpus dataset. Postimees is an Estonian daily newspaper. The Postimees Corpus is a subcorpus of the Estonian Reference Corpus[3] (Kaalep et al., 2010; Muischnek, 2011) that contains the year 1995–2000 Postimees articles.

This corpus was compiled of paragraphs (like the Valence Corpus). Train and test splits were created for the Postimees corpus. The test split consisted of paragraphs only from the year 1995. Train split consisted of the paragraphs from the years 1996–2000. The number of paragraphs and the number of original documents where the paragraphs were extracted can be seen in Table 2.

Table 2. The paragraphs and documents counts of the Postimees corpus.

Split	**Paragraphs Count**	**Documents Count**
Train (1996-2000)	98244	11629
Test (1995)	1429	200

[3] https://www.cl.ut.ee/korpused/segakorpus/index.php?lang=en

In the Estonian Reference Corpus, information about every daily paper is stored in its individual XML file. EstNLTK 1.6 (Laur et al., 2020) provides a parser for the corpus.

There were two main principles in choosing which texts to include in the gathered Postimees dataset, as not all texts from the original corpus were included. The principles were:

- Opinion story genre - Every article has meta information attached to it that contains information about the topic. Only articles on the topic "opinion" (in Estonian: *arvamus*) were selected because the Estonian Valence Corpus was constructed from similar texts and opinion stories generally tend to have a concrete sentiment.

- Paragraph token length set between 10 and 259 - Another restriction was set to the token length of the paragraphs. Only texts with a token length between 10 and 259 were chosen (both inclusive). Texts shorter than 10 tokens generally described the author of the story and are not relevant in terms of classification, and 259 was the maximal number of tokens in the longest paragraph of the Valence Corpus.

As an example, the following text is from the Postimees newspaper from December 4, 1995.

Text example: *Nii ongi, et noored sellest kombest midagi ei tea, neile pole seda ehk õpetatudki. Nõnda hakkavad vanad ja toredad kombed kaduma. Kahju küll.*: So it is that the young people do not know about this tradition, they might even not be taught regarding that. Like this, old and nice traditions are starting to go away. What a pity.

Although the corpus is unannotated, this text could be classified as negative.

4.1.4 Combined Dataset

The Valence and Postimees Corpora datasets' training sets will be concatenated to create a training set for the Combined dataset. The Valence development and test sets will be set as the corresponding sets for the Combined dataset as they are annotated.

4.2 Setup of the Models

In this section, some details will be provided regarding obtaining weak labels, models' hyperparameters, and all models' training setup.

4.2.1 Engineering Labelling Functions

Estonian Valence Lexicon[4] (Pajupuu et al., 2012) was used as a knowledge base for generating some LFs. The lexicon contains keywords which refer to positive and negative sentiments. In the lexicon, words with positive sentiment have been assigned a positive integer score, whereas negative sentiment words have been assigned negative scores. The lexicon does not contain neutral words. The lexicon also contains words of different cases because some cases negate the word. One example of this is the word *abita* (without help) which has a negative score, and, on the other hand, the word *abiga* (with help) has a positive sentiment score.

The first intention is to design LFs that check whether positive or negative keywords are present in the texts. Initially, the training set of the Valence dataset was used for finding dataset-specific keywords. Top-50 highest TF-IDF (Sparck Jones, 1972) value words were found separately using the training set. Some words were not class-indicative in the author's subjective opinion and were too dataset-specific, which were then removed by hand. After numerous experiments, this idea was dropped because it did not provide better validation accuracy.

The final heuristics used in LFs are the following:

1. Valence lexicon scores' sum (`valence_prediction`) – sum all of the input text's words' lexicon values. In case a word is a negation (*ei* – no; *ega* – nor), the following word's lexicon score will be subtracted (neutral word's score will also be subtracted by 1). If $valence\_score \geq 1$ output `POSITIVE`, if $valence\_score \leq -1$, output `NEGATIVE`, otherwise output `ABSTAIN` ($valence\_score = 0$).

2. Positive keywords (`keywords_positive`) – if the text's tokens or lemmas include any positive word from the lexicon, output `POSITIVE`, otherwise `ABSTAIN`.

3. Negative keywords (`keywords_negative`) – if the text's tokens or lemmas include any negative word from the lexicon, output `NEGATIVE`, otherwise `ABSTAIN`.

4. Positive and negative keywords' counts (`count_positive_negative`) – if there are no positive nor negative words from the lexicon among tokens and lemmas, output `NEUTRAL`, otherwise `ABSTAIN`.

The LFs were applied to the Valence and Postimees datasets using the `Snorkel`[5] (Ratner et al., 2017) Python library. The Valence dataset LF statistics can be seen in Table 3.

[4] `http://peeter.eki.ee:5000/Valentsisonastik_Eesti%20Keele%20Instituut%202014.xlsx`
[5] `https://www.snorkel.org/`

Table 3. Labelling functions statistical information of the Valence dataset.

LF name	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
valence_prediction	[0, 1]	68.51	68.51	50.16	1215	473	71.98
keywords_positive	[1]	79.67	78.00	66.72	562	1401	28.63
keywords_negative	[0]	69.44	69.20	62.13	1073	638	62.71
count_positive_negative	[2]	26.54	18.75	18.75	274	380	41.90

If there were only two classes, positive and negative, then the first labelling function, valence_prediction would achieve an accuracy of 71.97% on its own on the Valence dataset. As seen from the next two LFs, identifying positive keywords is more difficult than negative ones because the empirical accuracy is lower. The last LF is the only one detecting the neutral class.

The same LFs were applied to the training set of the Postimees dataset. A similar table can be constructed for the Postimees dataset, but since it is unannotated, it is impossible to have correctly and incorrectly classified labels' counts and calculate empirical accuracy. The Postimees dataset statistics can be seen in Table 4.

Table 4. Labelling functions statistical information of the Postimees dataset.

LF name	Polarity	Coverage	Overlaps	Conflicts
valence_prediction	[0, 1]	75.02	75.02	63.98
keywords_positive	[1]	90.80	89.48	82.68
keywords_negative	[0]	84.96	84.87	80.64
count_positive_negative	[2]	14.21	12.36	12.36

The first three LFs have larger coverage, overlaps and conflicts than the Valence corpus. The only LF that detects neutral words has smaller coverage, overlaps, and conflicts than the LFs of the Valence corpus.

4.2.2 Label Model Setup

The MeTaL (Ratner et al., 2018) label model will be used in this work to generate weak labels for all of the texts because its average accuracy across many different datasets proved to be the highest in the WRENCH study (Zhang et al., 2021). MeTaL (Ratner et al., 2018) learns a reweighted model of LFs and uses the combined signal to train a hierarchical multi-task network which is automatically compiled from the structure of the sub-tasks.

A grid search was performed to find the best-performing hyperparameters using the development set accuracy as the evaluation metric. The selected hyperparameters from the grid search can be seen in Table 5.

Table 5. MeTaL label model hyperparameters. Parameters are from (Zhang et al., 2021). Underlined values are the best-performing hyperparameters for the Valence dataset. **Bold** parameters are the best values for the Combined dataset.

Hyperparameter	Description	Range
lr	learning rate	1e-6,1e-5,1e-4,1e-3,**1e-2**,1e-1
weight_decay	weight decay	**1e-6**,1e-5,1e-4,1e-3,1e-2,1e-1
num_epoch	the number of training epochs	5,10,50,100,200,300,**500**,1000

The random seed was fixed for the duration of the grid search. When converting probabilistic labels to predictions, the random tie-breaking policy[6] was used, using a deterministic hash for reproducibility. When the model had to choose between the two best choices with the exact weights, it was done randomly, but the choices were consistent for every program run (not truly random).

4.2.3 End & Joint Model Setup

Two different end models and one joint model will be trained in this work. The models are:

1. BERT (Devlin et al., 2019) – all layers of the pre-trained EstBERT (Tanvir et al., 2021) are fine-tuned. The final prediction is the output of the fully-connected classification layer added on top of the [CLS] token (see section 2.6).

2. Contrastive Self-Training for Fine-Tuning Pre-trained Language Model (COSINE) (Yu et al., 2021) – contrastive regularisation and confidence-based reweighting are used, gradually improving model fitting while effectively suppressing error propagation (see section 2.7.2).

3. Weakly-Supervised End-To-End Learner Model (WeaSEL) (Cachay et al., 2021) – directly learning the downstream model by maximising its agreement with probabilistic labels generated by reparameterising prior probabilistic posteriors with a neural network (see section 2.7.3).

Models were trained using the code provided with the WRENCH framework (Zhang et al., 2021). The hyperparameters used in the grid search mainly were the same as outlined in the WRENCH paper (with some minor modifications) and can be seen in Table 6. The WRENCH library is based on Python and PyTorch. For all of the models, the common parameters are the following:

- AdamW optimiser;

[6] https://snorkel.readthedocs.io/en/v0.9.3/packages/_autosummary/labeling/snorkel.labeling.LabelModel.html#snorkel.labeling.LabelModel

Table 6. Hyperparameters and search space (Zhang et al., 2021). Underlined values are hyperparameters used for training the Valence dataset models, **bold** values for training the Combined dataset models.

Model	Hyperparameter	Description	Range
Supervised BERT	batch_size	the input batch size	16,32
	lr	learning rate	1e-6,5e-6,1e-5,3e-5,5e-5,1e-4
	dropout	the proportion of neurons to drop during training	0.1,0.2,0.3,0.4,0.5
	weight_decay	weight decay	1e-4
Weakly-supervised BERT	batch_size	the input batch size	16,**32**
	lr	learning rate	1e-6,5e-6,1e-5,**3e-5**,5e-5,1e-4
	dropout	the proportion of neurons to drop during training	0.1,0.2,**0.3**,0.4,0.5
	weight_decay	weight decay	**1e-4**
COSINE	batch_size	the input batch size	**16**
	lr	learning rate	1e-6,5e-6,1e-5,**3e-5**,5e-5,7e-5,1e-4
	weight_decay	weight decay	**1e-4**
	T	the period of updating model	50,100,**200**
	ξ	the confident threshold	**0.2**,0.5,1.0
	λ	the weight for confidence regularisation	**0.01**,0.05
	μ	the weight for contrastive regularisation	**0.1**,0.5,1
	γ	the margin for contrastive regularisation	**0.1**,0.5
	dropout	the proportion of neurons to drop during training	0.3,0.4,**0.5**
WeaSEL	batch_size	the input batch size	**16**
	lr	learning rate	**1e-6**, 5e-6, 1e-5, 3e-5, 5e-5, 7e-5, 1e-4
	weight_decay	weight decay	**1e-4**
	hidden_size	number of neurons in the fully-connected layer	**100**,200,500
	temperature	the temperature parameter	0.1, 0.3, 0.5, **1.0**
	dropout	the proportion of neurons to drop during training	0.3,0.4,**0.5**

- Linear learning rate scheduler;
- Models were trained for 10 000 training steps;
- WRENCH uses HuggingFace for pre-trained transformer models, EstBERT[7] (Tanvir et al., 2021) was used as the implementation of BERT;
- All other hyperparameters were left as default.

All of the experiments were carried out at the University of Tartu High-Performance Computing Center (University of Tartu, 2018). One Tesla V100 GPU was used to train all of the models.

[7] https://huggingface.co/tartuNLP/EstBERT

5 Experiment Results

In this section, all of the results will be presented. The code can be accessed from the author's GitHub repository[8].

All of the different dataset splits will be evaluated using the gold labels set by Pajupuu et al. (2016) in all of the experiments, including the training set.

5.1 Label Model

The performance metrics of the label model trained with the best hyperparameters using the Valence dataset can be seen in Table 7.

Table 7. Label model performance metrics for the Valence dataset. Results are shown in percentages, test accuracy in **bold**.

MeTaL	Accuracy	Weighted F1	Weighted Precision	Macro Recall
Train	65.06	65.18	65.61	61.00
Dev	65.63	64.89	64.75	59.24
Test	**64.81**	64.67	64.85	59.94

The label model was trained from scratch again for the Combined dataset. As the Postimees dataset does not contain gold labels, it is impossible to calculate the train set performance metrics. Evaluation results for the Combined dataset can be seen in Table 8. The development and test sets are the same Valence dataset development and test sets (as were shown in Table 7).

Table 8. Label model performance metrics percentages for the Combined dataset. Results are shown in percentages, test accuracy in **bold**.

MeTaL	Accuracy	Weighted F1	Weighted Precision	Macro Recall
Dev	63.92	62.18	62.04	55.98
Test	**64.81**	63.58	63.56	57.42

The performance of both label models is surprisingly high, considering that no machine learning approaches were used and that only four heuristic labelling functions were used. The aim of the end models is to get even better performance out of the weak labels.

[8] https://github.com/andreaspung/ws-estonian-sentiment-analysis

5.2 Valence Dataset

The Valence dataset has been classified using several supervised transformer-based models in previous works (Tanvir et al., 2021; Kittask et al., 2020). For a better comparison with the results of this thesis, the classification results by Tanvir et al. (2021) are presented in Table 9. The models shown are trained on two sequence lengths, 128 and 512, but in this thesis, the `EstBERT` (Tanvir et al., 2021) model trained on a sequence length of 128 will be used.

The fully-supervised method (trained using XLM-RoBERTa representations) achieved an accuracy of 76.07% on the test set. In this thesis, the train, development and test sets are the same as those used by Tanvir et al. (2021), so it is acceptable to compare these results.

Table 9. Fully supervised models' Valence dataset test set accuracies (Tanvir et al., 2021). The highest scores in each column are in **bold**.

Model	**Seq=128**	**Seq=512**
`EstBERT`	74.36	74.50
WikiBERT-et	68.09	69.37
mBERT	70.23	69.52
XLM-RoBERTa	**74.50**	**76.07**

This thesis seeks to evaluate the models without using any gold labels and only weak labels. The weakly-supervised models' evaluation results for the Valence dataset can be seen in Table 10. The fully-supervised model had the best results overall. The `EstBERT` model trained on the gold labels got a similar average test accuracy of 74.44% when comparing the results to Tanvir et al. (2021). They reported a test accuracy of 74.36%. The weakly-supervised `BERT` model got around 12% worse accuracy and weighted F1 scores when compared to the fully-supervised model. The `COSINE` model achieved a little better accuracy and weighted F1 score than the weakly-supervised BERT, 64.22% and 63.42%, respectively. The `WeaSEL` achieved even better accuracy than the `COSINE` model but had a much lower weighted F1 score.

Table 10. Models' performance metrics on the Valence dataset. Average percentages over five runs are displayed; value in parenthesis shows standard deviation.

Model & Split	Accuracy	Weighted F1	Weighted Precision	Macro Recall
Supervised EstBERT-Train	99.48 (1.14)	99.48(1.14)	99.48 (1.13)	99.35 (1.42)
EstBERT-Dev	78.64 (0.88)	78.16 (0.82)	78.24 (0.92)	73.10 (1.08)
EstBERT-Test	74.44 (0.90)	73.82 (0.89)	73.90 (0.83)	67.75 (1.41)
Weakly-supervised EstBERT-Train	64.38 (0.77)	64.80 (0.64)	65.95 (0.36)	61.50 (0.35)
EstBERT-Dev	66.25 (1.09)	65.67 (0.90)	66.05 (0.65)	59.77 (0.57)
EstBERT-Test	62.88 (0.95)	62.73 (0.83)	63.71 (0.62)	57.83 (0.66)
COSINE-Train	66.27 (0.72)	66.21 (0.53)	66.70 (0.23)	61.03 (0.80)
COSINE-Dev	66.36 (0.82)	65.28 (1.38)	65.24 (1.34)	58.81 (1.34)
COSINE-Test	64.22 (1.04)	63.42 (1.28)	63.45 (0.97)	56.72 (1.56)
WeaSEL-Train	64.94 (1.79)	59.07 (2.80)	59.04 (7.88)	52.97 (2.86)
WeaSEL-Dev	67.22 (0.89)	60.06 (2.09)	58.03 (7.23)	54.19 (2.57)
WeaSEL-Test	65.27 (1.27)	58.86 (2.39)	60.34 (8.89)	52.01 (2.73)

The train set performance is also displayed. The supervised model can get very high train set accuracies, but neither of the weakly-supervised models can learn beyond the label model's performance. This is because the *weak* labels that were found by the author are not the same as the *gold* training set labels provided by Pajupuu et al. (2016).

5.3 Combined Dataset

The performance metrics of the models trained on the Combined dataset can be seen in Table 11. The weakly-supervised BERT model managed to get, on average better accuracy and weighted F1 scores when the model was trained on the Combined dataset instead of the Valence dataset. The COSINE model achieved the best results among all weakly-supervised models. On the test set, it achieved an accuracy of 67.15% and a weighted F1 score of 66.77%. Compared with the model trained on the Valence dataset, it managed to get a little better results. The WeaSEL model's results were worse on average when trained on the Combined dataset.

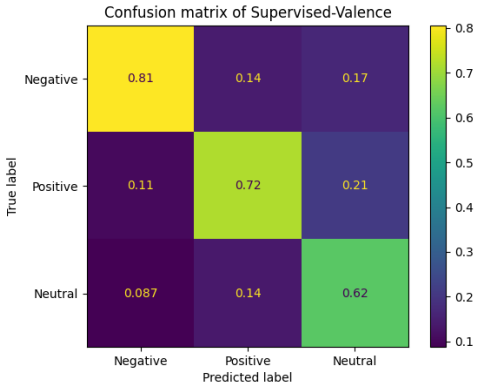
Table 11. Performance metrics of the models trained on the Combined dataset. Average percentages over five runs are displayed; value in parenthesis shows standard deviation.

Model & Split	Accuracy	Weighted F1	Weighted Precision	Macro Recall
EstBERT-Dev	65.63 (0.57)	63.72 (0.58)	63.49 (0.58)	56.51 (0.83)
EstBERT-Test	65.73 (1.46)	64.39 (1.56)	64.45 (1.61)	57.66 (1.70)
COSINE-Dev	67.05 (0.00)	66.31 (0.37)	66.55 (0.61)	59.34 (0.74)
COSINE-Test	67.15 (0.13)	66.77 (0.14)	66.96 (0.43)	60.65 (0.43)
WeaSEL-Dev	66.65 (1.36)	58.70 (1.34)	55.82 (6.17)	52.83 (2.62)
WeaSEL-Test	64.42 (1.62)	57.26 (1.39)	61.75 (10.09)	50.35 (2.29)

Generally, the BERT and COSINE models were able to achieve better results on the larger, Combined dataset. That shows that having a larger training dataset had a good impact on these models' performance.

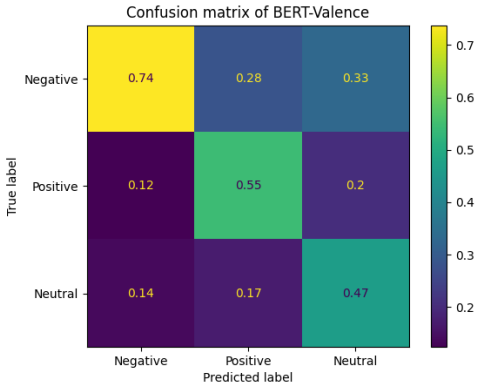
5.4 Confusion Matrices Analysis

Confusion matrices were found for all of the seven models – the fully-supervised BERT, weakly-supervised BERT, COSINE, WeaSEL models trained on the Valence dataset and weakly-supervised BERT, COSINE, WeaSEL models trained on the Combined dataset. The confusion matrices can be seen in Figure 6.

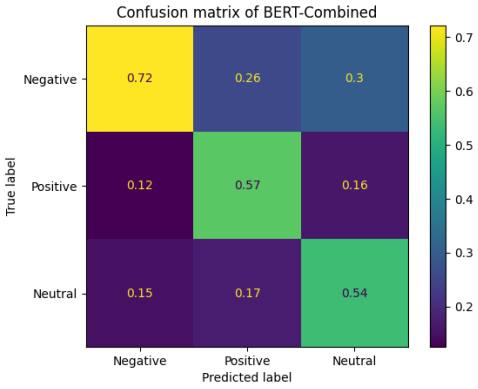


(a) The supervised BERT model trained on the Valence dataset with gold labels.

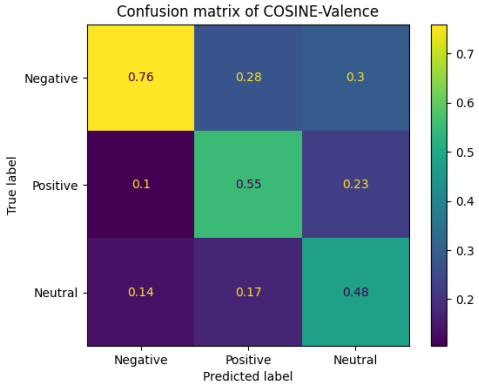
Figure 6. Confusion matrices of models trained on the Valence dataset.



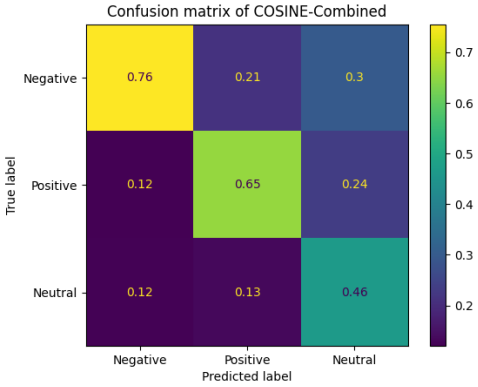
(b) The BERT model trained on the Valence dataset with weak labels.



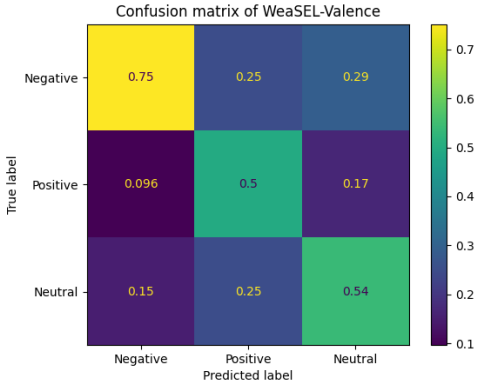
(c) The BERT model trained on the Combined dataset with weak labels.



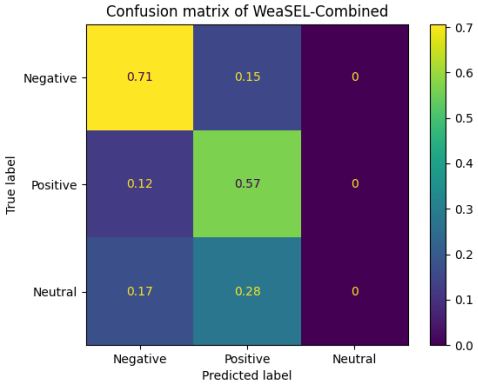
(d) The COSINE model trained on the Valence dataset with weak labels.



(e) The COSINE model trained on the Combined dataset with weak labels.



(f) The WeaSEL model trained on the Valence dataset with weak labels.



(g) The WeaSEL model trained on the Combined dataset with weak labels.

Figure 6. Confusion matrices of models trained on the Valence dataset.

The fully-supervised BERT model had the most mistakes when classifying neutral texts, which were positive. The negative class was detected the best when considering the three classes. The BERT model trained on the Valence dataset had the most difficulties classifying the neutral class. The same model but trained on the Combined dataset could detect the neutral class 7% better. The model learned some useful features having a larger training dataset. The COSINE model trained on the Valence dataset also had the worst performance in detecting the neutral class. Interestingly, when the model was trained on a larger dataset, the accuracy of detecting the neutral class dropped 2%. On the other hand, the accuracy of detecting the positive class was 10% higher. The WeaSEL model trained on the Valence dataset had the most difficulties in detecting the positive class, having an accuracy of 50%. The same model trained on the Combined dataset did not output any neutral labels.

5.5 Aggregation of Results

An aggregated overview of all main results can be seen in Table 12. The label model achieved an accuracy of around 10% higher than just predicting the negative class, so using a label model for this task and dataset overperforms just predicting the majority class and is definitely worth using. Among the models trained on the Valence dataset, the WeaSEL model managed to get a better average result than the label model. Among the models trained on the Combined dataset, only the WeaSEL model had worse test accuracy than the label model. It is worth noting that during the hyperparameter grid search of the COSINE model, there were a few runs where the development accuracy exceeded 70%. The development accuracy of the COSINE model was consistently higher on the development set than on the test set. This might show that the model has started to overfit and lose its generalisation power.

Table 12. Aggregation of the most important accuracies in percentages (averaged over five runs). The top section shows accuracies for general cases. The middle section shows accuracies for the models trained on the Valence dataset and lower section for the models trained on the Combined dataset.

Model	**Dev Accuracy**	**Test Accuracy**
Random Class	33.33	33.33
Majority Class (negative)	54.26	54.84
Supervised BERT (EstBERT)	78.64	74.44
Valence dataset		
MeTaL label model	65.63	64.81
BERT (EstBERT)	66.25	62.88
COSINE	66.36	64.22
WeaSEL	67.22	65.27
Combined dataset		
MeTaL label model	63.92	64.81
BERT (EstBERT)	65.63	65.73
COSINE	67.05	67.15
WeaSEL	66.65	64.42

Considering that no gold labels were used for training weakly-supervised models, and the best weakly-supervised model by average achieved a result of 7.29% less than using solely gold labels for model training (supervised BERT), it is up to the application's developer to decide whether such performance loss is satisfactory or not. Generally, weakly-supervised models tend to have worse performance than fully-supervised models with a few exceptions, for example, using the Youtube dataset for model training (Zhang et al., 2021).

5.6 Examples of Incorrectly Classified Texts

There were many texts where one or multiple models assigned an incorrect class. This section will provide examples of six texts where at least one of the models' predictions was incorrect. Classification results for the following texts can be seen in Table 13.

Table 13. Examples of classification results (0 is negative, 1 is positive, 2 is neutral; V - trained on the Valence dataset, C - trained on the Combined dataset).

Text Number	Gold Label	Supervised-V	BERT-V	BERT-C	COSINE-V	COSINE-C	WeaSEL-V	WeaSEL-C
1	1	0	0	0	0	0	1	0
2	1	0	2	0	0	0	0	0
3	0	0	1	1	1	2	1	1
4	0	0	0	0	2	2	2	0
5	2	0	1	1	2	1	1	1
6	2	2	2	2	0	2	0	0

1. True label positive but prediction negative. *Au poomüüjale, tema tegi mida eesti politsei ei julge teha: üks kord ja korralikult probleemist lahti saada*: Respect to the shopkeeper, he/she did what the Estonian Police is afraid of doing, getting rid of the problem once and for all.

2. True label positive but prediction neutral. *Kolmapäevaks olid meeleavaldused peaaegu vaibunud.*: By Monday, the protests had almost subsided.

3. True label Negative but predicted Positive. *Keda se huvitab? Nõme tegelane...*: Who cares? Silly person...

4. True label negative but prediction neutral. *Tuli hävitas Kohilas elumaja.*: Fire burned down a residential house in Kohila.

5. True label neutral but prediction positive. *Loe tähelepanelikult. Kui ta alustab juttu maxima külastamisest ja jõuab välja simnna, et see aitab rahast raha teha, on ta ise kuidagi selle tsunftiga seotud.*: Read carefully. When he/she starts talking about visiting Maxima and reaches the point that this will help to make money out of money, he/she is somehow connected with this guild.

6. True label neutral but prediction negative. *Täna pärastlõunal enne kella 16 olid teepinna temperatuurid Põhja-Eestis miinuskraadides, mujal Eestis plusskraadides.*: The road surface temperatures were negative today afternoon before 16 o'clock in Northern Estonia and positive in Estonia elsewhere.

The following subsection will explain how human labelling was carried out.

5.7 Human Labelling

All texts from the Postimees test set were classified by all seven models. To understand how the models perform on new unseen data, 100 texts were chosen from the Postimees test set, where all seven classifiers made as different decisions as possible. These texts were labelled by three human annotators, and after that, the models' performance metrics were calculated.

For every text in the Postimees test set, the counts of each individual label predictions were found. The texts were ordered in the following order of predicted class counts – 2+2+3, 1+3+3, 1+2+4, 1+1+5. For example, a text that received two positive, two negative, and three neutral labels had a higher priority than a text with one neutral, three positive, and three negative labels. The ordering of the labels was not taken into account, and all of the labels were treated as equally important. Top-100 texts were selected according to the previous ordering.

Three human annotators were asked to label all of the 100 Postimees texts by two different aspects. Firstly, it was asked to classify a text into one of the three categories – positive, negative, and neutral. Even if the text seemed ambiguous, the most dominating sentiment was asked to be labelled. Secondly, separately from the previous label, it was asked to classify the text as ambiguous or not ambiguous binarily. The annotators were asked to choose this option if they had difficulty deciding between one or the other label. In addition to that, the annotators could write a comment for each of the texts.

The labellers included the author of this thesis, his mother and his sister. They were sent an Excel file and examples of the gold standard of annotation from the original source (Pajupuu et al., 2016) and annotation instructions. The annotation instructions and example of the labelling Excel file can be seen in Appendix I.

5.8 Human Labelling Statistics

In Table 14, it is possible to see the assigned labels' counts by all three labellers. The ordering of the labellers is random. The counts of labels differed quite a lot among the labellers. For example, the counts of positive labels ranged from 5 to 26. In addition to that, the counts of ambiguous labels differed from 7 to 18 texts.

Table 14. Label counts of both aspects of all three labellers. The ordering of the labellers is random but will be consistently the same in this section.

Label	**Labeller 1**	**Labeller 2**	**Labeller 3**
Positive	5	26	18
Negative	37	30	40
Neutral	58	44	42
Ambiguous	18	7	12
Not Ambiguous	82	93	88

It is also possible to calculate the Cohen's Kappas (Cohen, 1960) between all of the labeller pairs. The values can be seen in Table 15. Labeller 2 and labeller 3 had the highest Cohen's Kappa of 0.5527, indicating moderate agreement. The lowest Cohen's Kappa was between labeller 1 and labeller 2 – 0.2912, which indicates a fair

agreement. No pairwise agreements were substantial or better. This indicates that it is not a trivial dataset to annotate, and every labeller has their interpretation and feeling of the text's dominating sentiment.

Table 15. Cohen's Kappas between the labels assigned by the three labellers.

Cohen's Kappas	**Labeller 1**	**Labeller 2**	**Labeller 3**
Labeller 1	1	0.2912	0.4494
Labeller 2	0.2912	1	0.5527
Labeller 3	0.4494	0.5527	1

Cohen's Kappas between all labeller pairs can be calculated for the ambiguous flag. The values can be seen in Table 16. The Kappas are much lower and indicate slight or fair agreement compared to the previous table results. It might be that classifying texts into ambiguous or not ambiguous is even more open to different interpretations by the labellers.

Table 16. Cohen's Kappas between the ambiguous flags assigned by the three labellers.

Cohen's Kappas	**Labeller 1**	**Labeller 2**	**Labeller 3**
Labeller 1	1	0.2438	0.0654
Labeller 2	0.2438	1	0.2494
Labeller 3	0.0654	0.2494	1

An inter-annotator agreement score was calculated between the three labellers using Fleiss' Kappa score (Fleiss, 1971). It equalled 0.4327 with the ambiguous texts included, which indicates a moderate agreement. The ambiguous texts were removed, and the Fleiss' Kappa score was computed again. The Fleiss' Kappa equalled 0.5548 without the ambiguous texts, indicating moderate agreement.

Based on the results of human labelling, ground truth labels could be found for the subset of top-100 most challenging to classify texts from the Postimees test set. Texts where all of the labellers had a different decision (one label of negative, positive and neutral), were removed, resulting in 96 texts. The ground truth label was set as a majority vote – if two labellers voted for one class and the third labeller for another class, then the first class was set as the ground truth label.

5.9 Postimees Test Subset Performance Analysis

Texts from the Postimees test subset were evaluated using the ground truth labels found using majority voting by three human labellers.

The analysis can be done in two different ways. The first approach is to do calculations using all 96 texts with previously defined ground truth labels, including all of the texts where at least one labeller labelled the text as ambiguous in the analysis. The other approach would be to remove all of the texts where at least one labeller classified the text as ambiguous. In this approach, there would be 70 texts left.

5.9.1 Results with Ambiguous Texts

The analysis was performed with the ambiguous texts included. The same performance metrics were calculated as were for the Valence dataset. Results of the Postimees dataset with the ambiguous texts included can be seen in Table 17.

The fully-supervised `BERT` model trained on the gold labels achieved the best performance metrics, although the accuracy and weighted F1 scores did not reach the Valence test set metrics. Understandably, the Valence test set did not include any ambiguous texts, but in this analysis, they were included, which may explain the lower performance.

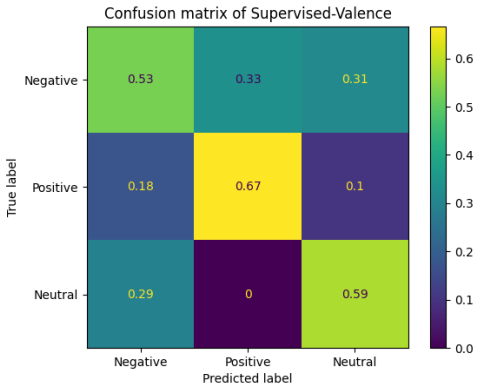
The `BERT` model trained on the weak labels for the smaller Valence dataset had better performance metrics (except weighted precision) than the model trained using the larger, Combined dataset. This might hint that the regular `BERT` model may not be able to learn the weak labels' noise correctly, and with a larger dataset, the performance decreases.

The `COSINE` model, on the other hand, works exactly as expected – the performance increases considerably. When comparing the trained models on the smaller and larger datasets, the accuracy increases from 34.38% to 45.83% (33.3% increase) and the weighted F1 score increases from 36.67% to 45.24% (23.4% increase).

The `WeaSEL` model, when trained on the larger dataset, managed to get better performance metrics as well, but the performance is not up to par with the `COSINE` model, except for macro recall.

Table 17. Performance metrics of texts with human labelled ground truth annotations with ambiguous texts. Bold values indicate higher performance metric values when comparing the same model trained on the Valence and Combined datasets. Underlined value shows the best value across all seven models.

Model & Dataset	Accuracy	Weighted F1	Weighted Precision	Macro Recall
Supervised-Valence	**58.33**	**54.37**	**57.93**	**51.03**
EstBERT-Valence	**35.42**	**34.94**	40.21	**32.47**
EstBERT-Combined	30.21	29.83	**45.58**	31.05
COSINE-Valence	34.38	36.67	44.47	31.14
COSINE-Combined	**45.83**	**45.24**	**47.37**	**40.98**
Weasel-Valence	28.12	19.65	17.66	41.91
Weasel-Combined	**32.29**	**23.71**	**19.93**	**42.52**



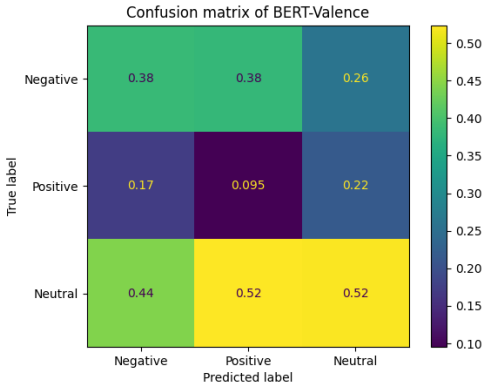
(a) The supervised BERT model trained on the Valence dataset with gold labels.

Figure 7. Confusion matrices of the trained models using the human ground truth labels with the ambiguous texts included.

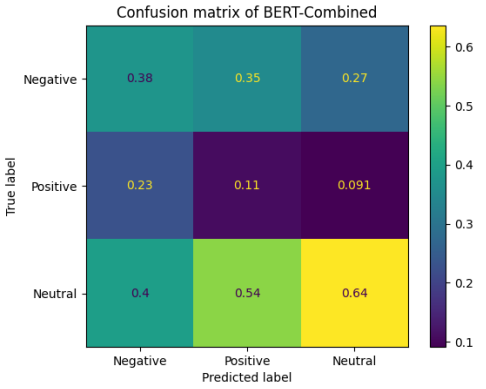
The confusion matrices taking the ambiguous texts into account of the seven models can be seen in Figure 7.

The fully-supervised BERT trained on the Valence dataset had the most mistakes when the true label was negative, and interestingly, more of the texts were classified as positive and neutral for this case. No actually neutral texts were classified as positive.

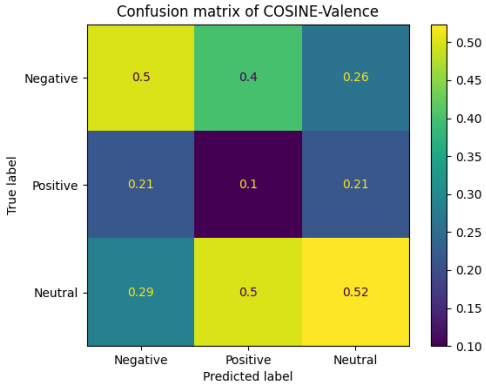
The BERT model trained on the Valence dataset had difficulties, especially in classifying the neutral class. When comparing to the BERT model trained on the Combined dataset, there were around twice as few misclassifications for the positive class, which were classified as neutral.



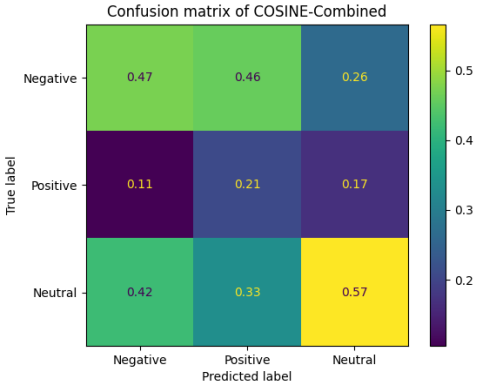
(b) The BERT model trained on the Valence dataset with weak labels.



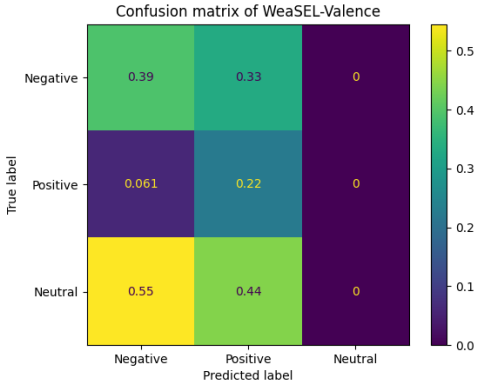
(c) The BERT model trained on the Combined dataset with weak labels.



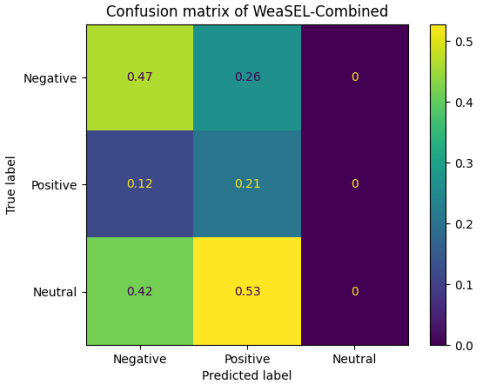
(d) The COSINE model trained on the Valence dataset with weak labels.



(e) The COSINE model trained on the Combined dataset with weak labels.



(f) The WeaSEL model trained on the Valence dataset with weak labels.



(g) The WeaSEL model trained on the Combined dataset with weak labels.

Figure 7. Confusion matrices of the trained models using the human ground truth labels with the ambiguous texts included.

The COSINE model trained on the Valence dataset misclassified the positive class the most, only around 10% of the texts were classified correctly. The model trained on the Combined dataset was still performing poorly for detecting the positive class, but the classification accuracy was around 21%, more than twice as much.

The WeaSEL model did not predict the neutral class at all. When comparing how the WeaSEL model improved on the larger dataset, it learned to predict the negative class a little better.

5.9.2 Results without Ambiguous Texts

Similar analyses will be presented, but the ambiguous texts will now be removed. There are a total of 70 texts. In some sense, removing ambiguous texts is more relevant because the models were not trained on the ambiguous class and this approach follows Pajupuu et al. (2016); Tanvir et al. (2021). The results can be seen in Table 18.

The fully-supervised BERT model trained on the gold labels achieved the best overall performance metrics. Compared to the previous analyses (see Table 17) where ambiguous texts were included, this time, the performance metrics are better. The accuracy increased from 58.33 to 64.29% (around a 10.2% increase) and the weighted F1 score from 54.37 to 61.17% (an increase of 12.5%).

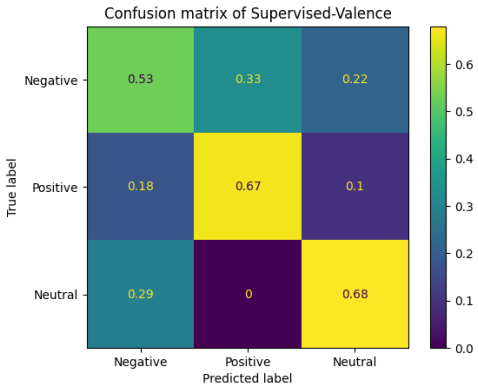
The performance metrics of the BERT model trained on the weak labels did not change significantly in this analysis. The model trained on the smaller dataset still performed better (except for weighted precision).

The results of the COSINE model changed significantly when the ambiguous texts were removed. The model trained on the Combined dataset managed to get accuracy and F1 score of 52.86% and 52.38%, respectively (48.0% and 31.9% increase compared to the model trained on the Valence dataset). In addition to that, when compared with the previous analyses, the model trained on the Combined dataset had better performance metrics. Still, the COSINE models did not reach the performance of the supervised model.

The WeaSEL model did not output neutral class. The model trained on the larger dataset had better performance across all four performance metrics. Despite not working as well as the other models, it can be seen that a larger weakly-labelled training dataset yields models with better performance.

Table 18. Performance metrics of texts with human labelled ground truth annotations with ambiguous texts removed. Bold values indicate higher performance metric values when comparing the same model trained on the Valence and Combined datasets. Underlined value shows the best value across all seven models.

Model & Dataset	Accuracy	Weighted F1	Weighted Precision	Macro Recall
Supervised-Valence	**64.29**	**61.17**	**63.29**	**50.01**
EstBERT-Valence	**35.71**	**36.73**	44.76	**34.38**
EstBERT-Combined	27.14	28.41	**48.92**	28.52
COSINE-Valence	35.71	39.70	47.61	26.78
COSINE-Combined	**52.86**	**52.38**	**53.06**	**43.46**
Weasel-Valence	22.86	14.59	12.14	39.37
Weasel-Combined	**28.57**	**19.09**	**15.58**	**45.71**



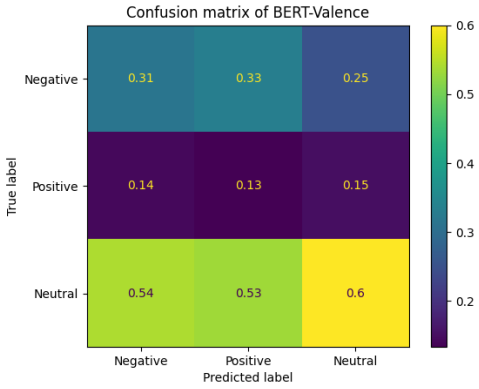
(a) The supervised BERT model trained on the Valence dataset with gold labels.

Figure 8. Confusion matrices of the trained models using the human ground truth labels with the ambiguous texts removed.

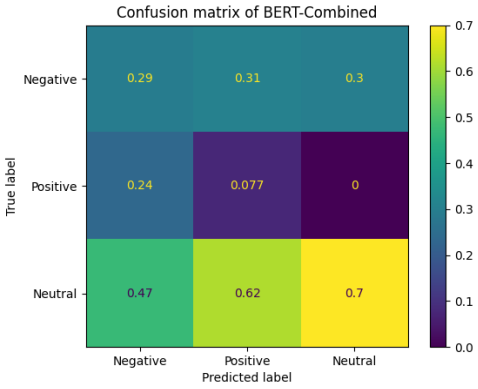
The confusion matrices of the models with the ambiguous texts removed can be seen in Figure 8.

The fully-supervised BERT trained on the Valence dataset had the most mistakes when the true label was negative, but the predicted label was positive. No actually neutral texts were classified as positive, and this might be due to the low number of positive texts in the Postimees test set.

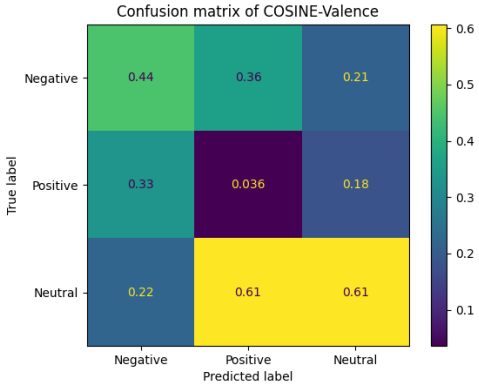
The BERT model trained on the Valence dataset had difficulties, especially in classifying the neutral class once again. When comparing to the BERT model trained on the Combined dataset, the most mistakes were made in classifying neutral texts, which were assigned a positive label. This might tell that the model is much



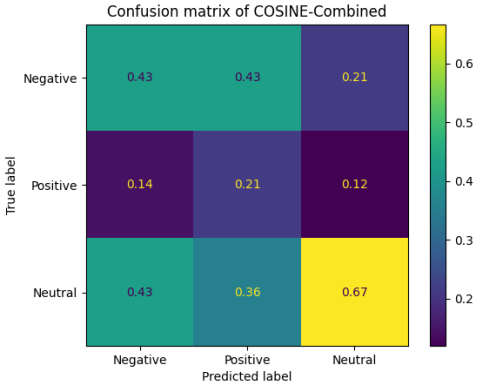
(b) The BERT model trained on the Valence dataset with weak labels.



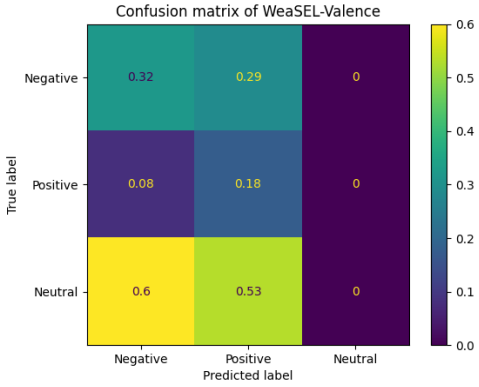
(c) The BERT model trained on the Combined dataset with weak labels.



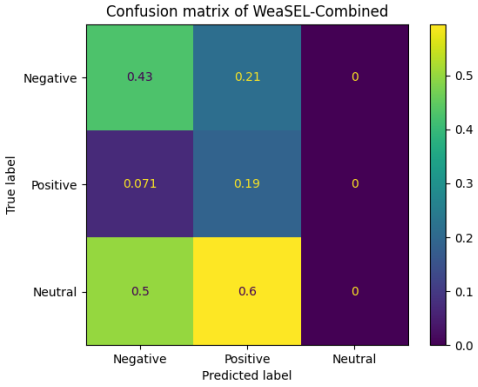
(d) The COSINE model trained on the Valence dataset with weak labels.



(e) The COSINE model trained on the Combined dataset with weak labels.



(f) The WeaSEL model trained on the Valence dataset with weak labels.



(g) The WeaSEL model trained on the Combined dataset with weak labels.

Figure 8. Confusion matrices of the trained models using the human ground truth labels with the ambiguous texts removed.

more inclined towards outputting positive labels than neutral labels and that the labelling functions might be constructed poorly.

The COSINE model trained on the Valence dataset misclassified the positive class the most, only around 10% of the texts were classified correctly, but the positive class had the least amount of texts as well. The model had particular difficulty classifying the neutral class, for which the positive label was given. The model trained on the Combined dataset was still performing poorly for detecting the positive class, but the classification accuracy was around 21%, more than twice as much. In addition to that, the model did not output the negative and positive classes that much when compared with the model trained on the Valence dataset, when the actual true class was neutral.

The WeaSEL model did not predict the neutral class. When comparing how the WeaSEL model improved on the larger dataset, it learned to predict the negative class a little better and predicted the positive class with around the same accuracy.

5.10 Discussion & Future Work

In this section, some of the most important findings will be discussed.

Tradeoff between hand-labelling more texts and leveraging weak supervision. When comparing fully-supervised and weakly-supervised models, there is a tradeoff between model performance and the work needed to either hand-label texts or develop labelling functions. Hand-labelling takes a significant amount of time, effort and is costly. Developing labelling functions takes less time, but many times larger datasets could be automatically assigned weak labels. It might be possible to try to answer the following questions:

- Does a model trained on a small annotated dataset outperform a weakly-supervised model trained on around 40× larger weakly-labelled dataset? – No weakly-supervised models surpassed the fully-supervised BERT model trained on the gold labels.

- Is it worth using such weakly-supervised models, or should people still annotate more texts with gold labels and use fully-supervised classifiers? – This mostly depends heavily on the application and concrete texts that need to be classified. It is challenging to give a concrete answer. Based on the findings in this thesis, the best weakly-supervised approach for Estonian sentiment analysis managed to get an average accuracy of 7.29% less (7.05% worse weighted F1-score) than using solely gold labels for model training (supervised BERT). Considering the labels set by majority voting, the best-performing weakly-supervised model had an average classification accuracy of 11.43% worse (8.79% worse weighted F1-score) than the fully-supervised BERT model.

Creating good quality weak labels is one of the most critical engineering tasks in weakly-supervised text classification. The weakly-supervised models did not surpass or come close to fully-supervised models in terms of performance. One of the main reasons why the weakly-supervised models did not work as well as possible may be the poor quality of labelling functions. Usually, labelling functions in different works (Zhang et al., 2021) are very fine-grained, and there are many of them (more than 10s, up to 100s), and they have high precision and high recall. In this thesis, there were only four labelling functions used. The performance metrics of the label models trained using labelling functions' weak labels were worse compared to other studies. The intention was to have very general labelling functions that would not overfit the specific Valence dataset but instead generally work for different Estonian sentiment analysis datasets. This way, the Postimees dataset could be labelled on a fair basis.

Valence dataset is a complex dataset to train classifiers on in general. The performance of classifiers trained using the Valence dataset is not very high. It might be that the Valence dataset is more challenging to classify than some popular English sentiment analysis datasets like the IMDB, which had the best accuracy of 88.86% using the `MeTaL` (Ratner et al., 2018) label model and `COSINE` (Yu et al., 2021) model. Similarly, the Youtube dataset, which had the best accuracy of 98.00% using Majority Vote and `COSINE` (Yu et al., 2021) models, according to the experiments by Zhang et al. (2021). Valence dataset texts are from daily newspapers and internet comments. That is not as specific of a genre as movie reviews or spam comments. In addition to that, the texts contain some quite specific irony, which might not be very straightforward to learn. The Valence dataset does not contain classical sentiment expressions like "I really love/hate this" but more colloquial and not direct sentiment expressions. To classify some texts, more general knowledge about history and politics may be needed.

Valence dataset's gold labels might not be entirely correct as there are various interpretations to assign labels for some of the texts. Some texts could have been assigned different labels in the opinion of the author of this thesis. As it was possible to see from section 5.6, where incorrectly labelled texts were analysed, some labels could be considered to be incorrect, or some other label would suit some texts much better.

Inter-annotator agreement of sentiment analysis labels. Human labelling was carried out, and the inter-annotator agreement scores turned out to have only moderate agreement. The labels are assigned pretty subjectively. It is very challenging to set ground truth labels as every labeller has their cultural background,

sense of history and politics and personal preferences, which all might influence their decisions. As some texts might be labelled according to one subset of interpretations and some other texts labelled according to another subset of possible interpretations, it is complicated for the classifiers to learn which concrete aspect and interpretation the sentiment analysis is based on.

Using the strongest form of available supervision. A general rule of thumb that can also be seen from this thesis is to use the strongest form of available supervision. If there is an annotated dataset, a fully-supervised model should be trained. If there is a small annotated dataset and a larger unannotated dataset, then semi-supervised approaches should be explored. Weak supervision could be a great tradeoff if it is possible to create labelling functions for the classification task. The labelling functions themselves should leverage as many supervision sources as possible. Unsupervised approaches should be considered when there is no other opportunity to use a stronger form of supervision.

The `WeaSEL` model did not detect the neutral class. The model detected neutral class only when it was trained on the Valence dataset. One possible explanation could be that the only labelling function that detected the neutral class performed quite poorly. The `WeaSEL` model might be more sensitive to class imbalance, or the found hyperparameters were still not optimal.

Future work. There are many different possibilities to expand this work further. One possible way would be to gather a different and much larger dataset from a different sentiment analysis genre, for example, movie or product review texts. Similar models could be trained on the new dataset using enhanced labelling functions. Such models could then be evaluated on the Valence dataset development and test sets or other datasets. It would be interesting to see how the models' performance would change if semi-supervised text classification were used. The small annotated Valence dataset coupled with the much larger unannotated Postimees dataset could lead to some state-of-the-art results for the Valence test set. In addition to weak supervision, there are other possibilities for mitigating the problem of hand-labelling more texts to get performance gain. Examples would include translating English datasets to Estonian to train fully-supervised classifiers and artificially generating more training data using back-translation. One of the most critical expansions could be further developing labelling functions for Estonian sentiment analysis. Finding better weak supervision sources might yield better results.

6 Conclusion

In this Master's Thesis, weakly-supervised text classification models were applied to different Estonian sentiment analysis datasets. The models included supervised and weakly-supervised BERT (EstBERT); weakly-supervised COSINE, and WeaSEL models. Labelling functions were used to create the weak labels using distant supervision and heuristic rules with the MeTaL label model from the Snorkel framework. The supervised BERT model (trained on gold labels) and weakly-supervised BERT, COSINE and WeaSEL models were first trained and evaluated on a small, annotated Estonian sentiment analysis dataset, the Valence dataset to compare the results to previous studies and get a baseline understanding of the models' performance. Around 40× larger unannotated training dataset was created by joining the Valence train set with the Postimees dataset's training set, which was called the Combined dataset. The models were trained again on the Combined dataset, and it was evaluated whether there was a gain in performance. Finally, human labelling was carried out to understand the performance of the models better.

The main model performance evaluation results are the following. On the Valence dataset, the fully-supervised BERT model trained with gold labels outperformed all of the other models. The fully-supervised BERT model achieved a test set classification accuracy of 74.44% (73.82% weighted F1-score) averaged over five runs. The other models did not get a better test set classification accuracy than the MeTaL label model, except the WeaSEL model. On the Combined dataset, all of the models (except the WeaSEL model, which did not output the neutral class) were able to get a better average test classification accuracy than the MeTaL label model. The COSINE model had the best average test classification accuracy of 67.15% (66.77% weighted F1-score). Compared to the COSINE model trained on the Valence dataset, the accuracy was, on average, 2.93% (3.35% higher weighted F1-score) higher, so a larger training dataset did help the model get better performance on the Valence test set.

A thorough analysis was performed using the ground truth labels set by human labellers with a majority vote. For the Postimees test subset, the weakly-supervised COSINE and WeaSEL models had better performance when they were trained on the Combined dataset. However, they did not exceed the accuracy of the fully-supervised BERT model. In the setting where the ambiguous texts were removed, the best-performing model, COSINE trained on the Combined dataset, had an average classification accuracy of 11.43% (8.79% worse weighted F1-score) worse than the best-performing, fully-supervised BERT model.

The aim of this Master's Thesis was to assess the applicability of weakly-supervised methods in Estonian sentiment analysis as an alternative to annotating more data. The fully-supervised models still outperformed the weakly-supervised

models. Nevertheless, there was a significant difference – no labelled training data was needed to train a weakly-supervised model. The lower performance of weakly-supervised models might be caused by the low quality of labelling functions – developing them further might lead to better results. If the performance of the model is a top priority, then fully-supervised models still might be a better choice, and thus texts should be hand-labelled. There is a tradeoff between the model performance and the time needed to hand-label the texts or develop labelling functions. Every text classification task and dataset should be critically evaluated, which approach would be the most sensible. There is a perspective on two activities in future work - hand-labelling additional texts for Estonian sentiment analysis or developing better labelling functions for the Estonian language.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(1):993–1022, 2003. ISSN 15324435. doi: 10.5555/944919.944937. URL https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Salva Rühling Cachay, Benedikt Boecking, and Artur Dubrawski. End-to-End Weak Supervision. *Advances in Neural Information Processing Systems*, 34, 7 2021. doi: 10.48550/ARXIV.2107.02233. URL http://arxiv.org/abs/2107.02233.

Ming Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *23rd AAAI Conference on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conference, AAAI-08/IAAI-08*, volume 2, pages 830–835, 2008. URL https://www.aaai.org/Papers/AAAI/2008/AAAI08-132.pdf.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. Dataless text classification with descriptive LDA. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, volume 3 of *AAAI'15*, pages 2224–2231, 2015. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/download/9524/9546.

Zewei Chu, Karl Stratos, and Kevin Gimpel. Unsupervised Label Refinement Improves Dataless Text Classification. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4165–4178, 12 2020. URL http://arxiv.org/abs/2012.04194.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 4 1960. ISSN 0013-1644. doi: 10.1177/001316446002000104. URL http://journals.sagepub.com/doi/10.1177/001316446002000104.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. URL https://rss.onlinelibrary.wiley.com/doi/epdf/10.1111/j.2517-6161.1977.tb01600.x.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North*, 1:4171–4186, 10 2019. URL http://arxiv.org/abs/1810.04805.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv preprint arXiv:2002.06305*, 2 2020. URL https://arxiv.org/abs/2002.06305.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. ISSN 0033-2909. doi: 10.1037/h0031619. URL http://content.apa.org/journals/bul/76/5/378.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 7, pages 1606–1611, 2007. URL https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-259.pdf.

Swapnil Hingmire and Sutanu Chakraborti. Sprinkling Topics for Weakly Supervised Text Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 55–60, Stroudsburg, PA, USA, 2014a. Association for Computational Linguistics. doi: 10.3115/v1/P14-2010. URL http://aclweb.org/anthology/P14-2010.

Swapnil Hingmire and Sutanu Chakraborti. Topic labeled text classification. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 385–394, New York, NY, USA, 7 2014b. ACM. ISBN 9781450322577. doi: 10.1145/2600428.2609565. URL https://dl.acm.org/doi/10.1145/2600428.2609565.

Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and Sutanu Chakraborti. Document classification by topic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 877–880, New York, NY, USA, 7 2013. ACM. ISBN 9781450320344. doi: 10.1145/2484028.2484140. URL https://dl.acm.org/doi/10.1145/2484028.2484140.

Daniel Jurafsky and James H Martin. *Speech and Language Processing (3rd ed. draft)*. 2021. URL https://web.stanford.edu/~jurafsky/slp3/ed3book_jan122022.pdf.

Heiki Jaan Kaalep, Kadri Muischnek, Kristel Uiboaed, and Kaarel Veskis. The Estonian reference corpus: Its composition and morphology-aware user interface. In *Proceedings of the 2010 conference on Human Language Technologies–The Baltic Perspective: Proceedings of the Fourth International Conference Baltic HLT*, volume 219, pages 143–146. Human Language Technologies–The Baltic

Perspective, 8 2010. doi: 10.3233/978-1-60750-641-6-143. URL https://doi.org/10.3233/978-1-60750-641-6-143.

Jelena Kallas and Kristina Koppel. Estonian National Corpus 2017, 2018. URL https://doi.org/10.15155/3-00-0000-0000-0000-071E7L.

Claudia Kittask, Kirill Milintsevich, and Kairit Sirts. Evaluating Multilingual BERT for Estonian. In *Frontiers in Artificial Intelligence and Applications*, volume 328. IOS Press, 9 2020. doi: 10.3233/FAIA200597. URL http://ebooks.iospress.nl/doi/10.3233/FAIA200597.

Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown. Text Classification Algorithms: A Survey. *Information*, 10(4):150, 4 2019. ISSN 2078-2489. doi: 10.3390/info10040150. URL https://www.mdpi.com/2078-2489/10/4/150.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*, 9 2019. URL https://arxiv.org/abs/1909.11942.

Sven Laur, Siim Orasmaa, Dage Särg, and Paul Tammo. EstNLTK 1.6: Remastered estonian NLP pipeline. In *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.884.

Changchun Li, Ximing Li, and Jihong Ouyang. Semi-Supervised Text Classification with Balanced Deep Representation Distributions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5044–5053, Stroudsburg, PA, USA, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.391. URL https://aclanthology.org/2021.acl-long.391.

Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. Effective Document Labeling with Very Few Seed Words. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, volume 24-28-October-2016, pages 85–94, New York, NY, USA, 10 2016. ACM. ISBN 9781450340731. doi: 10.1145/2983323.2983721. URL https://dl.acm.org/doi/10.1145/2983323.2983721.

Keqian Li, Hanwen Zha, Yu Su, and Xifeng Yan. Unsupervised Neural Categorization for Scientific Publications. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 37–45. Society for Industrial and Applied

Mathematics, Philadelphia, PA, 5 2018a. doi: 10.1137/1.9781611975321.5. URL https://epubs.siam.org/doi/10.1137/1.9781611975321.5.

Ximing Li, Changchun Li, Jinjin Chi, Jihong Ouyang, and Chenliang Li. Dataless Text Classification: A Topic Modeling Approach with Document Manifold. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 973–982, New York, NY, USA, 10 2018b. ACM. ISBN 9781450360142. doi: 10.1145/3269206.3271671. URL https://dl.acm.org/doi/10.1145/3269206.3271671.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv*, 7 2019. doi: 10.48550/ARXIV.1907.11692. URL http://arxiv.org/abs/1907.11692.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. URL https://ieeexplore.ieee.org/abstract/document/1056489.

Andrew McCallum and Kamal Nigam. Text Classification by Bootstrapping with Keywords, EM and Shrinkage. *In ACL99 - Workshop for Unsupervised Learning in Natural Language Processing*, 1999. URL https://aclanthology.org/W99-0908.

Dheeraj Mekala and Jingbo Shang. Contextualized Weak Supervision for Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Stroudsburg, PA, USA, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.30. URL https://www.aclweb.org/anthology/2020.acl-main.30.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-Supervised Neural Text Classification. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992, 9 2018. doi: 10.1145/3269206.3271737. URL http://dx.doi.org/10.1145/3269206.3271737.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text Classification Using Label Names Only: A Language Model Self-Training Approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017, Stroudsburg, PA, USA, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.724. URL https://www.aclweb.org/anthology/2020.emnlp-main.724.

Kadri Muischnek. Estonian Reference Corpus, 11 2011. URL https://doi.org/10.15155/9-00-0000-0000-0000-0017FL.

Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103–134, 2000. ISSN 08856125. doi: 10.1023/a:1007692713085. URL https://link.springer.com/content/pdf/10.1023/A:1007692713085.pdf.

Birgitta Ojamaa, Päivi Kristiina Jokinen, and Kadri Muischenk. Sentiment analysis on conversational texts. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 233–237, Vilnius, Lithuania, 5 2015. Linköping University Electronic Press, Sweden. URL https://aclanthology.org/W15-1829.

Hille Pajupuu, Krista Kerge, and Rene Altrov. Detecting emotional valence of text by using a small dictionary. In Izaskun Elorza, Ovidi Carbonell i Cortés, Reyes Albarrán, Blanca García Riaza, and Miriam Pérez-Veneros, editors, *Empiricism and analytical tools for 21 Century applied linguistics: selected papers from the XXIX International Conference of the Spanish Association of Applied Linguistics (AESLA)*, volume 185, pages 229–241, Salamanca, 10 2012. Universidad de Salamanca. ISBN 978-84-9012-154-2. URL https://www.etis.ee/File/DownloadPublic/4d394f3d-d350-4cbd-bfa3-86b049907102?name=Fail_Pajupuu_Kerge_Altrov_2012.pdf&type=application%2Fpdf.

Hille Pajupuu, Rene Altrov, and Jaan Pajupuu. Identifying Polarity in Different Text Types. *Folklore: Electronic Journal of Folklore*, 64:125–142, 6 2016. ISSN 14060957. doi: 10.7592/FEJF2016.64.polarity. URL http://dx.doi.org/10.7592/FEJF2016.64.polarity.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing Neural Networks by Penalizing Confident Output Distributions. *arXiv*, 1 2017. URL http://arxiv.org/abs/1701.06548.

Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data Programming: Creating Large Training Sets, Quickly. *Advances in Neural Information Processing Systems*, 5 2016. URL http://arxiv.org/abs/1605.07723.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid Training Data Creationwith Weak Supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282, 11 2017. ISSN

2150-8097. doi: 10.14778/3157794.3157797. URL https://dl.acm.org/doi/10.14778/3157794.3157797.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. Training Complex Models with Multi-Task Weak Supervision. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 4763–4771, 10 2018. URL http://arxiv.org/abs/1810.02840.

Yangqiu Song and Dan Roth. On dataless hierarchical text classification. In *28th AAAI Conference on Artificial Intelligence, AAAI 2014, 26th Innovative Applications of Artificial Intelligence Conference, IAAI 2014 and the 5th Symposium on Educational Advances in Artificial Intelligence, EAAI 2014*, volume 28, pages 1579–1585, 2014. URL https://ojs.aaai.org/index.php/AAAI/article/download/8938/8797.

Karen Sparck Jones. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21, 1 1972. ISSN 0022-0418. doi: 10.1108/eb026526. URL https://www.emerald.com/insight/content/doi/10.1108/eb026526/full/html.

Dominik Stammbach and Elliott Ash. DocSCAN: Unsupervised Text Classification via Learning from Neighbors. *Center for Law & Economics Working Paper Series*, 2021(08), 5 2021. doi: 10.3929/ethz-b-000484681. URL https://doi.org/10.3929/ethz-b-000484681.

Hasan Tanvir, Claudia Kittask, Sandra Eiche, and Kairit Sirts. EstBERT: A pretrained language-specific BERT for Estonian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 11–19, Reykjavik, Iceland (Online), May 31–2 June 2021. Linköping University Electronic Press, Sweden. URL https://aclanthology.org/2021.nodalida-main.2.

University of Tartu. UT Rocket, 2018. URL https://share.neic.no/#/marketplace-public-offering/c8107e145e0d41f7a016b72825072287/.

Siim Kaspar Uustalu. Automated Detection and Sentiment Analysis of Registered Entity Mentions in Estonian Language News Media. Master's thesis, Tallinn University of Technology, Tallinn, 5 2019. URL https://digikogu.taltech.ee/en/Download/778ff477-e065-4f13-ba62-5c30d230dd4c.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances*

in Neural Information Processing Systems, 2017-December, 6 2017. URL http://arxiv.org/abs/1706.03762.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, 9 2016. URL http://arxiv.org/abs/1609.08144.

Yi Yang, Hongan Wang, Jiaqi Zhu, Yunkun Wu, Kailong Jiang, Wenli Guo, and Wandong Shi. Dataless Short Text Classification Based on Biterm Topic Model and Word Embeddings. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, volume 2021-January, pages 3969–3975, California, 7 2020. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-6-5. doi: 10.24963/ijcai.2020/549. URL https://www.ijcai.org/proceedings/2020/549.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1404. URL https://aclanthology.org/D19-1404.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1063–1077, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.84. URL https://aclanthology.org/2021.naacl-main.84.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. WRENCH: A Comprehensive Benchmark for Weak Supervision. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1, 9 2021. URL http://arxiv.org/abs/2109.11377.

Lei Zhang, Shuai Wang, and Bing Liu. Deep Learning for Sentiment Analysis : A Survey. *WIREs Data Mining and Knowledge Discovery*, 8(4), 1 2018. URL http://arxiv.org/abs/1801.07883.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 6 2015. doi: 10.1109/ICCV.2015.11. URL https://doi.org/10.1109/ICCV.2015.11.

Appendix

I. Labelling Instructions and File Example

The annotators were sent an email with an attached Microsoft Excel XSLX file. They were given the following instructions (in Estonian only) and the following file.

Labelling Instructions

> Palun märgendage XSLX-failis toodud tekstid esimeses tühjas veerus (assigned label) kolme klassi: positiivne, negatiivne või neutraalne. Valige Teie hinnangul kõige domineerivam teksti meelsus. Juhul, kui kõhklete kahe klassi vahel ja kindel domineeriv meelsus puudub, valige Teie arvates tugevam meelsus ning järgmisesse veergu (is ambiguous) kirjutage "jah". Samuti võite viimasesse veergu (comment) kirjutada, miks niimoodi otsustasite või mis tekitab kõhklust. Samuti on XSLX-failis toodud näited, kuidas on faili korrektne täita.
>
> Näited:
>
> Näide positiivse meelsusega tekstist: "Koht, mis varem ei olnud püha, võib selleks saada. Kui istutame tammikud, muudame need kohad pühaks. Hoolitseme ka selle eest, et tammikutes kasvaks kaunis kask ja püha pihlakas, et kaugete esivanemate vaimud end seal hästi tunneksid."
>
> Negatiivse meelsusega tekst: "Tabati ka üks kriminaalses joobes sõidukijuht. See juhtus pühapäeva öösel kella 4 ajal, kui Viljandis Lääne tänaval peeti kinni sõiduauto BMW, mille roolis oli 21-aastane noormees. Tema suhtes alustati kriminaalmenetlus"
>
> Neutraalse meelsusega tekst: "Peaaegu samasugune nägi pööning välja märtsis, kui kunstnik oli sinna üles seadnud “Asjade” esimese osa. Vahepealse kuue kuu jooksul on katusealune ja seda külastanud vaatajad osa saanud suurtest muudatustest."
>
> Vastuolulise meelsusega tekst: "Uuringust tuli välja, et ligi pooled inimesed ei kavatse enam Eestisse tagasi tulla, kuid paljud vastajad tunnistasid, et kui Eestis oleks neil rohkem väljakutseid ja huvitav töö, siis kaaluksid nad tagasitulemist."
>
> Annoteerimine võtab aega umbes 45 minutit.

Please label the texts in the XSLX file in the first empty column (assigned label) into one of the three classes – positive, negative, or neutral. Pick the sentiment that is, in your opinion, the most dominating. If you are uncertain about choosing one or the other class and there is no dominating sentiment, pick the sentiment that, in your opinion, is stronger and in the next column (is ambiguous), write "jah". In addition, you may write in the last column (comment) why you made such a decision or what is causing uncertainty. There are also examples in the XSLX file of how the file should be filled in. Examples: Example of a text with a positive sentiment: "A place that previously was not holy can become like that. We can make it holy ourselves by planting an oak forest. Moreover, let us take care that the oak forest also features the beautiful birch and the protective rowan, just to make the distant ancestral spirits feel good."

Text with the negative sentiment: "Also, a criminally intoxicated driver was apprehended. It happened at 4 o'clock Sunday morning that a BMW driven by a 21-year old was stopped on Lääne St. in Viljandi. Criminal charges were filed."

Text with a neutral sentiment: "The attic looked almost the same in March, just after the artist had set up the first part of the "Things". During the six months passed, the attic and its visitors were exposed to some considerable changes."

Text with an ambiguous sentiment: "It was found out from the study that around half of the people do not anticipate returning to Estonia, but many participants admitted that if there were more challenges and an interesting job, they would consider returning."

Annotating takes approximately 45 minutes.

File Example

An example of the Excel XSLX file sent to human labellers can be seen in Figure 9. Rows 2–5 show examples, and starting from row 6, labellers are asked to label 100 texts.

1	id	text	assigned label	is ambiguous (only assigned in this column)	comment	Please label the following texts' sentiment with one of the following tags: 0 (negative), 1 (positive), 2 (neutral).
2	-	Koht, mis varem ei olnud püha, võib selleks saada. Kui istutame tammikud, muudame need kohad pühaks. Hoolitseme ka selle eest, et tammikutes kasvaks kaunis kask ja püha pihlakas, et kaugete esivanemate vaimud end seal hästi tunneksid	1		domineeriv meelsus on positiivne	
3	-	Tabati ka üks kriminaalses joobes sõidukijuht. See juhtus pühapäeva öösel kella 4 ajal, kui Viljandis Lääne tänaval peeti kinni sõiduauto BMW, mille roolis oli 21-aastane noormees. Tema suhtes alustati kriminaalmenetlus	0			
4	-	Peaaegu samasugune nägi pööning välja märtsis, kui kunstnik oli sinna üles seadnud "Asjade" esimese osa. Vahepealse kuue kuu jooksul on katusealune ja seda külastanud vaatajad osa saanud suurtest muudatustest.	2		pole ei positiivne ega negatiivne otseselt	
5	-	Uuringust tuli välja, et ligi pooled inimesed ei kavatse enam Eestisse tagasi tulla, kuid paljud vastajad tunnistasid, et kui Eestis oleks neil rohkem väljakutseid ja huvitav töö, siis kaaluksid nad tagasitulemist.	2	jah	esimeses pooles negatiivne meelsus, teises jällegi positiivne	
6	29	Hoopis vähem mõeldakse sellele , et mainitud leping vajab ka ratifitseerimist . Sellele peab mõtlema see minister , kes parlamendi ees ettekande esitab . Eesti oludes on selleks ministriks välisminister . Ning leping vajab Riigikogus heakskiitmiseks kaht kolmandikku parlamendiliikmete häältest .				
7	66	Tegelikult on Venemaa meelest suhted normaalsed siis , kui Eesti on sisuliselt nõustunud nullvariandiga kodakondsuse küsimustes . Selles küsimuses orienteerub Euroopa pigem Venemaa kui Eesti lõppjäreldusele . Sellele viitab viimane ÜRO aruanne inimõiguste olukorrast .				
8	72	Käibelolevate Eesti Posti eeskirjade kohaselt tuleb panderoll siduda nööriga . Olen Võru posti kaudu saatnud USAsse mitmeid selliseid karvase ja takuse nööriga (millist omal ajal kasutati kolhoosides linapeode sidumiseks) seotud panderolle .				
9	74	Millisel seisukohal on Eesti Posti juhtkond ? Kas ta nõustub minu ettepanekuga või leiab , et antud küsimuses on süüdi jällegi kohalikud ametimehed , nagu see tavaks on saanud .				

Figure 9. An example of the Excel file sent to human labellers.

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Andreas Pung**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Weakly-Supervised Text Classification for Estonian Sentiment Analysis,

   supervised by Kairit Sirts, Ph.D.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Andreas Pung
17/05/2022