

# A Weak-supervision Method for Automating Training Set Creation in Multi-domain Aspect Sentiment Classification

Massimo Ruffolo<sup>1,2</sup>  and Francesco Visalli<sup>1</sup>

<sup>1</sup>High Performance Computing and Networking Institute of the National Research Council (ICAR-CNR),  
Via Pietro Bucci 8/9C, Rende (CS), 87036, Italy

<sup>2</sup>Altilia.ai, Technest - University of Calabria, Piazza Vermicelli, Rende (CS), 87036, Italy

**Keywords:** Weak-supervision, Data Programming, Deep Learning, Aspect Based Sentiment Analysis, Transformers, Natural Language Processing.

**Abstract:** Aspect Based Sentiment Analysis (ABSA) is receiving growing attention from the research community because it has applications in several real world use cases. To train deep learning models for ABSA in vertical domains may result a laborious process requiring a significant human effort in creating proper training sets. In this work we present initial studies regarding the definition of an easy-to-use, flexible, and reusable weakly-supervised method for the Aspect Sentence Classification task of ABSA. Our method mainly consists in a process where templates of Labeling Functions automatically annotate sentences, and then the generative model of Snorkel constructs a probabilistic training set. In order to test effectiveness and applicability of our method we trained machine learning models where the loss function is informed about the probabilistic nature of the labels. In particular, we fine-tuned BERT models on two famous disjoint SemEval datasets related to laptops and restaurants.

## 1 INTRODUCTION

Aspect Based Sentiment Analysis (ABSA) is a Natural Language Processing (NLP) problem that is receiving growing attention from the research community because it can be productively used in many different real world use cases (Hu and Liu, 2004). For example, ABSA enables extracting relevant features, along with buyers opinions, from product reviews available online.

The **battery** gets so **hot** it is scary.

All the **food** was **hot** tasty.


Figure 1: Examples of aspects and related sentiments.

ABSA comes in two mainly variants (Pontiki et al., 2014), one of these provides for two subtasks which are Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). Figure 1 shows two sentences extracted from laptops and restaurants domains, respectively. Aspects are highlighted in blue, AE takes

care of retrieving them from sentences. Instead, ASC identifies the polarity of terms referred to the aspect and that express a sentiment/opinion. ABSA is a difficult problem to address in multiple domains because an opinion term that could be positive for a domain may be not for another.

For example, in Figure 1, the term “hot” expresses a *Negative* opinion about the aspect “battery”, whereas the same term referred to the aspect “food” assumes a *Positive* connotation. A battery that gets hot is not desirable, while a hot tasty food is good.

In last years ABSA methods based on deep learning are becoming mainstream. Deep learning automates the feature engineering process. Since 2012, when AlexNet (Krizhevsky et al., 2012) won the ImageNet Large Scale Visual Recognition Competition<sup>1</sup>, deep neural network architectures have contaminated also the NLP area, becoming the state of the art for many tasks in this field (Devlin et al., 2019; Yang et al., 2019) comprised ABSA (Xu et al., 2019; Rietzler et al., 2019). Despite such a success, developing enterprise-grade deep learning-based applications still pose many challenges. In particular, deep

<sup>a</sup>  <https://orcid.org/0000-0002-4094-4810>

<sup>1</sup><http://www.image-net.org/challenges/LSVRC/>

learning models are often hungry for data, as they are rich of parameters. Leveraging supervised learning methods to train these models requires a large amount of annotated examples. Such training sets are enormously expensive to create, especially when domain expertise is required. Moreover, the specifications of a system often change, requiring the re-labeling of the datasets. Therefore, it is not always possible to rely on subject matter experts for labeling the data. This is one of the most costly bottlenecks to a wide and pervasive adoption of deep learning methods in real world use cases. Hence, alleviating the cost of human annotation is a major issue in supervised learning.

To tackle this problem, vary approaches such as: transfer learning, semi-supervised learning, where both unsupervised and supervised learning methods are exploited, and weak supervision have been proposed. Transfer learning methods such as (Ding et al., 2017; Wang and Pan, 2018) rely on the fact that a model trained on a specific domain (source) can be exploited to do the same task on another domain (target), thus reducing the need for labeled data in the target domain.

One of the most important example of semi-supervised learning, recently appeared in literature, is Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) that strongly reduces the volume of needed labeled data. BERT mainly is a model that provides strong contextual word embeddings learned in an unsupervised way by training on large text corpus. But BERT is also a generic deep learning architecture for many NLP tasks because it can be fine-tuned in a supervised way in order to learn various down-stream NLP tasks. The idea behind fine-tuning is that most of the features have already been learned and the model just needs to be specialized for the specific NLP task. This way, fine-tuning BERT for a specific NLP problem, such as ABSA, requires much less annotated data than learning the entire task from scratch.

Weak supervision simplifies the annotation process in order to make it more automatic and scalable, even though less accurate and noisier. Weak supervised methods rely on several different data annotation techniques such as the use of heuristics, distant learning, pattern matching, weak classifiers and so on. Recently, (Ratner et al., 2016) proposed data programming as a paradigm for semi-automatic datasets labeling, and Snorkel (Ratner et al., 2017) the system that implements it. Data programming is based on the concept of Labeling Functions (LFs), where LFs are procedures that automatically assign labels to data on the base of domain knowledge embedded in form of annotation rules. (Bach et al., 2019) at Google ex-

tended Snorkel in order to achieve better scalability and knowledge base re-usability for enterprise-grade training sets labeling.

In this paper we propose a weakly-supervised approach to the ASC task in ABSA problems. In our approach we leverage BERT fine-tuning method for sentiment classification as in (Xu et al., 2019), and Snorkel (Ratner et al., 2017) to apply data programming principles to reviews. The main contributions of this work are:

- The definition of a set of easy-to-use, flexible, and reusable LFs templates that make viable the weakly-supervised method for the ASC task;
- The experimental evaluation of the generality, effectiveness, and robustness of the weakly-supervised method proposed in (Ratner et al., 2017) when applied to complex problems like the ASC task.

In order to prove the flexibility and the robustness of our approach we tested it on two disjoint domains, laptops and restaurants. In particular we used the datasets of SemEval task 4 subtask 2 (Pontiki et al., 2014). Finally, we compared the obtained results with those of supervised learning methods. Results appear remarkable for a weakly supervised system. They show that our approach can be used for practical purpose on multiple domains.

The rest of this paper is organized as follows: in Section 2 we introduce a list of ABSA works that try to reduce the need for human effort; in Section 3 we present our method in terms of LFs template we have defined; in Section 4 the experiments carried out and the results obtained are shown and discussed; finally, in Section 5 conclusions are drawn and the future work is presented.

## 2 RELATED WORK

While there is a large corpus of scientific papers related to the ABSA problem, to the best of our knowledge there are very few works that propose the application of weak-supervision methods to ABSA. (Pablos et al., 2014) uses some variations of (Qiu et al., 2009) and (Qiu et al., 2011) to perform AE and ASC.

In (Pablos et al., 2015) the AE task is done by bootstrapping a list of candidate domain aspect terms and using them to annotate the reviews of the same domain. The polarity detection is performed using a polarity lexicon exploiting the Word2Vec model (Mikolov et al., 2013) for each domain (however the task is a bit different from ASC, they classify Entity-Attribute pair, where Entity and Attribute belong to

predefined lists, e.g. food, price, location for Entity and food-price, food-quality for Attribute).

(Pablos et al., 2018) presents a fully “almost unsupervised” ABSA system. Starting from a customer reviews dataset and a few words list of aspects they extract a list of words per aspect and two lists of positive and negative words for every selected aspect. It is based on a topic modelling approach combined with continuous word embeddings and a Maximum Entropy classifier.

(Purpura et al., 2018) performs the AE phase with a topic modeling technique called Non-negative Matrix Factorization. It allows the user to embed a list of seed words to guide the algorithm towards more significant topic definition. The ASC is done by using a list of positive and negative words, with a few sentiment terms for each topic. This list is then extended with the Word2Vec model (Mikolov et al., 2013).

In (Pereg et al., 2019) aspect and opinion lexicons are extracted from an unsupervised dataset belonging to the same domain as the target domain. The process is initialized with a seed lexicon of generic opinion terms. New aspect and opinion terms are extracted by using the dependency rules proposed in (Qiu et al., 2011). The opinion lexicon is then filtered and scored while the aspect lexicon can be modified by hand in a weak supervised manner. ASC is performed on a target domain by detecting a direct or second-order dependency relation of any type between aspect-opinion pairs.

All related works are specifically designed for ABSA and leverage low level techniques. We propose a more high level approach, easy to extend, that simplifies and automates the annotation of sentiment terminology making the ASC task easily applicable in multiple domains. Moreover, the nature of our approach allows using any discriminative model. In particular, it allows taking advantage of deep learning models that have reached the state of the art on NLP tasks.

### 3 WEAK SUPERVISION FOR ASC

Our intent is to address the ASC task in a weakly-supervised way. For the scopes of this paper we assume that the AE task has already been done. So, we have as input a dataset composed of sentence-aspect pairs. This kind of dataset can be built in many ways. Sentences can be easily extracted from reviews with a sentence splitter. Aspects can also be retrieved with different approaches, see (Pablos et al., 2018; Pereg et al., 2019) for some examples.

Our weak-supervision method, specifically de-

signed for the ASC task of the ABSA problem, is grounded on data programming (Ratner et al., 2016) that is a weak-supervised paradigm based on the concept of Labeling Functions (LFs), where LFs are procedures, designed by data scientists and/or subject matter experts, that automatically assign labels to data on the base of domain knowledge embedded in form of annotation rules.

More in detail, our method consists in a set of predefined, easy-to-use, and flexible LFs capable of automatically assigning a sentiment to sentence-aspect pairs. The method is based on the ideas that: (i) it must require minimum NLP knowledge to the user, and (ii) it must be reusable in multiple domains with minimal effort for domain adaptation.

One of the most important characteristic of data programming is that LFs are noisy (e.g. different LFs may label same data in different ways, LFs can label false positive examples). In this paper, in order to deal with the ASC task of ABSA problems by data programming, we used the Snorkel system (Ratner et al., 2017) that enables handling the entire life-cycle of data programming tasks. Once LFs have been written, Snorkel applies them to data and automatically learns a generative model over these LFs which estimates their accuracy and correlations, with no ground-truth data needed. It is noteworthy that Snorkel applies LFs as a generative process which automatically de-noises the resulting dataset by learning the accuracy of the LFs along with their correlation structure. Thus, the output of this process is a training set composed of probabilistic labels that can be used as input for deep learning algorithms that have to use a noise-aware loss function.

In our weak-supervised method LFs take as input pairs having the form  $\langle s; a \rangle$  where  $s$  is a sentence and  $a$  is an aspect, and return triples having the form  $\langle s; a; l \rangle$  where  $s$  and  $a$  are the sentence and the aspect in input respectively, and  $l \in \{Positive, Negative, Neutral\} \cup \{Abstain\}$  is the label in output. In fact, a LF can choose to abstain if it doesn't have sufficient information to assign a sentiment. The following pseudo-code shows the structure of a LF in our method.

A fundamental step of the annotation process in each LF is to chunk the text in input. So, every sentence in input to a LF is analyzed by  $chunker(s)$ . To implement  $chunker(s)$  we leverage the Stanford CoreNLP Parser (Manning et al., 2014).  $chunker(s)$  returns a list of chunks belonging to two different types:  $\overline{NP}$  and  $\overline{VP}$ . To assign tokens of a sentence to  $\overline{NP}$  and  $\overline{VP}$  we use the following strategy in traversing the parse tree: all the tokens around a  $NP$  tag are assigned to a  $\overline{NP}$  chunk, until we meet a  $VP$  tag. Considering the encountered  $VP$  tag all the

tokens around it are assigned to a  $\overline{VP}$  chunk until we meet another  $NP$  tag and so on.

```

INPUT: <s;a>
OUTPUT: <s;a;l>

BEGIN METHOD:

  C := chunker(s)

  if a belongs to a chunk in C and
  a is the longest aspect for that chunk:
    D := dependency_parser(C, a)
    L := labeling(D)

    return <s;a;l>

  return <s;a;Abstain>

END METHOD

```

It is noteworthy that if tokens of an aspect  $a$ , of a given pair  $\langle s;a \rangle$ , span over multiple  $\overline{NP}$  or  $\overline{VP}$  chunks the LF assigns the *Abstain* label to the pair. Moreover, if in a chunk in  $C$  there are more than one aspect the LF considers the aspect having maximum number of tokens to try to assign the polarity label and assigns the *Abstain* label to other aspects in the same chunk. This behavior avoids introducing too much noise in the labeling process.

When conditions about the aspect  $a$  are verified the LF calls the `dependency_parser(C, a)` method. As dependency parser we use the Stanford CoreNLP (Manning et al., 2014). `dependency_parser(C, a)` assigns to  $D$  the chunk that contains the aspect  $a$  and all those chunks that have a direct parsing dependency of some kind with it.

The method `labeling(D)` assigns a label  $l$  to the pair  $\langle s;a \rangle$  by evaluating sentiments of chunks in  $D$ . To compute sentiment polarity of chunks in  $D$  we adopt already trained external sentiment analyzers. In particular, we use Stanford CoreNLP (Manning et al., 2014), TextBlob<sup>2</sup>, NLTK (Bird, 2006), and Pattern<sup>3</sup>. We adopt two strategies in assigning labels. In the first strategy we compute the polarity of each single chunk in  $D$ . If all chunks have the same polarity the method returns the corresponding label. If chunks have mixed *Positive* and *Negative* polarity the method returns the *Abstain* label. When there are *Neutral* chunks mixed with at least one *Positive* chunk the method returns the *Positive* label, and the same happens for mixed *Neutral* and *Negative* chunks. The second strategy consists in appending chunks in  $D$  to compute the global polarity of the resulting text

<sup>2</sup><https://textblob.readthedocs.io/>

<sup>3</sup><https://www.clips.uantwerpen.be/pages/pattern-en/>

string. In this case the method simply returns the polarity  $l$  where  $l \in \{Positive, Negative, Neutral\}$ . Considering the two different strategies and the four sentiment analyzers the approach includes a total of 8 LFs.

It is noteworthy that these labeling function templates are conceptually simple and powerful because they enable everyone to reuse existing knowledge embedded in already available NLP tools to create new training sets. Table 1 and Table 2 show statistics about the LFs when applied to laptops and restaurants datasets, respectively. In particular, columns of the tables represent coverage, overlaps, conflicts, and empirical accuracy of LFs when they are executed to a small number (150) of manually labeled examples called dev set. Rows in the tables correspond to the eight LFs we defined by using NLP tools and strategies described above, where each row of the tables contains values computed for a specific labeling function.

Experiments on laptops in Table 1 show that LFs have about 77% of coverage and 50% of empirical accuracy, while Table 2 shows that restaurants have a coverage of about 69% and empirical accuracy of 50%. Results on coverage and empirical accuracy suggest that defined LFs work properly and can be used to annotate the two datasets.

The result of the labeling process is a matrix of labels  $\Lambda \in (\{Positive, Negative, Neutral\} \cup \{Abstain\})^{m \times n}$ , where  $m$  is the cardinality of the training set and  $n$  is the number of LFs. This matrix is the input of the Snorkel generative model. Such model produces a list of probabilistic training labels  $\tilde{Y} = (\tilde{y}_1, \dots, \tilde{y}_m)$ , where each  $\tilde{y}_i \in \mathbb{R}^3$  is a probability distribution over the classes  $\{Positive, Negative, Neutral\}$ . This probabilistic dataset is the input of discriminative models that use noise-aware loss functions.

## 4 EXPERIMENTS

In this section we describe experiments aiming to verify effectiveness and robustness of our weak-supervision method for the ASC task. To carry out experiments we used discriminative models proposed in (Xu et al., 2019). Such models are obtained by post-training BERT (Devlin et al., 2019) a deep learning architecture based on Transformers (Vaswani et al., 2017). BERT is pre-trained on Wikipedia and BooksCorpus dataset (Zhu et al., 2015) and is currently widely used in many models that reach the state of the art in several NLP tasks.

Post-training enables injecting into BERT the missing domain knowledge. (Xu et al., 2019) shows



Table 1: LFs application stats on laptops domain.

	Coverage	Overlaps	Conflicts	Emp. Acc.
$LF_{(StanfordCoreNLP, FirstStrategy)}$	0.7667	0.7667	0.5267	0.5478
$LF_{(StanfordCoreNLP, SecondStrategy)}$	0.7933	0.7933	0.5333	0.5042
$LF_{(TextBlob, FirstStrategy)}$	0.7400	0.7400	0.4867	0.4775
$LF_{(TextBlob, SecondStrategy)}$	0.7933	0.7933	0.5333	0.5042
$LF_{(NLTK, FirstStrategy)}$	0.7533	0.7533	0.4933	0.4956
$LF_{(NLTK, SecondStrategy)}$	0.7933	0.7933	0.5333	0.4874
$LF_{(Pattern.en, FirstStrategy)}$	0.7467	0.7467	0.4933	0.4821
$LF_{(Pattern.en, SecondStrategy)}$	0.7933	0.7933	0.5333	0.4790

Table 2: LFs application stats on restaurants domain.

	Coverage	Overlaps	Conflicts	Emp. Acc.
$LF_{(StanfordCoreNLP, FirstStrategy)}$	0.6200	0.6200	0.3867	0.4946
$LF_{(StanfordCoreNLP, SecondStrategy)}$	0.6800	0.6800	0.4200	0.5882
$LF_{(TextBlob, FirstStrategy)}$	0.6667	0.6667	0.4067	0.5100
$LF_{(TextBlob, SecondStrategy)}$	0.6800	0.6800	0.4200	0.5098
$LF_{(NLTK, FirstStrategy)}$	0.6667	0.6667	0.4133	0.5000
$LF_{(NLTK, SecondStrategy)}$	0.6800	0.6800	0.4200	0.4098
$LF_{(Pattern.en, FirstStrategy)}$	0.6667	0.6667	0.4067	0.5100
$LF_{(Pattern.en, SecondStrategy)}$	0.6800	0.6800	0.4200	0.5000

that post-training the model on a specific domain contributes to performances improvement. The model resulting from the post-training is finally fine-tuned for the down-stream task. Post-training is the foundation for state of the art ASC methods (Rietzler et al., 2019).

Figure 2 shows the input and the output of BERT for the ASC task. First of all the sentence and the aspect in input are tokenized by the WordPiece algorithm (Wu et al., 2016). Hence,  $q_1, \dots, q_m$  is the embedding of an input aspect with  $m$  tokens and  $d_1, \dots, d_n$  is the embedding of an input sentence with  $n$  tokens.  $[CLS]$  and  $[SEP]$  are two special tokens.  $[CLS]$  is used for classification problem and  $h[CLS]$  is the aspect-aware representation of the whole input through BERT (for further detail see (Devlin et al., 2019)). The  $[SEP]$  token is used to separate two different inputs.

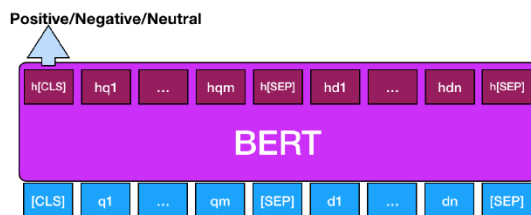


Figure 2: BERT input/output for the ASC task.

Discriminative models like BERT have to be informed about the probabilistic nature of the training set resulting from our weak-supervision method. Hence, we use a noise-aware loss function. The cross-entropy function fits perfectly for this purpose,

because it measures the discrepancy between a true probability distribution and an estimated distribution ( $p$  and  $q$  respectively in Equation 1). In general, the estimated distribution is the output of a classifier, while the true distribution is usually a one hot vector, where the bit set to one indicates which class the training example belongs to. In the probabilistic world of data programming,  $p$  is a probability distribution over the classes (i.e. the output of the Snorkel generative model for a training example).

$$H(p, q) = - \sum_{x \in X} p(x) \log q(x) \quad (1)$$

## 4.1 Dataset

In order to test the robustness of our method and to compare the results with those of supervised algorithms, we chose two datasets belonging to SemEval task 4 subtask 2 (Pontiki et al., 2014) (SemEval from now), and coming from disjoint domains, i.e. laptops and restaurants.

Each dataset is already split in training, development, and test set. The number of examples for each set is shown in Table 3. Each partition was hand-labeled by subject matter experts with labels within the set  $\{Positive, Negative, Neutral\}$ .

In our experiments, we replaced original annotations in the training sets with probability distributions computed by Snorkel generative models. We used the dev set for tuning the LFs and the hyper-parameters of the generative models.

Table 3: Number of examples for each dataset.

	Train Set	Dev Set	Test Set
Laptops	2163	150	638
Restaurants	3452	150	1120

## 4.2 Experimental Settings

The hyperparameters of the generative models were searched through grid search. To tune the hyperparameters we used the dev sets of SemEval. Let’s introduce the triple  $\langle e;lr;o \rangle$  in order to denote a search configuration, where  $e \in \{100, 200, 500, 1000, 2000\}$  is the number of epochs,  $lr \in \{0.01, 0.001, 0.0001\}$  is the learning rate and  $o \in \{sgd, adam, adamax\}$  is the optimizer.

The best configuration for laptops domain is  $\langle 100;0.01;adamax \rangle$ . With these settings the generative model applied to SemEval produced a dataset of 1702 probabilistic examples on laptops. Best results for restaurants domain are obtained with  $\langle 100;0.001;adamax \rangle$ . The cardinality of the probabilistic dataset produced by Snorkel on restaurants is 2471.

Table 4 shows the number of examples for each label. Because the labels are probabilistic, we report the class with highest probability for every training example.

Restaurants training set produced by Snorkel results unbalanced. In order to balance it we limit the number of *Positive* and *Neutral* examples to 700. Results discussed in the next section are computed by averaging the metrics of 10 models trained with *Positive* and *Neutral* examples obtained by different sampling strategies.

Table 4: Number of most probable examples for each label.

	Positive	Negative	Neutral
Laptops	627	448	627
Restaurants	1371	301	792
Restaurants (sampled)	700	301	700

As in (Xu et al., 2019) we trained a simple softmax classifier whose output belongs to  $\mathbb{R}^3$  (3 is the number of polarities) on top of BERT post-trained models. We fine-tuned the discriminative models for 4 epochs using a batch size of 32 and the Adam optimizer with a learning rate of  $3e-5$ . Results are obtained by averaging 10 runs sampling the mini-batches differently.

## 4.3 Results and Discussion

The following tables show the results of accuracy and macro F1 on laptops and restaurants domains. On the left there are the names of fine-tuned models, where

XU-150, XU-300, XU-450 are the models obtained by fine-tuning (Xu et al., 2019) with 150, 300, and 450 hand-labeled examples respectively belonging to the SemEval training set. The examples were randomly extracted to obtain balanced datasets. In particular, we sampled the train set 10 times averaging the results. XU-Weak is the model trained with probabilistic labels computed by our weak-supervision method.

Table 5: Results of the experiments on laptops.

	Accuracy	Macro F1
XU-150	57.77	52.60
XU-300	71.59	68.00
XU-Weak	69.36	65.37

Table 6: Results of the experiments on restaurants.

	Accuracy	Macro F1
XU-150	48.62	39.15
XU-300	69.49	60.79
XU-450	77.93	67.57
XU-Weak	75.39	67.33

Results in Table 5 and Table 6 suggest that the usage in discriminative models of automatically labeled probabilistic examples is promising. Furthermore, our initial experiments suggest that we can easily scale the number of labeled examples needed to get better performances. In particular, our method constitutes a way to deal with real world use cases where the dimension of hand-labeled training sets may become a bottleneck for the implementation of ASC models.

Experimental results seem, also, to confirm that the approach can be easily used cross-domain. However, further experiments on bigger datasets are required to completely assess these initial intuitions.

Best results are obtained on restaurants domain. This could be due to the fact that the restaurants model was post-trained for more epochs and with more examples in (Xu et al., 2019).

Our goal has been to offer an easy-to-use alternative to manual labeling in cross-domain usage of discriminative models for the ASC task. Our method is ready-to-use, it just needs a few labeled examples (we only used 150 examples belonging to SemEval dev sets) in order to tune the Snorkel generative model. It is very useful when applied to many different domains with lots of unlabeled data.

## 5 CONCLUSIONS AND FUTURE WORK

In this work we proposed a weak-supervised approach to the ASC task. We developed a method that exploits the data programming paradigm in order to address the problem. We tested the method on two disjoint domains, laptops and restaurants. We tuned a Snorkel generative model for each domain and used them to label the data in a probabilistic manner. Resulting probabilistic training sets were used to fine-tune discriminative models proposed in (Xu et al., 2019) that were previously post-trained with unlabeled domain data.

The experiments we carried out offer many hints for future work. In particular, obtained results suggest that the approach can be used cross-domain. We plan to perform more experiments on larger datasets in order to confirm initial intuitions. Our future work will be focused on real world datasets to perform extensive experiments on scalability and performances of the method we have proposed in this paper. Moreover, we will improve the method by defining further LFs templates while maintaining its simplicity.

## REFERENCES

- Bach, S. H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., Sen, S., Ratner, A., Hancock, B., Alborzi, H., Kuchhal, R., Ré, C., and Malkin, R. (2019). Snorkel drybell: A case study in deploying weak supervision at industrial scale. In Boncz, P. A., Manegold, S., Ailamaki, A., Deshpande, A., and Kraska, T., editors, *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019.*, pages 362–375. ACM.
- Bird, S. (2006). NLTK: the natural language toolkit. In Calzolari, N., Cardie, C., and Isabelle, P., editors, *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006.* The Association for Computer Linguistics.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Ding, Y., Yu, J., and Jiang, J. (2017). Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3436–3442.
- Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In Kim, W., Kohavi, R., Gehrke, J., and DuMouchel, W., editors, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.
- Pablos, A. G., Cuadros, M., and Rigau, G. (2014). V3: unsupervised generation of domain aspect terms for aspect based sentiment analysis. In Nakov, P. and Zesch, T., editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 833–837. The Association for Computer Linguistics.
- Pablos, A. G., Cuadros, M., and Rigau, G. (2015). V3: unsupervised aspect based sentiment analysis for semeval2015 task 12. In Cer, D. M., Jurgens, D., Nakov, P., and Zesch, T., editors, *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 714–718. The Association for Computer Linguistics.
- Pablos, A. G., Cuadros, M., and Rigau, G. (2018). W2VLDA: almost unsupervised system for aspect based sentiment analysis. *Expert Syst. Appl.*, 91:127–137.
- Pereg, O., Korat, D., Wasserblat, M., Mamou, J., and Dagan, I. (2019). Absapp: A portable weakly-supervised aspect-based sentiment extraction system. *CoRR*, abs/1909.05608.
- Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014).

- Semeval-2014 task 4: Aspect based sentiment analysis. In Nakov, P. and Zesch, T., editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 27–35. The Association for Computer Linguistics.
- Purpura, A., Masiero, C., and Susto, G. A. (2018). WS4ABSA: an nmf-based weakly-supervised approach for aspect-based sentiment analysis with application to online reviews. In Soldatova, L. N., Vanschoren, J., Papadopoulos, G. A., and Ceci, M., editors, *Discovery Science - 21st International Conference, DS 2018, Limassol, Cyprus, October 29-31, 2018, Proceedings*, volume 11198 of *Lecture Notes in Computer Science*, pages 386–401. Springer.
- Qiu, G., Liu, B., Bu, J., and Chen, C. (2009). Expanding domain sentiment lexicon through double propagation. In Boutilier, C., editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1199–1204.
- Qiu, G., Liu, B., Bu, J., and Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.
- Ratner, A., Bach, S. H., Ehrenberg, H. R., Fries, J. A., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. *PVLDB*, 11(3):269–282.
- Ratner, A. J., Sa, C. D., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3567–3575.
- Rietzler, A., Stabinger, S., Opitz, P., and Engl, S. (2019). Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. *CoRR*, abs/1908.11860.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.
- Wang, W. and Pan, S. J. (2018). Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2171–2181.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.
- Xu, H., Liu, B., Shu, L., and Yu, P. S. (2019). BERT post-training for review reading comprehension and aspect-based sentiment analysis. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2324–2335. Association for Computational Linguistics.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.
- Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.