LT-ABSA: An Extensible Open-Source System for Document-Level and Aspect-Based Sentiment Analysis

Eugen Ruppert[‡] and **Abhishek Kumar**[†] and **Chris Biemann**[‡]

[‡]**Language Technology Group**
Computer Science Dept.
Universität Hamburg
http://lt.informatik.uni-hamburg.de

[†]**Indian Institute of Technology Patna**
AI-NLP-ML group
Patna, India
http://www.iitp.ac.in

Abstract

This paper presents a system for document-level and aspect-based sentiment analysis, developed during the inception of the GermEval 2017 Shared Task on aspect-based sentiment analysis (ABSA) (Wojatzki et al., 2017). It is a fully-featured open-source solution that offers competitive performance on previous tasks as well as a strong performance on the GermEval 2017 Shared Task. We describe the architecture of the system in detail and show competitive evaluation results on ABSA datasets in four languages. The software is distributed under a lenient license, allowing royalty-free use in academia and industry.

1 Introduction

Sentiment analysis has gained a lot of attention in recent years in the CL/NLP community. Aggregating over the sentiment in a large amount of textual material helps governments and companies to deal with the large increase of user-generated content due to the popularity of social media. Companies can react to upcoming problems and prepare strategies to help users to navigate reviews and to improve their reputation.

While determining document-level sentiment can be framed as a classification task with two or three classes (positive, negative, possibly neutral), identifying and evaluating aspect-based sentiment is more challenging: here, we are not only interested in the polarity of the sentiment, but also to what particular aspect the sentiment refers to – for example people might express in the same product review that they like the high-resolution screen of a phone while complaining about its poor battery life. Aspects are typically classified into a flat taxonomy, and are lexicalized in opinion target expressions (OTEs), which shall be identified by ABSA systems.

Even though a steady number of sentiment analysis tasks have been conducted in the past years on aspect-based as well as other flavors of sentiment analysis, e.g. (Pontiki et al., 2015; Pontiki et al., 2016; Wojatzki et al., 2017), participants mostly do not share their systems, so that others could use or extend them. Even if systems are shared, they are usually not easy to operate, since they typically stay on the level of research software prototypes. A notable exception is Stanford's CoreNLP project, which however only performs document-level sentiment on English (Socher et al., 2013).

In this paper, we present a fully-featured open-source[1] system for ABSA. Configurations regarding the use of features or the choice of training data can be shared, enabling reproducible results. Our system is flexible enough to support document-level and aspect-based sentiment analysis on multiple languages. Since we also provide feature induction on background corpora as part of the system, it can be applied out of the box.

We focus on engineering aspects. For related work regarding aspect-based sentiment analysis, we refer to the task description papers cited above, as well as recent surveys, e.g. (Medhat et al., 2014).

2 Architecture

The system is designed as an extensible framework that can be adapted to many different datasets. It is able to perform document-level classification as well as the identification of opinion target expressions (OTEs). NLP pre-processing is engineered in the UIMA framework (Ferrucci and Lally, 2004), which contributes to adaptability and modularity. It is a full-fledged system that contains all stages of preprocessing, from reading in different data formats over tokenization to various target outputs, and is aimed at productive use.

[1] The system is available under the permissive Apache Software License 2.0, http://apache.org/licenses/LICENSE-2.0.html

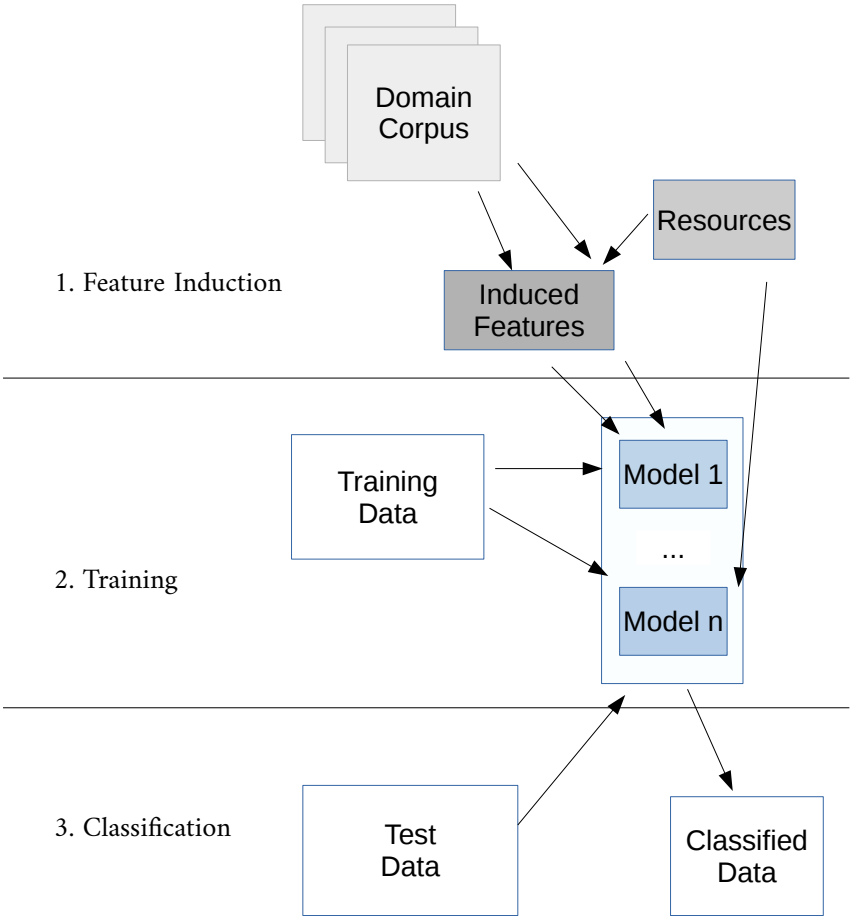

Figure 1: The system workflow of the LT-ABSA system

2.1 Execution and Workflow

The general workflow consists of three major steps (see Figure 1). To prepare model creation, we perform feature induction (1). This step has to be conducted only once when creating a model for a new language or domain. The operator can provide an in-domain corpus to induce features derived from whole-corpus statistics, like tf-idf scores. Furthermore, we support the corpus-informed extension of word lists, such as augmenting a list of positive words with similar words from a background corpus, as described in more detail below. While our system also uses word embeddings as features, their training is not part of our system but needs to be done externally.

In the training step (2), models are trained using labeled training data. The processing pipeline includes readers for several formats to create a document representation, language-specific NLP tools and feature extractors to to create feature vectors. We train machine learning models on these feature representation in order to support two general setups: document-level classification into an arbitrary number of classes, and sequence tagging for extracting spans, such as OTEs.

Finally, the models are used for the classification of new documents (3). This step supports the same file formats and conducts the same feature extraction as in the training step. Additionally, we have included a small web server with an RESTful API with HTML and JSON output (see Listing 1 for an example).

The NLP pipeline includes the rule-based segmenter described in Remus et al. (2016), which allows adapting the tokenization to the target domain, e.g. handle hashtags, cashtags and other types of tokens for social media content. For POS tagging, we rely on OpenNLP[2] for the reason of license compatibility.

3 Features

In this section, we describe our feature induction on background corpora and list the features for

[2] `http://opennlp.apache.org/`

```
{
    "aspect": {
        "label": "DB_App_und_Website#Haupt",
        "score": 0.21274166800759153
    },
    "aspect_coarse": {
        "label": "DB_App_und_Website",
        "score": 0.228312850597364
    },
    "input": "Die App funktioniert nicht, nichts geht mehr",
    "relevance": {
        "label": "true",
        "score": 0.8396798158862353
    },
    "sentiment": {
        "label": "negative",
        "score": 0.46157282933962135
    },
    "targets": ["App"]
}
```


Listing 1: Example response from the web API

document-level classification with support vector machines (SVMs) and sequence tagging with a conditional random field (CRF).

3.1 Feature Induction

Background Corpus We use an in-domain corpus to induce features and semantic models. E.g., for the background corpus on the GermEval 2017 dataset, we used a web crawl obtained by the language-model-based crawler of (Remus and Biemann, 2016). If in-domain data is not available, we still recommend to perform feature induction with a background corpus from the same language, On the background corpus, we compute a distributional thesaurus (DT) (Biemann and Riedl, 2013) and a word2vec model (Mikolov et al., 2013) using the according software packages, which are not part of the distribution. However, we provide the models as well as usage instructions on how to compute them. Further, we compute inverse document frequencies (IDF) of words (Spärck Jones, 1973).

Training Data Using the training data and the idf scores, we determine the tokens with the highest tf-idf scores for each document-level class. The top 30 tokens for each class are used as binary features.

Polarity Lexicon Expansion Assuming the existence of a polarity lexicon (e.g. Waltinger (2010) for German), we automatically expand such lexicon for a language using the method described in our previous work (Kumar et al., 2016): First, we collect the top 10 distributionally most similar words for each entry in each polarity class (positive, negative, sometimes also neutral). Then, we filter these expansions by a minimum corpus frequency threshold of 50 in the background corpus. Next, we only keep the expansions that were present in at least 10 of the seed terms. While distributional similarity does not preserve polarity, described aggregation strategy results in a high-precision high-coverage domain-specific polarity lexicon.

For all expansion terms, we calculate the normalized scores for each polarity, resulting in a real-valued weight for each polarity.

3.2 Document-Based Classifier

We use a linear SVM classifier (Fan et al., 2008) for document-based classification. As the feature space is fairly large and sparse (100+K features for GermEval 2017), we can resort to a linear kernel and do not require more CPU-intensive kernel methods.

- **TF-IDF:** We calculate the tf-idf weights for each token using the IDF from the background corpus and the frequency of the token in the current document, using token weights as features. The overall TF-IDF feature vector is normalized with the L2 norm.
- **Word Embeddings:** We use word embeddings of 300 dimensions trained with word2vec (Mikolov et al., 2013) on background corpora. For the document representation, word representation for each word is obtained and then averaged up to get a 300

dimensional feature vector. Word embedding averaging is done unweighted as well as weighted by the token's tf-idf score. Finally, the averaged feature vector is normalized using the L2 norm.

- **Lexicon:** This feature class allows to supply word lists, recording their presence or absence in a sparse feature vector. We use this feature class for supplying polarity lexicons to our classifier.
- **Aggregated Lexicon:** This feature class also relies on word lists with labels, but aggregates over words from the same class: we supply the relative amount of positive, negative and neutral words in the document, normalized by document length.
- **Expanded Polarity Lexicon:** We use the induced expanded polarity lexicon to generate a low-dimensional feature vector (2-3 features). The expanded polarity lexicon provides a polarity distribution for each term, e.g., schnell (*fast*) – 0.32 (neg-value) – 0.68 (pos-value). We use this feature by summing up the distributions of the tokens that appear in the expanded lexicon and averaging them.

3.3 CRF

The CRF classifier (Okazaki, 2007) is used for annotation of Opinion Target Expressions, cast in a sequence tagging setup. It uses the following symbolic features in the ClearTk[3] framework:

- current token (surface form + lowercased)
- POS tag
- lemma (not available for all languages)
- character prefixes (2–5 characters)
- suffixes (2–5 characters)
- capitalization
- numeric type (identifies types, when numbers are present; e.g. digits, alphanumeric, year)
- character categories (patterns based on Unicode categories)
- hyphenation

These features are computed in a window of +/– 2 tokens around the target token.

[3] http://cleartk.github.io/cleartk/

4 Results

In the experimental results reported below, we have used the following background corpora for feature induction: For German, we have compiled a corpus from a focused webcrawl (Remus et al., 2016). For the SemEval tasks, we employ COW (Schäfer, 2016) web corpora[4] for English, Spanish and Dutch.

4.1 GermEval 2017 Shared Task

The GermEval 2017 Shared Task on ABSA (Wojatzki et al., 2017) features a large German dataset consisting of user-generated content from the railway transportation domain. There are four subtasks that cover document-based and aspect-based sentiment analysis. Participants should classify the binary relevance and the document-level sentiment in Subtasks A and B. Next, they should identify aspects in the document and their corresponding sentiment (Subtask C). Finally, OTEs are identified by span and labeled with an aspect and a sentiment polarity in Subtask D. The task features two test sets: documents from the same period as the training data (synchronic) and documents from a later point in time (diachronic). For evaluation, micro-averaged F1 scores are used.

Our system has been developed in the same project that funded the creation of the dataset used in GermEval 2017. Naturally, as the organizer's entry, it did not compete in the shared task. Nevertheless, we report the ranks our system would have obtained in this task.

Table 1 presents the results on the synchronic dataset and Table 2 on the diachronic dataset. Our system outperforms all baselines and would have ranked highly in the competition, outperforming most submissions on almost every task. On Subtasks A and B, our system is outperformed by a small margin, on Subtasks C and D, we show the best performance overall. We conclude that LT-ABSA is a highly competitive system for sentiment classification on German.

4.2 SemEval-2016 Task 5: Aspect Based Sentiment Analysis

The SemEval-2016 task on aspect-based sentiment analysis (Task 5; (Pontiki et al., 2016)) is comparable in structure to Subtasks B, C and D in the GermEval-2017 evaluation. While the overall task was conducted on datasets in eight languages and

[4] http://corporafromtheweb.org/

Table 1: GermEval 2017 results, synchronic testset (F1 score)

System	Relevance	Sentiment	Aspect	Aspect + Sentiment	OTE (exact)	OTE (overlap)
MCB	0.816	0.656	0.442	0.315	–	–
Baseline system	0.852	0.667	0.481	0.322	0.170	0.237
Best contender	0.903	0.749	0.482	0.354	0.220	0.348
Our system	0.895	0.767	0.537	0.396	0.229	0.306
Rank	3	1	1	1	1	2

Table 2: GermEval 2017 results, diachronic testset (F1 score)

System	Relevance	Sentiment	Aspect	Aspect + Sentiment	OTE (exact)	OTE (overlap)
MCB	0.839	0.672	0.465	0.384	–	–
Baseline system	0.868	0.694	0.495	0.389	0.216	0.271
Best contender	0.906	0.750	0.460	0.401	0.281	0.282
Our system	0.894	0.744	0.556	0.424	0.301	0.365
Rank	3	2	1	1	1	1

Table 3: Results on SemEval-2016, Task 5

Dataset	System	SB1, Slot 1 (F)	SB1, Slot 3 (Acc)	SB2, 2 (Acc)
English	Baseline	0.599	0.765	0.743
Restaurants	Top system	0.730	0.881	0.819
	LT-ABSA	0.651	0.782	0.731
	Rank	16	19	5
English	Baseline	0.375	0.700	0.730
Laptops	Top system	0.519	0.828	0.750
	LT-ABSA	0.412	0.736	0.675
	Rank	17	12	5
Dutch	Baseline	0.428	0.693	0.732
Restaurants	Top system	0.602	0.778	–
	LT-ABSA	0.578	0.824	0.863
	Rank	2	1	–
Spanish	Baseline	0.547	0.778	0.745
Restaurants	Top system	0.706	0.836	0.772
	LT-ABSA	0.586	0.821	0.797
	Rank	9	2	1

multiple domains, we have only experimented with the English, Spanish and Dutch datasets. Table 3 presents the results, again with ranks that our system would have obtained in the task. We report scores on Subtask 1, Slots 1 (Sentence-level Aspect Identification) and 3 (Sentiment Polarity), and on Subtask 2, Slot 2 (Document-level Sentiment Polarity).[5]

Overall LT-ABSA is able to beat all baselines for the reported slots. Only for SB2, Slot 2 on English, where the baselines rank in the middle, we are outperformed by the baselines. The performance varies across tasks. For the highly contested English datasets, we rank in the lower midfield for SB1 and in the top 5 for SB2. For the less contested Spanish and Dutch datasets, we show a competitive performance.

[5]We used our system out-of-the-box, without adaptation to the tasks. E.g., in Subtask 2, the entities are already given and need to be classified. We also identify the aspects.

5 Conclusion

We present a flexible, extensible open source system for document-level and aspect-based sentiment analysis and have reported state of the art results on two shared tasks in four different languages. Code and documentation are available on GitHub.[6]. We also provide complete feature sets and trained models for all experiments reported in this paper.[7]

Acknowledgements

This work has been supported by the Innovation Alliance between the Deutsche Bahn and TU Darmstadt, Germany, as well as a DAAD WISE internship grant.

References

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

David Ferrucci and Adam Lally. 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering 2004*, 10(3-4):327–348.

Ayush Kumar, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. 2016. IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1129–1135.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Electrical engineering. *Ain Shams Engineering Journal*, 5(4):1093–1113.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop at International Conference on Learning Representations (ICLR)*, pages 1310–1318, Scottsdale, AZ, USA.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

[6]Code: `https://github.com/uhh-lt/LT-ABSA`

[7]Data and Models: `http://ltdata1.informatik.uni-hamburg.de/sentiment/`

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, CO, USA.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud Maria Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5 : aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, CA, USA.

Steffen Remus and Chris Biemann. 2016. Domain-Specific Corpus Expansion with Focused Webcrawling. In *Proceedings of the 10th Web as Corpus Workshop*, pages 106–114, Berlin, Germany.

Steffen Remus, Gerold Hintz, Darina Benikova, Thomas Arnold, Judith Eckle-Kohler, Christian M. Meyer, Margot Mieskes, and Chris Biemann. 2016. EmpiriST: AIPHES. Robust Tokenization and POS-Tagging for Different Genres. In *Proceedings Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 106–114, Portorož, Slovenia.

Roland Schäfer. 2016. On bias-free crawling and representative web corpora. In *Proceedings of the 10th Web as Corpus Workshop*, pages 99–105, Berlin, Germany.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, WA, USA.

Karen Spärck Jones. 1973. Index term weighting. *Information Storage and Retrieval*, 9(11):619 – 633.

Ulli Waltinger. 2010. Sentiment Analysis Reloaded: A Comparative Study On Sentiment Polarity Identification Combining Machine Learning And Subjectivity Features. In *Proceedings of the 6th International Conference on Web Information Systems and Technologies (WEBIST '10)*, Valencia, Spain.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In *Proceedings of the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, Berlin, Germany.