An Annotation Framework for Luxembourgish Sentiment Analysis

Joshgun Sirajzade[1], Daniela Gierschek[2], Christoph Schommer[1]
[1]MINE Lab, Computer Science, Université du Luxembourg
[2]Institute of Luxemburgish Linguistics and Literatures, Université du Luxembourg
{joshgun.sirajzade, daniela.gierschek, christoph.schommer}@uni.lu

Abstract

The aim of this paper is to present a framework developed for crowdsourcing sentiment annotation for the low-resource language Luxembourgish. Our tool is easily accessible through a web interface and facilitates sentence-level annotation of several annotators in parallel. In the heart of our framework is an XML database, which serves as central part linking several components. The corpus in the database consists of news articles and user comments. One of the components is LuNa, a tool for linguistic preprocessing of the data set. It tokenizes the text, splits it into sentences and assigns POS-tags to the tokens. After that, the preprocessed text is stored in XML format into the database. The Sentiment Annotation Tool, which is a browser-based tool, then enables the annotation of split sentences from the database. The Sentiment Engine, a separate module, is trained with this material in order to annotate the whole data set and analyze the sentiment of the comments over time and in relationship to the news articles. The gained knowledge can again be used to improve the sentiment classification on the one hand and on the other hand to understand the sentiment phenomenon from the linguistic point of view.

Keywords: Opinion Mining / Sentiment Analysis, Corpus (Creation, Annotation, etc.), Crowdsourcing

1. Introduction

Dealing with a low-resource language like Luxembourgish comes with special challenges. Especially in the case of sentiment analysis, one is faced with the scarcity of resources. Neither sufficient amount of pre-labeled data, like Twitter data sets for English or several other languages, nor lexical resources such as SentiWordNet (Baccianella et al., 2010) exist for Luxembourgish. Nevertheless, doing research on sentiment detection for this language is important. First, it is a technical and scientific challenge for the algorithms. They need to be adapted to a scarce data set. The state-of-the-art NLP algorithms, especially those used in deep learning, require a lot of data. Many attempts have been made to adjust them to low-resource languages. Leveraging more information from the data, like the usage of subword units, is one of those attempts. Second, it is important to study these languages, because they play a crucial role in the political, economic and social lives of their speakers. For these reasons, we have decided to build an infrastructure for the Luxembourgish language which is freely accessible over the web using data from RTL Luxembourg (Radio Télévision Lëtzebuerg).

2. Luxembourgish Language

Luxembourgish, French and German are the three official languages of the Grand Duchy of Luxembourg, a European country with about 590,000 inhabitants (Gilles, in press). In 1984, the Luxembourgish language, which arised out of a Central Franconian dialect, was allocated the status of the only national language of the country and is an essential symbol for national identity today. Despite being related to German due to its history, Luxembourgish is perceived as an independent language by the speech community. If all people taking part in the conversation speak Luxembourgish, code-switching to another language would be unthinkable regardless of the formality or informality of the situation ((Gilles, in press); (Gilles, 2015)). One big challenge of implementing a natural language processing task

with texts written in Luxembourgish is that a big part of it is not written following the official spelling rules of the language. This characteristic results from the educational system mainly focusing on French and German and not that much on Luxembourgish (Gilles, 2015).

3. Sentiment Analysis

The research field of sentiment analysis, or opinion mining, deals with analyzing people's opinions towards certain entities (Liu, 2012). There are two different main approaches to solving sentiment analysis tasks, i.e. lexical and machine learning methods. Lexical approaches leverage automatically or manually developed dictionaries containing the sentiment of specific words to calculate the overall sentiment score of an entity. Those dictionaries can consist of different words and their positivity or negativity while sometimes adding even the degree of this (Taboada, 2016). On the machine learning side, sentiment analysis was for a long time treated as a simple text classification task (Pang et al., 2002). Recent research however shows that sentiment analysis can be performed on different levels, such as on a sentence or aspect based level (Liu, 2015). Besides, efforts to adapt text representations needed for text classification to sentiment analysis have been made. For instance, sentiment specific word embeddings were trained that incorporate the sentiment information in the continuous representation of words (Tang et al., 2014).

Also, attempts to use transfer learning for implementing sentiment analysis for situations in which data is scarce have been made (Bataa and Wu, 2019). Big data sets for pretraining a Luxembourgish language model before fine-tuning on sentiment data do not exist yet. Since we believe that the expression of sentiment is culture and language-dependent, pretraining on a large dataset from a related language such as German is not an option. The morphosyntactical system of Luxembourgish and Standard German are very different and for some features, Luxembourgish has developed new grammatical structures that do not ex-
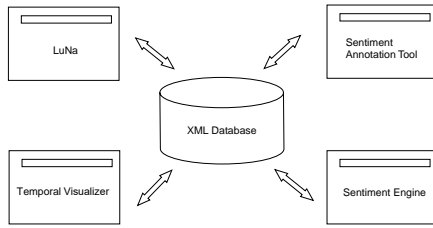
Figure 1: The architecture of our temporal text data warehouse

Figure 2: The GUI of the annotation tool

ist in Standard German (Gilles, in press) which make the two languages quite distinct from each other. Leveraging a lexical method for sentiment analysis is currently also not feasible due to a significant amount of spelling variation in Luxembourgish (see section 2.). An intensive amount of preprocessing is needed to remove all of this diversity and no such tool that captures all of the variation present in the Luxembourgish language has been built yet. In order to best fulfil the demand for a culture and language specific sentiment analysis setting, we have built our own tool for crowdsourcing Luxembourgish specific sentence level sentiment annotations.

4. Technical Implementation

4.1. A Database – Temporal Warehouse

We have set up a Temporal Warehouse for our project that acts as a data backbone to separate the data itself from the various applications that are linked to it. Users can access the Temporal Warehouse either by retrieving data using XQuery commands or by loading such data into the warehouse. For the database itself, we leverage the eXist-db infrastructure (Siegel and Retter, 2014). Each data entry in our database is of textual form and possesses both a time and a sentiment stamp. All texts in our warehouse are in XML format (Gierschek et al., 2019). If needed, the Temporal Warehouse can easily be reproduced from its data sources following an Extract-Transform-Load (ETL) pipeline (Kimball and Ross, 2013) which is, at present, facilitated by various Python and Java scripts. In order to provide textual data that can be loaded into our Temporal Warehouse, we can use two different tools. One is the LuNa Open Toolbox for the Luxembourgish Language which we used for tokenization, sentence splitting and part-of-speech tagging of the data (Sirajzade and Schommer, 2019). The other one is the annotation tool that we propose in this paper. We have decided to incorporate this as part of our database structure, as it allows for new annotations to be gathered with little additional effort. The data loaded into the Temporal Warehouse through the annotation tool or the LuNa Open Toolbox for the Luxembourgish Language is crucial as training data for further analyses that can be done using the Temporal Visualizer and the Sentiment Engine to detect sentiment changes over time.

4.2. Sentiment Annotation Tool

As mentioned before, the power of our Temporal Warehouse as a linguistic resource lies in its well-structured

annotation. Being accessible as database it is possible to gather new annotation with little additional effort. The Sentiment Annotation Tool is a separate component of this warehouse. More precisely, it is a web application written in the programming language PHP. It connects to our eXist database, retrieves sentences from the user comments of our corpus and offers the annotator the possibility to annotate them with the labels *negativ* [negative], *neutral* [neutral] or *positiv* [positive]. If unsure how to label the instance, the button *iwwersprangen* [skip] might be clicked or the user can go back to the *Uleedung* [instruction] page. The sentences are presented in random order to the annotator to cover the whole time period of our text collection, i.e. from 2008 to 2018. To give a bit of context, one sentence before and one sentence after the sentence to be annotated are also shown. A counter returns the number of items already provided with a sentiment value as a motivational incentive. Figure 2 shows the graphical user interface [GUI] of our Sentiment Annotation Tool.

Label	Amount
negativ [negative]	1854 sentences
neutral [neutral]	1417 sentences
positiv [positive]	942 sentences
iwwersprangen [skip]	11 sentences

Table 1: Annotated sentences

5. Experiments

5.1. Data Set

The large data set leveraged for collecting sentiment annotations was provided by our project partner RTL Luxembourg (RTL Luxembourg, 2019a). It comprises over 180,000 news articles written between 1999 and 2018 and over 500,000 user comments to those articles. Most of those texts were published in Luxembourgish with only little occurence of publications in the other two official languages, i.e. German and French. The comments were recorded between 2008 and 2018 and include over 35 million running tokens whereas the news articles part amounts to about 30 million running tokens.

We carried out several preprocessing steps on this data to prepare our corpus for retrieval in the annotation tool. More precisely, we leveraged the LuNa Open Toolbox for the

```
<sentence id="7ep">
<w id="84" pos="P" sen="7" tagger="0,26">Mir</w>
<w id="85" pos="V" sen="7" tagger="0,27">ziichten</w>
<w id="86" pos="D" sen="7" tagger="0,19">eng</w>
<w id="87" pos="N" sen="7" tagger="0,24">imposeiert</w>
<w id="88" pos="N" sen="7" tagger="0,52">Meenung</w>
<c id="89" pos="$" sen="7" tagger="0,31">.</c>
</sentence>
```

Figure 3: Example of a sentence in our data set

Annotation measure	Result
Fleiss kappa	-0.018
Krippendorff's alpha	0.19

Table 2: Inter-annotator agreement

Luxembourgish Language to tokenize, split the sentencces and to include part-of-speech codes in our data (Sirajzade and Schommer, 2019). Figure 3 illustrates a preprocessed sentence from our data.

5.2. Annotation Process

The annotators were recruited through crowdsourcing techniques. A call for participation was posted on the RTL Luxembourg website (RTL Luxembourg, 2019b) and in Luxembourgish schools. In total, we recruited 26 annotators. They annotated 4,206 sentences, as shown in table 1. The annotation guidelines were very open to avoid influencing the annotator during the decision making process. We simply asked the users to annotate from the perspective of the author (Abdul-Mageed and Diab, 2011). 637 sentences of our corpus were annotated three times, 1302 two times and 2274 one time. The amount of data annotated by an annotator varied greatly, from one annotation to almost 600. We will use those sentences annotated three times to calculate the inter-annotator agreement.

5.3. Inter-Annotator Agreement

We measured the inter-annotator agreement for the 637 sentences that were annotated by three of the 26 annotators we recruited. For those calculations, we chose Fleiss' kappa and Krippendorff's alpha as both allow the calculation of reliability for multiple annotators. Fleiss' kappa assumes that the coders' distributions are independent from one another (Artstein and Poesio, 2008). Krippendorff's alpha is a reliability measure able to determine the agreement for any number of observers, categories and even if some values are missing (Krippendorff, 2011). The calculation of expected agreement is done by looking at the overall distribution of the different judgments, no matter which annotator provided them (Artstein and Poesio, 2008). An α value of 0 means the absence of reliability, whereas a value of 1 would be a perfect score (Krippendorff, 2011). For kappa, a value of 1 denotes perfect agreement and 0 pure chance agreement (Artstein and Poesio, 2008). Table 2 shows the calculated Fleiss' kappa and Krippendorff's alpha for our annotation setup. Both scores are close to zero. Our annotated data set thus shows low agreement. One reason for this might be that giving two sentences as context is too short to infer sentiment. Also, the annotation guidelines might be too open and therefore might make the annotators unsure in case of doubtful cases where sentiment is not clearly expressed. Note, however, that our Sentiment Annotation Tool is still open to new annotations and annotators. We therefore will recalculate Fleiss' kappa and Krippendorff's alpha in the future and compare those results with the ones presented here.

5.4. Sentiment Engine

We created our Sentiment Engine with deeplearning4j, an easy to use Java library for building ANNs. We use the word2vec algorithm (Mikolov et al., 2013) for constructing vectors that are used for the input layer of our network. More precisely, we first train our embedding model with over 100 million tokens of text written in Luxembourgish that were gathered at the Institute of Luxembourgish Linguistics and Literatures and include the comments and news articles from our RTL corpus. The word embeddings have 100 dimensions. Our network has a very simple RNN design. It has four layers: one input, two hidden and one output layer. As an activation function for the hidden layers we used hyperbolic tangent (tanh), which is an S-shaped function transforming the values x into the range $[-1, 1]$. The output layer gets softmax as its activation function, because we wanted to see the probability like values at the end of the classification. The first three layers have 256 neurons and the last one only three, corresponding to our labels. Word vector inputs created from the sentences are mostly padded and if there should be more than 256, it is truncated. The data set has a 80%/20% split and the network gives an accuracy score of 80%. We subsequently use the trained model to tag all sentences in our comments for the whole time span of 2008 until 2018 in order to see and to investigate the temporal patterns in our data.

5.5. Visualization

The setup of our Search Engine is similar to the architecture of the annotation tool. We built the engine using PHP and the XML database eXist-db to visualize the results of the performance of the Sentiment Engine. Figure 4 shows the functionality of it. The Search Engine makes it possible to search for words in our database and returns those words in their context. The sentences found are shown in the colors corresponding to their sentiment. Green is used for displaying positive, red for negative and grey for neutral sentences. We also developed a module which can present the change in sentiment over time. Figure 5 shows the absolute frequencies of positive/negative/neutral sentences per month in the period of 2008-2018 for the search word "Autobunn" [interstate].

6. Future Work

We have presented an annotation setup for collecting sentiment annotated data for the low-resourced language Luxembourgish. Our total of annotated data results to 4,206 sentences from 26 different annotators. In order to study the quality of the annotations gathered through crowdsourcing, we calculated Fleiss' kappa and Krippendorff's alpha. The results of our calculation of Fleiss' kappa and Krippendorff's alpha show that annotation for sentiment, especially
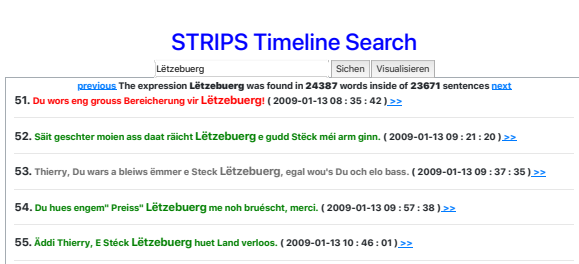
Figure 4: The GUI of our Search Engine

Figure 5: The change in sentiment over time in the example of the word Autobunn [interstate]

sentence-level annotation with open annotation guidelines, is a challenging task. Future work will include a further examination of the (open) annotation guidelines provided and the setup of crowdsourcing annotations. A possibility would be to carry out annotations with trained annotators, proving them with more detailed guidelines and then to compare the agreement to the one achieved with our crowdsourced annotators. We also plan to compare the data annotated by humans with the automatic annotations of our Sentiment Engine and to further improve our neural network. Future experiments will include the improvement of the training process by using all sentences where at least two of the three annotators agree for sentiment. This will increase the amount of training data. Furthermore, we would like to leverage annotated sentences with three different sentiments as well. By incorporating them into our training corpus with a smaller weight than the sentences annotated with the same sentiment by three or two users, we could increase the training set even more.

7. Acknowledgements

8. Bibliographical References

Abdul-Mageed, M. and Diab, M. (2011). Linguistically-motivated subjectivity and sentiment annotation and tagging of Modern Standard Arabic. *International Journal on Social Media MMM: Monitoring, Measurement, and Mining*.

Artstein, R. and Poesio, M. (2008). Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Nicoletta Calzolari, et al., editors, *LREC*. European Language Resources Association.
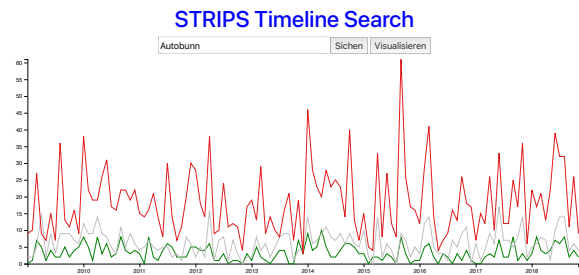
Bataa, E. and Wu, J. C. K. (2019). An Investigation of Transfer Learning-Based Sentiment Analysis in Japanese. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 4652–4657.

Gierschek, D., Gilles, P., Purschke, C., Schommer, C., and Sirajzade, J. (2019). A Temporal Warehouse for Modern Luxembourgish Text Collections.

Gilles, P. (2015). From status to corpus: Codification and implementation of spelling norms in Luxembourgish. *In W., Davies and E., Ziegler (Eds.), Macro and micro language planning (pp. 128-149). London: Palgrave Macmillan (2015).*

Gilles, P. (in press). Luxembourgish. *In P., Maitz, H. C., Boas (Ed.), A., Deumert (Ed.) and M., Louden (Ed.), Varieties of German Worldwide. Oxford: Oxford University Press (in press).*

Kimball, R. and Ross, M. (2013). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition.

Krippendorff, K. (2011). Computing Krippendorff 's Alpha-Reliability.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Liu, B. (2015). *Opinions, Sentiment, and Emotion in Text*. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *CoRR*, abs/1310.4546.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

RTL Luxembourg. (2019a). RTL. https://www.rtl.lu/. Accessed: 2019-09-05.

RTL Luxembourg. (2019b). Wéi si se geduecht: Positiv? Negativ? Neutral? https://www.rtl.lu/kultur/news/a/1445783.html. Accessed: 2020-02-08.

Siegel, E. and Retter, A. (2014). *eXist: A NoSQL Docu-*

ment Database and Application Platform. O'Reilly Media.

Sirajzade, J. and Schommer, C. (2019). The LuNa Open Toolbox for the Luxembourgish Language. *In Petra Perner (Ed.), 19th Industrial Conference, ICDM 2019 New York, USA, July 17 to July 21 2019, Poster Proceedings 2019, Advances in Data Mining, Applications and Theoretical Aspects.*

Taboada, M. (2016). Sentiment Analysis: An Overview from Linguistics. *Annual Review of Linguistics*, 2(1):325–347.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565.