STEREO: Automatic Extraction of Statistic and Experimental Conditions from Scientific Papers

STEFFEN EPP, MARCEL HOFFMANN, NICOLAS LELL, MICHAEL MOHR, and ANSGAR SCHERP*

This paper addresses the problem of extracting statistic along with topic and conditions from scientific publications. For this purpose we present STEREO, a two step pipeline. In the first step, statistics extraction, we propose an active wrapper approach based on regular expressions. We split the rule set into rules strictly following APA style and rules matching deviating writing styles. Step two is divided into two parts. Sentence topic extraction is done using an unsupervised machine learning algorithm, Attention-Based Aspect Extraction. Condition extraction is done using Grammar-Based Condition Extraction, a grammar- and keyword-based active wrapper. We applied and evaluated our tool to the CORD-19 dataset. The statistic extraction obtained nearly 100 % accuracy on APA conform and around 95 % on non APA conform writings styles. In step two we obtained mixed results. We conclude that the proposed approaches are good foundations for automatic analysis of scientific papers which can be used to get first impressions of an unknown paper or to analyze statistical writing styles in different domains.

1 INTRODUCTION

In many fields of science the research results are analyzed and presented with statistical methods, e. g., in life sciences, social sciences, or economics. Therefore, there is a large amount of scientific papers which contain statistical data in an unstructured way. Our objective is to identify and extract such data from these papers. There are already tools like Statcheck [1] that address this problem. This shows that the need for tools extracting significant information autonomously is high. Statcheck is already doing great by extracting statistical data written in American Psychology Association (APA) style [1]. Although our general assumption relies on APA style conform presentation of statistics as well, we also extract records which are different to some degree.

To implement this, we use active wrapper induction to find regular expressions even if the paper did not strictly follow APA style guidelines. With these rules we can also extract statistics like these "*Physical demand* $(t(23) = -2.22, p = 0.37)$ *and temporal demand* $(t(23) = 2.72, p = .012)$ *are significantly different*", although APA style dictates the statistics to be at the end of the sentence. Beside the robust extraction of not completely APA style conform statistics, we can extract the details of the statistical measure, e. g., "*There was no significant effect for sex, (t(38) = 1.7, p = .097) despite women attaining higher scores than men*", we extract $degreeOfFreddom = 38$, $statisticVal = 1.7, pvalue = .097$, $topic = personal\_data$ and $conditions = \{men, women\}$.

Beyond Statcheck, we extract experimental conditions and topic such as "men", "women" and "personal_data" as shown in the example above, to increase the interpretability of our extracted records. For extracting the conditions, we apply aspect extraction techniques. We use machine learning and rule-based approaches, namely Attention-based Aspect Extraction (ABAE) [2] and grammar-based condition extraction (GBCE). The grammar-based approach applies rules based on English grammar and frequently occurring tokens to extract experimental conditions of the corresponding statistic.

Currently, the supported statistics are *Pearson's Correlation, Spearman Correlation, Student's t-test, ANOVA, Mann-Whitney U Test, Wilcoxon Signed-Rank Test, and Chi-Square Test.* However it is easy to add other types of statistic and learn corresponding rules. The tool was applied and evaluated on the CORD-19 dataset. The presented results show high accuracy for the statistic extraction part with a drop in non-APA compared to APA conform statistics. Furthermore some statistic types were more often extracted than others, for example more pearson correlations than chi-square tests were found. In addition it was analyzed how different pairs of parameters were missing. For ABAE, we obtained mixed results. There were some models with high accuracy on both the APA and non-APA conform sentences but not all of the extracted aspects were useful for the idea of our tool. The GBCE obtained mixed results as well, also with a drop in non-APA compared to APA conform statistics.

The paper is organized as follows: In Section 2, we discuss some work related to our approach. In Sections 3,4, and 5 we outline the extraction pipeline and the theory of these approaches. Our experimental setup is described in Section 6. The results are presented in Section 7 and discussed in Section 8. In Section 9 some obstacles of the development process are explained. In the final Section 10 the major insights are summarized.

2 RELATED WORK

2.1 Statistic / Metadata Extraction

A tool quite similar to our work is "Statcheck", developed by Nuijten, Hartgerink, Assen, *et al.* [1]. It is based on regular expressions and the most common test statistics used in psychology, for example t, F and χ^2 statistics can be extracted. Only statistics written in APA style notation were considered. The regular expression for each kind of significance test have been hard-coded like `test statistic(df1, df2) =/</>...,p=/</>...`. In a second step, statistical information have been extracted and the p-values have been recomputed to validate the statistic stated by the author. In contrast to StatCheck, our general assumption is, that we expect an APA style conform presentation of statistics. However, we will not ignore statistics that deviate to some degree from the APA style guidelines.

Similar to statistic extraction, the extraction of metadata of articles is a well studied problem. Especially for general domain meta data such as titles, sections or bibliography, [3] [4]. Two examples for such systems are CERMINE and Grobid-quantities.

CERMINE- a comprehensive tool for automatic metadata extraction from born-digital scientific literature is a tool which extracts a rich pool of metadata, e. g. title, author, abstracts and many more, developed by Tkaczyk, Szostek, Fedoryszak, *et al.* [4]. Furthermore

*Supervisor

[1]https://apastyle.apa.org/

Authors' address: Steffen Epp, steffen.epp@uni-ulm.de; Marcel Hoffmann, marcel.hoffmann@uni-ulm.de.de; Nicolas Lell, nicolas.lell@uni-ulm.de; Michael Mohr, michael.mohr@uni-ulm.de; Ansgar Scherp, ansgar.scherp@uni-ulm.de.

it provides the bibliographic reference along with their metadata and the full text structured in sections and subsections. Similar, Grobid (GeneRation Of BIbliographic Data)[2], is a framework based on machine learning for extracting, parsing and re-structuring raw documents such as PDF into structured XML/TEI encoded documents [3].

CERMINE has two Phases. In Phase one it segments the page in meta structures like tiles, sections and bibliography. To achieve this, all the characters along with their dimensions and coordinates are extracted by the *iText library*. After that, the hierarchical structure in pages, zones, lines, words and characters are extracted by a bottom up docstrum algorithm. Finally the document's zones are classified by a support vector machine (SVM) into four categories: *metadata, body, references and other*. In contrast, Grobid uses multiple conditional random field (CRF) models, each specified on handling respective information sets. The most comprehensive model processes the header information of a scientific paper and extracts different metadata information such as titles, authors, affiliations, address, abstract, keywords, etc. In phase two of CERMINE, the metadata is extracted out of the before mentioned zones by a combination of a SVM and a rule-based approach. To extract the references they used k-means clustering to divide the reference zones into references strings and extract the reference metadata by using a CRF. In the end, the output is a XML document which represents the hierarchical structure of the document and has tags for the extracted metadata. In a complete PDF processing, Grobid manages 55 final labels used to build relatively fine-grained structures, from traditional publication metadata (title, author first/last/middlenames, affiliation types, detailed address, journal, volume, issue, pages, doi, pmid, etc.) to full text structures (section title, paragraph, reference markers, head/foot notes, figure headers, etc.) Similar to CERMINE, we structured our approach into two phases, using a combination of a rule-based and a machine learning approach.

Beyond these two general purpose tools, there also is a more specific metadata extraction tool that relates to our work. An extension of Grobid is Grobid-quantities, which is an open-source application for extracting and normalizing measurements from scientific and patent literature which was introduced by Foppiano, Romary, Ishii, *et al.* [5] and is built on top of Grobid. Since Grobid-quantities is extracting numerical data, it is closer related to our work. Grobid's machine learning architecture follows a cascade approach and the models are trained using CRF. The extraction supports quantities (atomic values, intervals and lists), units (such as length, weight), and different value representations (numeric, alphabetic, or scientific notation). These extracted measurements are then normalized toward the International Systems of Units (SI). The architecture of this model is separated into three steps: Tokenization, measurement extraction, and parsing and quantity normalization. Before the tokenization step, the text or PDF is structured using Grobid. The tokens are then created by splitting by punctuation marks and is then re-tokenized to separate adjacent digits and alphanumeric characters. The tokens are then passed through three different machine learning models in cascade (one CRF model for quantities, units and values respectively), to determine appropriate quantity, value and, unit tags. For this approach, a list containing units with their characteristics are created for English, German, and French. This so called Unit Lexicon is used for labeling. For the normalization, an external Java library called Units of Measurement is used.

Since the structure of statistical records can differ greatly, the tokenization of them is more complex. Therefore we chose a different approach than Grobid-quantities. We decided to use a rule-based approach to find statistics, inspired by "Statcheck" with less restrictions on the format of the statistical report by an adapted wrapper induction approach. This way, we are able to learn pattern that can detect statistics which deviate from the APA style guidelines and tolerate incomplete statistics to some degree.

2.2 Aspect Extraction

We discuss the related work on aspect extraction, as it is related to our task of detecting and extracting sentence topics and experimental conditions from text. The methodology presented by Liu, Gao, Liu, *et al.* [6] is an unsupervised and domain independent syntactical approach for aspect extraction. It employs rules about grammar dependency relations between opinion words and aspects. The approach aims to effectively select a set of rules automatically. Therefore, at first, a small subset of manually selected rules based on a set of dependency relations is used as a basis. Their proposed algorithm takes a set of rules and finds the best subset of rules for the dataset. The rules are divided into three types. The first type of rules are using opinion words to extract aspects, based on dependency relations between them (R1). The second type of rules are using aspects to extract other related aspects (R2) and the third are using aspects and opinion words to extract new opinion words. The rule-set selection algorithm runs in three steps. First, every proposed rule is applied to the training dataset and outputs the precision and recall values of the rule. For each ruleset (R1-R3) a ranking based on the precision of the rules is then calculated. In step three, leveraging on step one and two, the rules from the ranked rule set are added one by one in descending order and are evaluated.

This is repeated for every rule in the ranked list. The algorithm then prunes the lower-ranked rules from the rule set to produce the final set of rules only with the best result on the training dataset. However, the rules need to be carefully selected and tuned manually. This is not possible for our approach, since we do not have a labeled dataset to classify the usefulness of created rules.

So far, we are unaware of an approach that automatically creates rules for syntactical extraction. The related paper by Liu, Gao, Liu, *et al.* [6] uses an approach, that is related to associative classification. It uses association rule mining algorithms from Liu, Hsu, and Ma [7] to generate the complete set of association rules, and then selects a small set of high quality rules for classification. The problem with applying this approach in our case is the need for labeled training data.

Xu, Liu, Shu, *et al.* [8] propose a method of combining two different word embeddings with a convolutional neural network (CNN) for aspect extraction. Different embeddings and combinations of embeddings with CNNs and long short-term memory (LSTM) based neural networks were tested. It was found, that a general purpose embedding trained on a huge dataset (in their case glove.840B.300d)

[2]**Grobid** https://github.com/kermitt2/grobid

combined with a domain specific, smaller embedding that is trained for the specific task coupled with a CNN and a final softmax layer performed best in their tests. As with the approach by [6], we cannot use this one as it requires a lot of labeled data for training the CNN weights.

The paper by Karamanolakis, Hsu, and Gravano [9] presents a weakly supervised approach for training neural networks for aspect extraction with only a small set of seed words instead of labeled training data. Seed words are keywords describing an aspect that needs to be available for training. This method adopts the distillation approach by Hinton, Vinyals, and Dean [10], where a simple neural network (student) gets trained to imitate the predictions of a complex network (teacher). During training, the parameters of the teacher are "distilled" to the parameters of the student. In the best case, the student will perform comparably to the teacher for the given task but with less complexity. The teacher is normally trained on a labeled dataset. In this approach, it is a bag-of-words classifier using only seed words. Therefore, seed words that are predictive of the K aspects get incorporated into (generalized) linear bag-of-words classifiers. The student is an embedding-based neural network. First, a segment is embedded and then classified to the K aspects by using the softmax function. The two choices for the embedding functions are an unweighted average of word2vec[11] embeddings and contextualized embeddings using BERT[12]. The student network gets trained to imitate the teacher's predictions by minimizing the cross entropy between the student's and the teacher's predictions. The problem with this approach is the selection of suitable seed words. Good seed words are needed for every aspect which is possible to achieve in aspect extraction on reviews as there is only a known small number of different aspects. In restaurant reviews, for example, two of the aspects they mention are price with the seed words price, value, money, worth, and paid, and the second aspect drinks with the seed words wine, beer, glass, and cocktail. In our case this is not feasible since there can be an unlimited number of aspects considering there can be arbitrary many different experiments with statistics as result.

He, Lee, Ng, *et al.* [2] proposed an unsupervised approach for aspect extraction, which they call Attention-based Aspect Extraction (ABAE). It combines word embeddings with an attention mechanism to create sentence embeddings and tries to extract an aspect embedding with an autoencoder. It does not need any labels for training and seems like a good fit for our work. One problem is that one has to specify the number of aspects it should try to find in the data. We will apply ABAE to our problem. Thus, a detailed explanation will follow in Section 5.1

In their paper Angelidis and Lapata [13] propose a modification and extension to ABAE which they call Multi-Seed Aspect Extractor (MATE). They use the embedding of seed words for every aspect to create a seed matrix. By multiplication with a trained or chosen weight vector these seed matrices are reduced to a vector each and are concatenated to form the aspect matrix. This matrix is then used as aspect matrix in ABAE. Then they use this aspect extraction model together with a polarity prediction model and a segment selection policy to summarize opinions. In the standard ABAE, the aspect matrix is initialized with the centroids of a clustering on the embedding and then fine-tuned during training.

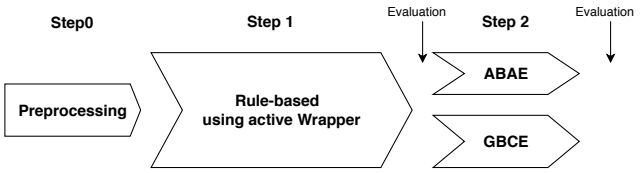

Fig. 1. The Extraction Pipeline to extract statistics, topics, and experimental conditions with the evaluation points.

This seed method seems to produce slightly better results than standard ABAE, but as seed words are needed for every aspect like in the approach of Karamanolakis, Hsu, and Gravano [9] that was introduced above, we cannot use this method.

NLP Libraries. A well known Python Natural Language Processing (NLP) library is Natural Language Toolkit (NLTK). It is a string processing library used for tasks such as tokenization, lemmatization, stemming, parsing, part-of-speech (POS) tagging, etc. This library has tools for almost all NLP tasks. It provides many third party extensions and is most popular in education and research. It provides several pre-trained models and corpora. But it is slow compared to other libraries and only splits text by sentences without analyzing the semantic structure. Another tool providing the same functionality is spaCy [14]. The tool is an advanced NLP library available in Python and Cython. It features tokenization for 50+ languages, convolutional neural network models for tagging, parsing, and named entity recognition. Both provide an easy to use POS tagging and for (generalized) chunking, both expose a rule-based interface. While NLTK is string-based, spaCy is object-oriented and offers the fastest syntactic parser currently available, according to independent researchers from Emory University and Yahoo [15]. Moreover, since the toolkit is written in Cython, it is also optimized towards performance and can be operated together with deep learning frameworks such as TensorFlow or PyTorch. While NLTK provides many different extensions, spaCy provides only one single highly optimized tool for each task. The tool spaCy provides a built-in iterator for extracting noun phrases and is therefore easier to use than NLTK.

3 EXTRACTION PIPELINE OVERVIEW AND PREPROCESSING

In this section, our extraction pipeline will be introduced. It consists of two steps. First, statistical information was extracted with a rule-based approach. Then two different approaches were introduced to further extract the topics and experimental conditions. In the following sections, the different parts will be described in more detail.

3.1 Pipeline Overview

The pipeline contains three stages. The primary focus in this work is the extraction of the statistics, followed by the extraction of the topics and experimental conditions. From here on the terms aspect and topic are used interchangeably, as the original paper for ABAE used the term aspect.

The basic steps are:

- **Step 0: Preprocessing**: Takes a document, splits it into sentences and applies custom filters.
- **Step 1 Statistic Extraction**: Takes a sentence, checks whether supported statistics are present and extracts its type and values. Here, a rule-based approach with wrapper induction learning was applied.
- **Step 2**: In this step sentence topics and conditions of the reported statistics are extracted.
  - ABEA: Takes a sentence and categorizes it to a fixed number of aspects. It employs an unsupervised attention and autoencoder architecture.
  - GBCE: Takes a sentence and extracts relevant words or phrases. As step 1, this approach is based on active learning, but applied on a grammar tree instead of the raw sentence.

3.2 Preprocessing

Our approach expects a set of files in JSON format as used in the CORD-19 dataset, but it can be adapted to any reasonable format. Tools like Grobid can be used to obtain the correct format, if the files from the given dataset are given in PDF. The language of each sentence is then checked by langdetect[3]. If it differs from English it will be skipped. Afterwards, sentence splitting is applied using a simple regular expression: `\.\s?[A-Z]`. The reason this rule is applied and not already defined methods from state of the art libraries are used is that, in some cases, these methods tend to cut in between the sentence. This may imply that statistic is cut, since it often includes a ".". Patterns, like a statistic reported in APA style is susceptible to be cut out by the state of the art methods. We opted to handle two or even multiple sentences instead of potentially loosing structural information. In the following example, a typical statistical record within a sentence conform to APA style is shown. "*The results of the paired sample t-tests indicated that negative emotion after inducement was significantly higher than at baseline (t(56) = 13.453, p < .05)*". Especially the last part of the statistical record *[...]p < .05)* is susceptible to be cut. It is also possible that the sentence will not be split in the statistic record but somewhere else in the sentence, which might make it impossible to determine the experimental conditions and topic. Each sentence not containing digits is filtered out because they should not contain any statistics.

Furthermore, through the process of converting PDF or html files to JSON, some conversion errors may occur. One such sentence found in CORD-19 is (from [16]):

"*Inactivation at* $100 \propto C$ *was, however, complete within seconds (Duizer et al., 2004a) .The resistance of FeCV (in suspension) to inactivation by UV 253.7 nm radiation was reported to be highly variable.*"

The first observation made is, that $100 \propto C$ should be $100\ ^{\circ}C$. These kind of errors can also occur in statistics. This will make the process of identifying a pattern much harder, because, in this case, it is not to be expected to find a temperature notation written like this. Second, this is not one single sentence, it is actually two different sentences, but due to the lack of a white-space character after the "." of the first sentence, the splitting sequence did not detect the end of the sentence and therefor interprets these two sentences as one.

[3]https://pypi.org/project/langdetect/

This instance can be intercepted by slightly modifying the splitting pattern to make the white-space "`\s`" optional, like "`\s?`". However, it is possible, that some sentences can start with a digit or lower case character instead of an upper case character. Defining a pattern that will match all these cases could also lead to increased false positive rate and splitting in the middle of a sentence, corrupting the results. Thus, we did not apply it.

4 STEP 1: ACTIVE WAPPER FOR STATISTICS EXTRACTION

To extract the statistics of the preprocessed paper, a rule-based approach was applied. This approach is based on two set of rules, R^+ and R^-. R^+ determines all numbers that describes statistics in a sentence. R^- determines all numbers not describing statistics. A sentence is finally classified when each number is covered by at least one rule. We support common types of inferential statistics, namely *Pearson's Correlation, Pearson Spearman Correlation, Student's t-test, ANOVA, Mann-Whitney U Test, Wilcoxon Signed-Rank Test,* and *Chi-Square Test*, but the concept is transferable to arbitrary types. Statistic of a not identifiable or not supported type was summarized as *other* type.

For each statistic type i there is at least one (likely multiple) rule r_i in R^+. Each rule r_i has a set of sub-rules $\{s_1, \ldots, s_k\} = S_i$. The elements of R^+ consist of tuples of the form (r_i, S_i). The rules r_i are used to detect the different statistic types, such as a student t-Test or Analysis of Variance (ANOVA). The rules $s_j \in S_i$ are used to detect the different statistic parameters. For example, in *(t(29) = -1.85, p = .074)* the degree of freedom is 29 the p value is 0.074 and the significance value is -1.85.

Appropriate rules are obtained by learning them with active wrapper induction. The main loop of the learning process can be seen in Algorithm 1. This algorithm is iterating over one document, splitting it into sentences (line 6), which are considered relevant and is classifying the sentence into containing statistic or not. For the sentence containing statistics the respective statistic type is defined by the rule set R^+. If R^+ classifies a sentence as statistic with rule r_i, the respective sub-rules set S_i are applied to extract the statistic parameters (lines 11 and 12). If neither R^+ nor R^- classifies a number of a sentence, it is printed to the user. The user is then able to define and add a new rule to the respective rules set (line 21). This algorithm can be applied to a whole corpus of documents.

To give an example sentence, conform to the APA style: "*The independent sample t-tests indicated that there were not significant differences in the effect of ibuprofen 400 between males and females, (t(29) = -1.85, p = .074).*" First the R^+ rules will be applied. Therefore, the *t-test* match is found first by a rule like this: `(?P<ttest>\(t\s?\(\d+\)\s?=\s?\d+\.\d+ \, \s?[p,P] \s? <?=? \s? \d+\.\d+ \))`. The `?P<ttest>` defines the type of the statistic, here a Student's t-test. The match is *(t(29) = -1.85, p = .074)*. Out of this sub-sentence, the values of this statistic type will be extracted by using the respective subset rule. These values are stored in *statsRecord*. The corresponding sub-rules could be:

- `t\s?\((?P<doF>\d+)\)\s?=\s?-?\d+\.\d+\,\s?[p,P] \s? <?=?\s?\d+\.\d+`

Algorithm 1 ActiveWrapper for extracting statistics

```
Input: D // Document to be processed
Input: R+ // Set of tuples of positive stats extraction rules (with main rule, and sub-rules)
Input: R− // Set of negative rules
Output: L // Statistics extracted from D
L ← ∅ // Initialize empty output list
S ← D.split // Split D into a set of sentences
while S ≠ ∅ do
    s ← S.nextElement() // Process next sentence string
    // String-based processing of each s
    while s contains unclassified numbers do // String is not ""
        statsType, subR+ ← apply(R+,s)
        statsRecord, s ← apply(subR+,s)        ▷ Extracted matching stats of type statstype, and updated s (truncated up to the extraction of record)
        if statsType ≠ NONE then
            // Found a stats using R+, so add to output
            L.add(statsRecord)
        else
            // no stat found? get confirmation from R− rule
            nonStat ← apply(R−,s)        ▷ Return if confirmed non-stat rule successfully applied
            if nonStat = FALSE then
                // If neither R+ nor R− work on string ...
                Invoke( 'ask user' to add rule for s)
            end if
        end if
    end while
end while
```

This algorithm describes the procedure to learn the rule sets R^+ and R^- including the related subrules.

- `t\s?\(\d+\)\s?=\s?(?P<tstat>-?\d+\.\d+)\,\s?[p,P]\s? <?=?\s?\d+\.+`
- `t\s?\(\d+\)\s?=\s?-?\d+\.\d+\,\s?[p,P]\s?<?=?\s?(?P<pval>+\.\d+)`.

The sub-rules have the same structure as the normal rule, except that the different statistic parameters are tagged, for example `P<pval>` means that the following number is a p value. The extracted values from the example above are $[df = 29, t = -1.85, p = 0.074]$ together with the sentence. But the sentence still contains numbers in *ibuprofen 400*. When there is no R^+ rule left for this case, a corresponding R^- rule should match the 400, e. g. `[a-zA-Z]+\s\d+\s[a-zA-Z]+`. If there are no numbers left, which are not covered by some rule, the sentence is completed and the next sentence is processed.

5 STEP 2: EXTRACTING EXPERIMENTAL CONDITIONS

Result of Step 1 is a set of sentences, which are known to contain statistics. These sentences are investigated further in the second step, to extract the topics and experimental conditions.

The method used for extracting sentence topics is ABAE. It is an unsupervised machine learning algorithm which combines word embeddings into a sentence embedding via an attention mechanism and then compresses the information further with an autoencoder-like structure to create an aspect embedding.

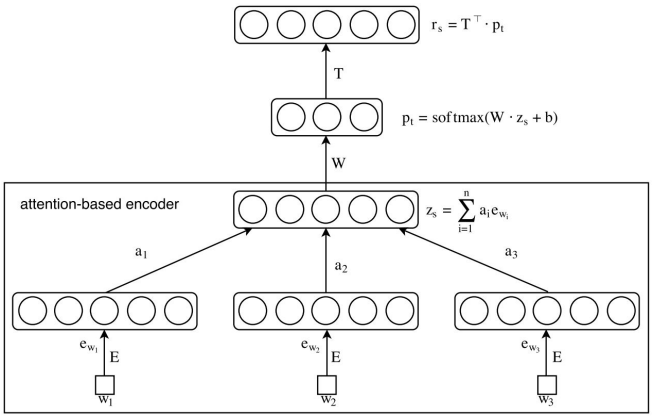

Fig. 2. ABAE structure, image taken from [2].

A grammar-based approach is used for extracting conditions. Hereby, the sentence exhibits a clear, structured pattern that is caused by following APA style. Therefore, an Active Wrapper approach to extract experimental data is applied, exploiting patterns that can be expressed with token rules. For natural language processing, the python module spaCy is used. Our expectation was that the grammar-based approach should work well on easy or common sentences. We give a more detailed explanation of both approaches in the following sections.

5.1 Unsupervised Aspect Extraction with ABAE

ABAE uses sentence-level input which fits nicely to our first step of extracting sentences containing statistics. It makes the assumption that there are K different aspects and that each sentence can be classified to one aspect. In the original paper on ABAE they only used small numbers for K, e. g., 14 for extracting aspects from restaurant reviews. This is reasonable in their case because there is only a limited number of relevant aspects in reviews over objects such as restaurants. In our case it is not clear if the number of aspects is limited because there can be arbitrary many different experimental setups. We trained ABAE with different values of K and tested which model delivers better results.

First the vocabulary size V and the word embedding dimension d of the underlying word embedding as well as the number of different aspects K have to be chosen. The input to the model is a sentence or set of words $s = \{w_1, \ldots, w_n\}$. Each word corresponds to a row of the embedding matrix $E \in \mathbb{R}^{V \times d}$ and can be transformed to a vector $e_{w_i} \in \mathbb{R}^d$. This word embedding describes the local context of the word. Figure 2 shows the whole architecture of ABAE, with this step corresponding to the bottom most arrows in the figure. Equation 1 shows how the sentence embedding z_s is computed from the word embeddings. This can also be seen in the center of Figure 2.

$$z_s = \sum_{i=1}^{n} a_i \cdot e_{w_i} \tag{1}$$

The weights a_i describe how important a word is for the meaning of the sentence. They are calculated by the attention mechanism shown in Equations 2 to 4. $y_s \in \mathbb{R}^d$ is the average word embedding of a sentence and describes the global context of a sentence. The matrix $M \in \mathbb{R}^{d \times d}$ in Equation 3 is a mapping between the global and local context.

$$a_i = \frac{exp(d_i)}{\sum_{j=1}^{n} exp(d_j)} \tag{2}$$

$$d_i = e_{w_i}^T \cdot M \cdot y_s \tag{3}$$

$$y_s = \frac{1}{n} \sum_{i=1}^{n} e_{w_i} \tag{4}$$

On top of the sentence embedding is an auto-encoder like structure, which can also be seen in the top part of Figure 2. The aspect probability vector p_t gets calculated like shown in Equation 5 with $W \in \mathbb{R}^{K \times d}$ and $b \in \mathbb{R}^K$.

$$p_t = \text{softmax}(W \cdot z_s + b) \tag{5}$$

Finally the sentence embedding is reconstructed from the aspect probability vector with the aspect embedding matrix $T \in \mathbb{R}^{K x d}$.

$$r_s = T^t \cdot p_t \tag{6}$$

The aim while training ABAE is to minimize the reconstruction error between r_s and z_s, and to maximize the difference between r_s and the average word embedding of any negative sample n_i. A negative sample is a sentence from the input data with a different aspect than the current sentence s. As ABAE is unsupervised neither the aspect of the current sentence nor the aspect of any other sentence is known before and during training, m negative samples are randomly drawn from the input data for each sentence and over multiple training epochs. Most negative samples should have had a different aspect than s. To implement this, a modified hinge loss is used that is proportional to $r_s n_i$ and negative proportional to $r_s z_s$ as shown in Equation 7.

To improve diversity of the learned aspects, a regularization U that promotes orthogonality of the aspect embedding matrix's rows is added. T_n is the matrix T with each row normalized to length 1 and I is the identity matrix. This regularization and it's combination with the hinge loss to the overall loss L are shown by the following equations:

$$J(\theta) = \sum_{s \in D} \sum_{i=1}^{m} max(0, 1 - r_s z_s + r_s n_i) \tag{7}$$

$$U(\theta) = \|T_n \cdot T_n^t - I\| \tag{8}$$

$$L(\theta) = J(\theta) + \lambda U(\theta). \tag{9}$$

Finally the most representative words of each aspect are extracted from the word and aspect embeddings and the aspects are manually inferred from those. For example in the original paper "*main dishes*" was inferred from the representative words "*beef, duck, pork, mahi, filet, veal*" and "*dessert*" from "*gelato, banana, caramel, cheesecake, pudding, vanilla*". "*main dishes*" and "*dessert*" where then mapped to the gold standard aspect "*Food*" as they had 14 inferred for 6 gold standard aspects. A sentence gets assigned one of those inferred aspects according to the aspect probability vector from Equation 5.

5.2 GBCE: Supervised Aspect Extraction using Noun Phrases

We apply a second Active Wrapper to learn the conditions extracted from the statistic sentences provided by step 1 as explained in Section 4. The following method is based on the assumption that the input consists of exactly one sentence containing all experimental conditions for a reported statistics. To extract experimental conditions, it is necessary to locate the parts of the sentence containing them.

According to spaCy[4], it is more promising to train a model for complex tasks. The general motivation for the GBCE is based on the fact that for training a model, labeled training data is a necessity. Since the CORD-19 dataset does not supply labels, we provide an alternative, namely rule-based matching. It is generally used when the text that needs to be processed follows a finite number of structured patterns, e. g., for specific syntax like HTML [17]. If this is the case, it is possible to create adjusted rules for tokens to extract experimental conditions.

5.2.1 Use Noun Phrases to Extract Experimental Conditions. A nominal phrase is a phrase, i. e., a syntactic, self-contained unit, whose core consists of a noun. In the sentence "*There is a positive statistically significant correlation between perceived knowledge and measured basic knowledge*" a noun phrase would be e. g. "*a positive statistically significant correlation*" with the core "*correlation*" whereas adjective and article are seen as dependent companions of the nominal head. The idea behind the method is that all information about experimental conditions contains such a noun. Therefore, all phrases that are not part of a noun phrase can be ignored when extracting statistical information.

spaCy for Noun Phrase Extraction. To further define what part of the sentence is also part of a noun phrase, the tool spaCy was used. It provides syntactical information about text. This includes general information about the text's topic, the meaning of the individual words in context or similarities between texts. Because of the difficulty of processing raw text, spaCy adds linguistic knowledge by providing a variety of annotations. The most important ones are the UnifiedPOS (UPOS) tag, the detailed/extended POS tag, e. g., that the verb is past tense or the noun is proper singular, and the syntactic dependency describing the relation between tokens, e. g., preposition and object of preposition. This is necessary to enable rule-based matching. The data structure spaCy is working on is a so called `Doc` object[5], which is a container of a sequence of tokens for accessing linguistic annotations. After tokenizing every word of the input sentence, the `Doc` object is processed. The default pipeline consists of a tagger, where each token is assigned a POS tag, a parser, adding dependency labels to aid natural language processing, and an entity recognizer. This makes it possible to put the individual words of a sentence into context and thus forms a tree structure.

[4]https://spacy.io/usage/rule-based-matching
[5]https://spacy.io/api/doc

To customize the default variant, it is possible to exclude default methods and add custom methods.

Navigating over the Sentence Tokens. The tool spaCy allows to iterate through the Doc object in two different ways as shown in Figure 3. On the one hand, it can be processed in sequential token order. This was mainly used for creating new rules as further described in Section 5.2.2. On the other hand, the Doc object can also be navigated following the parse tree. This was mainly used for the extraction of noun phrases.

5.2.2 Active Wrapper for Extracting Experimental Conditions. In the following, an algorithm is presented to extract experimental conditions of a sentence.

Preprocessing. First, the sentence given as input needs to be preprocessed. In GBCE, rules are only applied on noun phrases and not full sentences. Therefore, the preprocessing is especially important to identify the noun phrases correctly. For that purpose, additional processing steps are implemented. For preprocessing, all the brackets including their content are deleted, since brackets can interfere with the dependency parser. Additionally, since each piece of information about experimental conditions contain nouns, the point of interest can be reduced to noun phrases. Therefore, noun phrases are being identified, including their associated grammatical modifiers. We consider the following modifiers: numeric-, prepositional-, adverbial-, nominal-, appositional-, adjectival, adverbial clause-modifier and clausal modifier of nouns (adjectival clause). The tool spaCy is not capable of correctly processing quotation marks. To address this problem, if a noun phrase was identified inside quotation marks, the whole quotation gets included into the noun phrase. Also, spaCy interprets the usage of a semicolon as a new sentence. This would result in two separate parse trees, which our rules are not designed for. Since no experimental conditions were found after a semicolon, the rest of the sentence was excluded.

Rule sets. After preprocessing the input data, rules are learned with the goal of classifying information provided by noun phrases and extracting information. If a noun phrase can be classified as experimental condition, the information gets extracted and the noun phrase will not be considered further. In general, if no noun phrases are left to assign or there are no rules left to be applied, the program stops and outputs the results. According to these information, the user can decide whether the sentence got processed correctly or if a new rule needs to be added. The rules were divided into different rule sets depending on what kind of information can be extracted. We named them according to Section 4 since the functionality is analogously. The difference is, that instead of classifying numbers as statistic candidates they are confirming or removing noun phrases.

R- Rules. When using the active wrapper, it was possible to determine specific patterns that never included experimental conditions. These kind of patterns were exploited in the first rule set, *R-* rules. *R-* rules describe patterns that indicate noun phrases without relevant information regarding experimental conditions and can be removed. This includes, e. g., personal pronouns, "*we found*". Another rule in this rule set excludes aspects. This includes the case when the root of the sentence is not the main verb but instead a passive auxiliary. A passive auxiliary is a subclass of verbs that add functional or grammatical meaning to the main verb. Even though the target of the tool is not to identify the aspect of the sentence, it still makes it possible to exclude the noun phrase assigned as the aspect.

R+ Rules. This rule set is split in two sub-rule sets. The first one contains rules which, when matched, all experimental conditions can be extracted and no further rules need to be applied. These rules exploit the fact, that the English grammar often follows specific patterns. One such pattern when analyzing statistics is, that a sentence often includes comparative adjectives. Those are mainly used to compare differences between two objects. The basic pattern for this specific example is the following: "*Noun (subject) + verb + comparative adjective + than + noun (object),*" If no such rule can be applied, a second sub-rule set is applied for locating exactly one condition. An example is a rule that is identifying relative clauses, introduced by interrogative words. Relative clauses are non-essential parts of a sentence. They only add additional meaning to a noun phrase. If a relative clause is identified, it gets included into the noun phrase. Furthermore, when the relative clause is started by an interrogative word, the noun phrase is a condition. An example sentence could be: "*Participants who agreed that the COVID-19 outbreak was threatening their livelihood...*" . The assignment in rule sets allows to easily skip rules, which would not provide new information. The last rule that is always applied in the active wrapper when experimental conditions were found, is checking for enumerations. This rule recognizes whether such an enumeration is present, splits the elements, and stores them separately as individual conditions.

6 EXPERIMENTAL APPARATUS

In this section, we describe how we applied our system and evaluated the results. We evaluated each method separately. The evaluation methods are described in the following subsections.

6.1 Dataset

We use the Covid-19 Open Research Dataset (CORD-19). The CORD-19 dataset has been constructed to enable a basis to develop text and data mining tools that help the medical community answer high priority scientific questions, especially regarding the COVID-19 pandemic. This dataset has been created by a coalition of the White House and leading resource groups[6]. It includes around 200,000 scientific articles (21st September 2020) of which over 108.000 are scientific full text papers about COVID-19, SARS-CoV-2, and related corona-viruses. We preprocessed the dataset as mentioned in Section 3.2 including sentence splitting. Furthermore, we identified how many sentences potentially contain a test statistic. From the 108.802 full text papers in the corpus about 55% of all sentences contain at least a single digit, which were used as input for our approach.

6.2 Procedure - Statistic Extraction

To learn the R^- and R^+ rule sets, we applied the active wrapper approach from Section 4 on the CORD-19 dataset. Documents are processed in the order of how they are organized in the dataset, which is according to a random index. We stopped training after

[6]https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

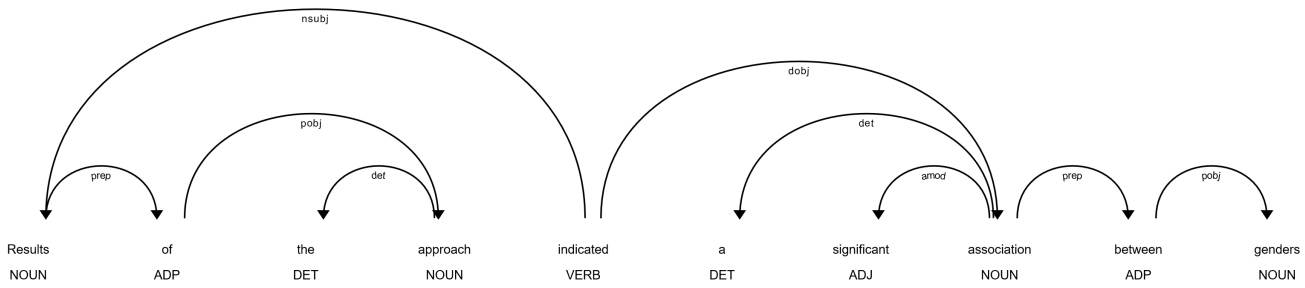

Fig. 3. Example of a parse tree sentence with experimental conditions, image created with spaCy visualization module.

500 documents and analyzed the results. To evaluate the statistic extraction, we took a random sample of 200 non-statistic and statistic sentences, except when there where less. In that case, we took all found sentences. We also did this for each statistic type twice. Once for the APA conform writing style and once for the non-APA conform writing style. We regard a rule conform to APA style, if all parameters are present and the formatting is correct. However we tolerate little derivations from the APA style formatting, i. e., $P = 0.07$ is not conform to APA, because the P is a capital letter and there is a leading zero before the ".". Nevertheless we regard it as APA conform, since the difference is quite low. An APA conform sentence was classified correctly if all attributes of the statistic could be extracted. On the other hand the non-APA conform sentences were classified correctly if just the type of statistic was correct. The sentences were extracted by the learned R^+ and R^- rules.

We then labeled these sentences manually to determine the true positives (tp), true negatives (tn), false negatives (fn) and false positives (fp). For the labeling we had two persons responsible and in ambiguous cases, they consulted each other to find consensus. On this labeled sub-sample of size ≤ 200 per class, we calculated our measures for each statistic type. The measure we used for evaluation are the precision by $prec = \frac{tp}{tp+fp}$ and a count of how many sentences were extracted in total. We used the combination of $prec$ and amount of extracted samples, because for some statistic types we could only obtain a small amount of samples. For these samples, the precision might not be expressive. Furthermore a coverage of our *R+* and *R-* was calculated, by taking a random sample of $10,000$ unseen documents and determining the proportion of uncovered sentences in these documents.

6.3 Procedure - ABAE

Multiple ABAE models were trained on different embeddings, data and with different numbers of aspects. In the following section we explain the parameter choices for the different models, their training, as well as how the models were evaluated. Three different datasets were used, these are:

- **cord**: preprocessed CORD-19 dataset
- **supp-sen**: extracted sentences containing only supported [7] statistics
- **all-sen**: extracted sentences containing any statistics, which is the union of supp-sen with other

Data	unique words	total words	sentences
cord	1,400,093	238,582,456	16,141,291
all-sen	45,071	1,467,485	113,147
supp-sen	7,189	81,936	6,092

Table 1. This table shows how many words and sentences are contained in the different datasets after preprocessing.

Standard NLP preprocessing like stopword removal and lemmatization was used on all three datasets. Additionally, langdetect[8] was used on the CORD dataset to filter non English sentences. 4.2% of the sentences were removed by the language filter. The removed data among others contained sentences in German, French, Spanish, and Dutch as well as some parse and detection errors, e. g., in equations, citations and abbreviations. The number of unique and total words as well as the number of sentences of the datasets are shown in Table 1.

Embedding. Three different Word2Vec (W2V) embeddings were trained, one each on cord, supp-sen, and all-sen.

For the cord embedding, the number of words in the embeddings was limited to about $50,000$ by choosing 100 as the minimum word frequency. This was done because most of the infrequent words do not contain information that the embedding can learn as well as to reduce the model size. The most frequent $50,000$ words cover about 97% of the total words in the cord dataset. This is also plotted in Figure 4.

The other two datasets contained less than $50,000$ words. For all-sen the minimum word frequency was left on W2V's standard of 5 which includes about 96.4% of the total words. For supp-sen, the minimum word frequency was chosen to be 3, which covered 94.1% of the total words. This is a trade-off between giving W2V enough examples for every word included in the embeddings and covering enough words of the dataset to have a meaningful result after applying the embedding. For supp-sen, the standard setting of 5 would have covered only 88.9% of the total words.

The W2V[11] embeddings with dimension $d = 200$ were trained with the skip-gram algorithm and a window size and negative sampling of 5.

[7] Student's t-test, Pearson Correlation, Spearman Correlation, ANOVA, Mann-Whitney U, Wilcoxon Signed-Rank, Chi-Square

[8] https://pypi.org/project/langdetect/

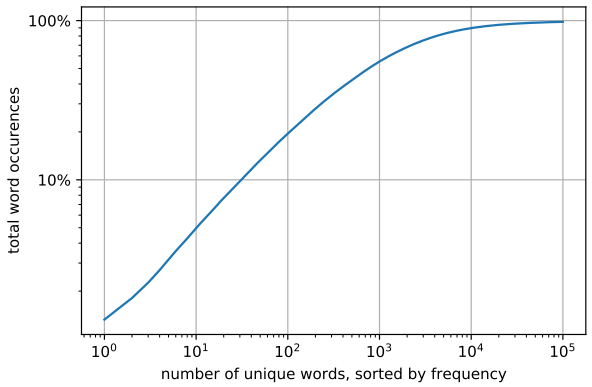

Fig. 4. Total word coverage over number of unique words to 100k words on CORD dataset after preprocessing. We chose a cutoff of 50k as that covers about 97.0% of the words and very little return on higher values, e. g. 98.2% at 100k.

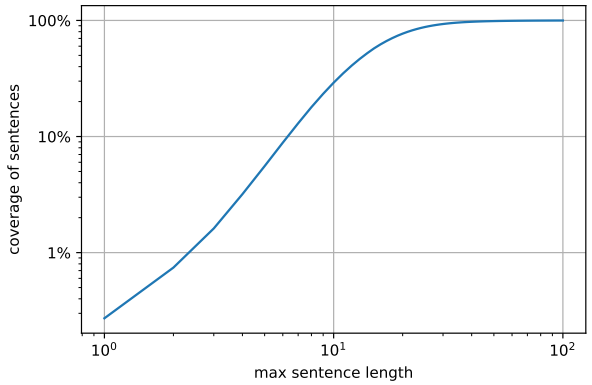

Fig. 5. Coverage percentage over sentence lengths to 100 on cord dataset after preprocessing. We chose a cutoff of 70 as that covers 99.5% of sentences. Most of the very long sentences contain parse or sentence splitting errors.

Model Hyperparameters. The longest supported sentence length was chosen to be 70, all shorter sentences got padded to that length. The relation between the longest sentence length and coverage of the CORD dataset can be seen in Figure 5.

The number of negative samples m was set to 20. Different values for the number of Aspects K were tested, namely 15, 30, and 60. The weight of the regularization in the loss function was set to $\lambda = 1$ like in He, Lee, Ng, *et al.* [2].

Training. The embedding Matrix E got initialized with one of the pre-trained embeddings and was then fixed during training of the other parameters of ABAE. The aspect embedding matrix T got initialized with the centroids of a k-means run on the word embeddings. The matrices M, W, and the vector b were initialized randomly. M, W, b and T were trained with Adaptive moment estimation (Adam). Adam was used for 50 epochs with a learning rate of 0.01, epsilon of 10^{-7}, a batch size of 10 on the smaller and 32 on the larger dataset, and all other parameters left on the standard setting. Like done in the ABAE paper the model after the epoch with the smallest loss was saved and evaluated.

The models were trained on two different training datasets. One set of models was trained on supp-sen, the other set of models was trained on on all-sen. Overall, the models were trained with all combinations of the three different embeddings mentioned above, two different training datasets and with the values for k from $\{15, 30, 60\}$. The only exception was the combination of embedding trained on supp-sen with the whole model trained on all-sen as the training data would contain many unknown words for the embedding.

After training, the aspects were inferred manually from the most representative words as described at the end of Section 5.1. If those representative words did not contain any concise groups of words, the aspect was set to "miscellaneous" which will always be evaluated as wrong.

Evaluation. For evaluation, 200 sentences from supp-sen were used. They were evenly sampled from 100 APA conform and 100 non-APA conform sentences. All models were evaluated on the same 200 sentences by manually checking if they extracted a correct aspect. This was done by agreement of two different reviewers, if their evaluation differed it was discussed to reach agreement.

6.4 Procedure - GBCE

Training. The GBCE was trained on 130 sentence from a pre-extracted list of sentences further described in Section 4 from the CORD-19 dataset. To learn the R- and R-rule sets, we applied the active wrapper approach from Section 5.2.2 and went through each sentence manually checking if the experimental conditions were extracted correctly. If this was not the case, an already existing rule was adapted, a completely new rule was created and added or specific words that often occurred were added to the bag-of-words.

Evaluation. For evaluation of the GBCE, the same sentences were used as for the evaluation of ABAE. We manually checked the same 100 APA conform and 100 non-APA conform randomly selected sentences if they extracted the right conditions. This was done by agreement of two different reviewers. For further evaluation of the tool, the main reasons for failure were summarized in five categories and the amount of their appearances were depicted.

- **Reason 1** describes errors when creating the Doc object. These errors include adding of false information in form of PoS-tags or errors in the dependency parser. These mainly occurred due to grammatical mistakes in the sentence structure or due to conversion errors from JSON format.
- **Reason 2** describes errors due to unusual sentences; For example some sentences do not include verbs. Since spaCy builds the parse tree with a verb as root, a missing verb also affects the grammar-based process of finding conditions.
- **Reason 3** describes causes rooted in preprocessing errors. This includes errors in the extraction of sentences containing statistics and in creating noun phrases.

- **Reason 4** describes errors in the extraction of statistics. If a statistic was not excluded before building the dependency parser, the statistic often misleads the dependency parser splitting the sentence at the statistic.
- **Reason 5** describes missing/ wrong rules. This leads to cases, where GBCE could not extract the conditions due to missing training for specific type of statistics or patterns.

7 RESULTS

The following section presents the results of our experiments. The first subsection is about the statistic extraction with the rules learned by the Active Wrapper approach. The second subsection covers the topic extraction and the third the experimental conditions extraction.

7.1 Statistics extraction

We applied the rule-based learning approach, introduced in Section 4, on 500 documents and learned 85 $R+$ rules and $1,425$ $R-$ rules. After learning, we checked the sentence coverage and saw that of all sentences in the corpus 5% were not covered by a R^+ or R^- rule. Therefore, after learning rules on 500 documents, which contained $38,099$ sentences, most of the sentences in the corpus were covered.

In Table 2, one can see how many sentences containing statistics were extracted from the whole CORD-19 dataset. We applied our model on these sentences and manually evaluated the results. In Table 3, the calculated precision values for each statistic type are shown. We achieved a precision of 100% in all APA conform cases except in the ANOVA and Wilcoxon Signed-Rank test. There we did not extract any APA conform sentences to evaluate them. In the non-APA case the precision scores were slightly lower than in the APA case and ranged from 91% to 100%. The Wilcoxon Signed-Rang test could not be evaluated since we did not extract any sentences. The smallest precision of extracted sentence was in the non-APA Student's t-test. The amount of covered sentences was 95%. We also randomly sampled 200 of all uncovered sentences. Of these sentences, 21 contained some kind of statistic, 157 were without statistic. 22 of the sentences contained a parse error, independent of them containing statistics or not. An example for such a parse error is "Notably, however, CD8a - DCs and also pDCs can cross-prime CD8 + T-cell responses under certain conditions (102) (103) (104) 123)". However the original sentence looks like "[...] responses under certain conditions (102–104, 123)"[9].

To evaluate the R^- sentences, we took an equally sized random sample of 200 R^- matches from the whole corpus except the first 500 documents we used for training. It was determined that 99.5% of the non statistic sentences are correctly classified.

In Tables 4 - 9, the amount of missing parameters from all extracted samples for every non-APA conform statistic are recorded. In the diagonal of the table, one can observe how often a parameter was missing on its own. In the other entries one can see how often a pair of parameters was missing, e. g., in Table 4 the entry in row degree of freedom (doF) and column t-value (tval) shows how often doF and tval were missing together. The column marginal shows how often a value was missing alone or in combination with another parameter.

[9]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4603245/

Statistic	APA conform	non-APA conform
Student's t-test	608	179
Pearson Correlation	113	4,962
Spearman Correlation	1	528
ANOVA	0	9
Mann-Whitney U	2	34
Wilcoxon Signed-Rank	0	0
Chi-Square	14	31
not supported	not applied	19,151
not determinable	not applicable	87,904

Table 2. This table shows how many statistics of each type were extracted from the whole CORD-19 dataset. Other is divided in not supported e. g. odds ratio, IQR etc., which are not investigated here; and not determinable like solely reported p value where the type of statistic could not be decided.

Statistic	APA conform	non-APA conform
Student's t-test	1.0	0.91
Pearson Correlation	1.0	0.98
Spearman Correlation	1.0	1.0
ANOVA	n/a	1.0
Mann-Whitney U	1.0	1.0
Wilcoxon Signed-Rank	n/a	n/a
Chi-Square	1.0	0.97
other	-	0.95

Table 3. This table shows the precision values for the different statistic types. The precision has been calculated for each statistic type on 200 samples. If less than 200 samples were extracted, the precision has been calculated on the respective amount of extracted samples.

Missing Parameter	doF	tval	pval	other	Sum
Degree of freedom (doF)	0	75	21	1	97
t value (tval)		0	0	1	76
Significance level (pval)			0	1	22

Table 4. Count of missing parameters from extracted non-APA conform Student's t-tests samples. Calculated on a total of 179 samples, so all extracted non-APA conform t-tests. The column "other" refers to a combination of all missing parameters doF + tval + pval.

Missing Parameter	doF	rs	pval	Sum
Degree of freedom (doF)	527	0	1	528
Spearman correlation (rs)		0	0	0
Significance level (pval)			0	1

Table 5. Missing parameters from extracted non-APA conform Spearman Correlation samples. Calculated on 528 samples.

7.2 ABAE

In this subsection the evaluation results of ABAE are presented. We had five different embedding and training data combinations. For

Missing Parameter	doF	r	pval	other	Sum
Degree of freedom (doF)	4961	0	0	1	4962
Pearson Correlation (r)		0	0	1	1
Significance level (pval)			0	1	1

Table 6. Missing parameters from extracted non-APA conform Pearson Correlation samples. Calculated on 4, 962 samples. The column "other" refers to a combination of all missing parameters doF + pval + r.

Missing Parameter	doF	fval	pval	r	other	Sum
Degree of freedom (doF)	0	0	0	0	2	2
F value (fval)		0	0	7	2	9
Significance level (pval)			0	0	1	1
Effect size (r)				0	2	9

Table 7. Missing parameters from extracted non-APA conform ANOVA samples. Calculated on 9 samples. The column "other" refers to a combination of missing parameters of doF + fval + r and doF + fval + r + pval.

Missing parameter	U	z	pval	r	oher	Sum
U	4	0	0	0	13	17
z		16	0	0	13	29
Significance level (pval)			0	0	0	0
Effect size (r)				0	13	13

Table 8. Missing parameters from extracted non-APA conform Mann-Whitney U samples. Calculated on 34 samples. The column "other" refers to a combination of missing parameters of U + z + r.

Missing Parameter	χ^2	N	pval	V	other	Sum
χ^2-value (χ^2)	0	0	0	0	2	2
Observations (N)		0	0	1	2	3
Significance level (pval)			0	0	2	2
V				28	2	31

Table 9. Missing parameters from extracted non-APA conform Chi-Square samples. Calculated on 31 samples. The column "other" refers to a combination of all missing parameters of χ^2 + N + pval + V.

each of those, we trained three models, one each with $K = 15, 30$, and 60. As explained in Section 6.3, the used datasets were preprocessed versions of CORD-19, the sentences containing statistics that is explicitly supported by step 1 (supp-sen) and all sentences containing statistics where either the type of statistic could not be identified or is one of the not supported types (all-sen). Those models were then evaluated on 100 sentences that strictly followed APA style ("Result APA") and 100 sentences that contained statistics but not strictly following APA style ("Results non-APA"). For each of the 200 sentences, all models were evaluated by both reviewers separately. Then, the results for the sentence were compared and, if different, discussed to reach agreement. The number of correct outputs for all models can be seen in Table 10. The model that performed best was created with $K = 30$, an embedding trained on supp-sen and was then trained on supp-sen. It is marked bold in the table.

emb	train	K	Result APA	Result non-APA
supp-sen	supp-sen	15	33	31
supp-sen	**supp-sen**	**30**	**75**	**73**
supp-sen	supp-sen	60	28	28
all-sen	supp-sen	15	51	57
all-sen	supp-sen	30	48	49
all-sen	supp-sen	60	58	50
all-sen	all-sen	15	46	41
all-sen	all-sen	30	7	22
all-sen	all-sen	60	49	43
cord	supp-sen	15	38	37
cord	supp-sen	30	42	44
cord	supp-sen	60	62	54
cord	all-sen	15	57	56
cord	all-sen	30	44	37
cord	all-sen	60	33	46

Table 10. Number of correct ABAE outputs. Evaluated on 2 times 100 sentences containing statistics, 100 that strictly followed APA style (Result APA) and 100 that did not (Result non-APA). The first three columns shows which model was used. The first abbreviation specifies the embedding dataset, the second one the training dataset, and the third one the number of aspects K.

7.3 GBCE

In this subsection the results of GBCE are presented. We build the Active Wrapper by manually analyzing 130 sentences which resulted in 35 rules for GBCE. We started by comparing sentences and tried to find similarities in their syntax. We were able to discover a high amount of comparing adjectives as indicators for experimental conditions. This has prompted us to build several rules about this pattern. The results of our evaluation, the number of correct outputs and reasons for errors, are depicted in Table 11. The evaluation was done by two reviewers in the same way and on the same sentences as for ABAE. The only differences were, that there is only one model/result for each sentence. Every time the experimental conditions were not extracted correctly, we counted the number of occurrences and also identified the reason for its failure. In several cases, there was a combination of different reasons. Therefore, the sum of occurrences of reasons is higher than the amount of incorrect cases.

8 DISCUSSION

Main Results. As our results show, the statistics extraction achieves quite a high precision. Especially in the APA conform writing style, we reached a precision of 1.0, which shows the high quality of the rules to APA conform patterns. It also shows that the APA style definitions are very distinctive in terms determining the type of the statistic. Unfortunately, we were unable to extract many statistics of every type. As our results in Table 2 show, also in non-APA we did not find a Wilcoxon signed rank test. In order to investigate this observation, we manually wrote one extraction rule for Wilcoxon tests, but it could not extract anything on the whole dataset. This is maybe due to the fact that there are none or that they are written in a unusual manner and we classified them as other. According to

GBCE	Result APA	Result non-APA
correct	46	30
reason 1	4	5
reason 2	10	3
reason 3	9	12
reason 4	18	2
reason 5	25	47

Table 11. Number of correct GBCE outputs and as explained in Section 6.4, the number of reasons for each category. In several cases, a combination of different reasons of failure were the case. Evaluated on 2 times 100 sentences containing statistics, 100 that strictly followed APA style (Result APA) and 100 that did not (Result non-APA). Reason 1 describes errors from spaCy when creating the Doc object, Reason 2 are errors due to unusual sentences, Reason 3 describes preprocessing errors, Reason 4 describes errors in the extraction of statistics and Reason 5 occur due to missing/ wrong rules.

Weissgerber, Milic, Winham, *et al.* [18] these kind of tests are not commonly used, therefore this is quite plausible. In Weissgerber, Garcia-Valencia, Garovic, *et al.* [19] they manually analyzed 328 paper regarding their statistic writing style. They found many incomplete statistic reports in their sample from the PubMed dataset. Since our dataset is also from the field of life science and we extracted a lot more non-APA conform statistics, we conclude that it is quite uncommon in life sciences to strictly follow this writing style.

Regarding aspect extraction with ABAE, we found neither a pattern that models with lower or higher values of k nor models with a specific embedding or trained on a specific dataset generally performed better than the other ones. But, as expected, each model performed similarly on the non-APA compared to the APA evaluation dataset. One reason of the relatively high percentage of correct answers is that most models had at least one “statistics” aspect. That is a correct result but not useful for our use-case, as that applies to every sentence that made it through step one of STEREO. We did not evaluate whether some models did output mainly “statistics” and others mainly ’useful’ aspects, but overall roughly half of the correct answers were “statistics”.

The basic idea of GBCE was that sentences containing statistics mostly follow a common sentence structure and should therefore be uniquely determinable by a finite set of rules exploiting those patterns. However, we encountered fundamental problems that were not so easily solved. Our initial hypothesis was largely accurate, but deviations occurred more frequently than expected. In addition, differences between statistical test types were observed. For example, correlations seemed to followed a common pattern more often, while chi-square tests did not. Furthermore, sentences containing multiple statistics were cropped in preprocessing. This caused problems, if the context of the sentence was only comprehensible through neighboring sentences or if the sentence was not grammatically correct. One example for this behavior is: "The results show, that female participants used national newspapers (STATISTIC) highly significant less than male participants and international sources (STATISTIC) and YouTube (STATISTIC)". In those cases, it was not possible to automatically distinguish between aspects and experimental conditions. However, a clear distinction could be made between APA and non-APA conform results. It was noticeable that non-APA conform sentences were mainly longer with a more complex structure and also contained more statistics per sentence. This is, as depicted in Table 6.4, particularly evident from the higher number for Reason 3 and Reason 5.

Threats to Validity. For some statistic types like Pearson correlation and Student’s t-test, we could extract many samples. For these statistic types the results are quite confident. On the other hand for statistic types like ANOVA where we just extracted 9 samples, the results could be not representative enough. Nevertheless the metrics are similar to the statistic types with more samples. Therefore, it is plausible to assume that the results transfer to the types with few samples, too. A different problem was the conversion from PDF to JSON as well as different writing styles of statistical parameters, for example chi-square was written or got parsed as: chi-square, χ^2, Xˆ2, X2. To address this problem we learned additional rules to include special characters and different writing styles.

In aspect extraction with ABAE, there are two steps where an error could occur. The first one is choosing the inferred aspect and the other one is evaluating, whether a model found the right aspect for a sentence. We do not think either is a big issue, because all unknown terms and abbreviations were manually looked up and evaluation was done by consent of two people.

For GBCE, the rules were created on the basis of the Collins Dictionary [10] in cooperation with an anglistics student. Since the evaluation procedure for GBCE did not vary from the evaluation of the aspect extraction, errors in GBCE evaluation did probably not occur as well. Especially, since the test cases were selected randomly, the overall results should generally fit to all possibilities. Also, we did not evaluate how the accuracy of GBCE differed between statistical test types. Therefore, the results could deviate between different datasets with statistical test types not that common in the CORD-19 dataset.

Generalization. It is possible to use our tool, if the dataset can be processed in such a way, that the files are represented in JSON format. Furthermore our tool can be applied to any dataset, independent of the domain (e. g. psychology, medicine, physics, etc.). Since APA is a common standard in different disciplines the rules should transfer to these too, including the non-APA writing styles already covered by the rules. However, it would be beneficial to fine-tune the existing rule sets on a dataset from a new domain. Especially the R^- rule set contains a lot domain-specific rules. The existing rules match already 95 percent of the unobserved sentences; which is expected, because although there are different writing styles for statistics, it still follows a set of common rules, even if the writing is non-APA conform. Additionally, the extraction quality for R^+ and R^- has been manually verified, one may conclude that the same quality of extraction is reached on the unobserved sentences.

Regarding the generalization of aspect extraction, the same technical restriction (JSON format) as just described for Step 1 apply. If the dataset covers a different domain, then the embeddings and

[10] https://grammar.collinsdictionary.com/grammar-pattern

models have to be retrained from scratch. This means there have to be enough example sentences in the dataset to train the embedding and models. If there is only a small number of example sentences, one could try a more general embedding, either like we did with the embedding trained on the whole CORD dataset or a publicly available, most likely non domain specific, embedding. That means if there is enough data for training, in principal this approach can be applied to any similar problem. The other limiting factor is time investment into the inferred aspects and the evaluation. The number of inferred aspects is equal to the number of different models times the average number of aspects per model. That means, if one wants to train a lot of different models to find the best parameter settings or models with a high number of aspects, enough time has to be planned for those steps.

On the one hand, if the application and available data allows for a supervised approach, that would probably be easier to implement. On the other hand ABAE or most other unsupervised approaches can find unexpected things in the data. In our case, after doing some test runs with ABAE, we found that contrary to our expectation there was a not insignificant percentage of non English papers in the dataset(4.2% of the sentences), which lead to the inclusion of the language filter.

GBCE in general is a rule-based approach. Since the different statistical test types and the English grammar including their structural patterns do not vary between domains, a generalization should be possible without further adjustments. The only restriction is that the language must be English.

Practical Impact and Future Work. Even for statistics perfectly written in APA style, typing errors can occur. Because of such errors, it is possible that our defined pattern for the specific statistic type might not find a match. It might increase the amount of found statistics if the pattern is defined in such a way, that it can comprehend typing errors or even parse errors. Thus, our tool is in general robust to these kind of errors. This is a design feature to increase recall. Additionally, we are explicitly able to distinguish the extracted statistic in APA and non-APA conform classification, which offers new possibilities for applications, e. g. reporting this back as feedback to the authors.

On the other hand, generalizing such a pattern is not trivial, since tolerating typos could also lead to an increase in false positives. To provide an aid in the learning process of the adaptive wrapper, one could implement an automatic test after creating a R- rule. This new R- rule will be applied on a predefined R^+ sentence set, to check if the rule would classify the sentences correctly. This way, the learn process would be supported to not overgeneralize and include false positives. Possible future work could also try to find a minimal set of rules. During the process of active wrapper induction arise many rules that are completely covered by another rule or a subset of other rules. These rules could be removed. Since the equivalence of regular expressions is NP hard, this task is challenging.

Regarding ABAE, there are some improvements that could be implemented for our tool. The whole experiment could be redone with more effort put into the dataset for things like sentence splitting, filtering of 'bad' sentences and to check, and adapt the language filter in regards to domain specific technical terms. One could also train models with a wider range of parameters, some that we touched like the number of aspects and different datasets as well as some that we did not cover like experimenting with the (word) embedding size or trying a different optimizer. The evaluation could also be expanded to separately consider the "statistics" aspects as well as to calculate the output distribution of the models to check if all outputs are used to an equal amount.

Regarding GBCE, the use of noun phrases as basis for reducing the point of interest was in general successful. However, one could think about further adding rule sets, specifically for the different types of tests. On the one hand, this should achieve better results. On the other hand, the success depends on the re-usability of the rules that can be applied for every test. One reason this may be counterproductive is the possible need for a high number of specifically customized rules. Since a purely rule-based approach performs well for structured patterns like IP addresses or URLs, it may be possible, that creating a labeled set of data for the training of a neural network could be a better trade-off than the creating rules for a rule-based approach. Especially since there are more differing structural patterns than expected.

9 LESSONS LEARNED

One problem that appeared in the statistic extraction learning phase for our rules is, that some sentences contained syntactic deviations from the original papers, which probably happened due to parsing errors from PDF to JSON. Problematic is that the kinds of parsing errors differ from paper to paper. Some errors are caused, because a single uni-code character could not be correctly translated. Some of these errors appeared often and could potentially have an impact on detecting statistics. One of these errors was that a lower case L (l) has been parsed into the digit 1. In the statistic notations from our supported statistics, we would not expect a lower case L at all and especially not instead of a numerical value. Therefore, if this parsing error would happen in a statistical record, the sentence would not be detected as $R+$. However, while learning with the active wrapper, we did not find any statistics containing such a parse error. Another error that appeared in statistical records is that for negative values, the minus character (-) would not match, because through the parsing process the character has been transformed into another Unicode character which looks almost the same. However, we were able to fix this issue by allowing alternative variants of the - in their respective Unicode encoding. Furthermore, in some documents, the citation syntax could not be properly parsed, so that instead of a digit inside a bracket *[...]*, e. g. *[7]*, only the digit would be printed. The digit was then concatenated to the beginning of the next sentence, so that our pattern for splitting sentences could not detect the ending of the sentence, since we normally do not expect a sentence to start with a digit. We also observed a similar behavior, when the original papers did contain line numbers. Overall, this issue could be technically challenged, but it is very cumbersome and labor intensive.

For some sentences containing a statistical record, it is not possible to extract the respective aspects, e. g. "*The subjects were found to be significantly (t (263) = 25.04, p<0.001)*". However the original sentence as it is contained in the CORD-19 dataset is "*The subjects*

were found to be significantly (t (263) = 25.04, p<0.001) overweight (23.01 ± 16.82 kg) and had a mean excess of 4.95± 16.82 kg of fat.". We made the assumption that the record is located at the end of the sentence, which would be conform with APA style. After a statistic is found, we only extract the part of the sentences up to where the statistic is located. However, in the original sentence, the aspect is located directly after the statistical record and thus not present in the sentence we have extracted. Although every variables of the statistical record of a t-test are found (i. e., the t-value, df, p-value ...), this sentence may not contain the information about the experimental conditions or topic. With our current model, we do not preserve this contextual information, since we work on a single sentence level the preceding and succeeding sentences are not considered.

In the active wrapper approach for learning statistical records, after loading a document, it will be checked in which language the paper is written. If another language than English is detected, the document will be skipped. This is done using the python library langdetect. This routine does not work perfectly, i. e. "Viruses are unique in nature" will be detected as French with a confidence of> 99.9%. However, in general the language detection tool has a high accuracy 0.845. [11]. Thus, we do not assume that this is a large problem.

Specifically for GBCE, it is sometimes not enough to have one sentence as input, since the context is not clear without the additional information provided by neighboring sentences. For example, in the sentence "*Increased number of OUCC patients received antibiotics within* 60 *minutes after algorithm implementation*", it is not clear without contextual information, if "OUCC patients" is a condition or if the full example is an aspect. Furthermore, a distinction in the results between different statistical test types could be of value to further increase the certainty of a high generalizability of GBCE.

10 CONCLUSION

We have presented STEREO, a tool to analyze and extract sentences containing statistics from scientific papers. As we have shown in our results, finding and extracting sentences containing statistics with our regex-based active wrapper worked fairly well. As explained in the discussion, there were some issues with analyzing these extracted sentences further. Some of those could be improved with more effort, but we think that using or creating a labeled dataset and applying a supervised approach could lead to more promising results in step 2.

ACKNOWLEDGMENTS

Also, we thank Johannes Keller for providing us his knowledge as an anglistics student for locating grammatical patterns which are the basis for the rules created.

REFERENCES

[1] M. B. Nuijten, C. H. J. Hartgerink, M. A. L. M. van Assen, S. Epskamp, and J. M. Wicherts, "The prevalence of statistical reporting errors in psychology (1985-2013)," Behavior Research Methods, 48(4), 1205-1226, 2016. DOI: 10.3758/s13428-015-0664-2.

[2] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, Eds., Association for Computational Linguistics, 2017, pp. 388–397. DOI: 10.18653/v1/P17-1036.

[3] P. Lopez, "GROBID: combining automatic bibliographic data recognition and term extraction for scholarship publications," in *Research and Advanced Technology for Digital Libraries, 13th European Conference, ECDL 2009, Corfu, Greece, September 27 - October 2, 2009. Proceedings*, M. Agosti, J. L. Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, Eds., ser. Lecture Notes in Computer Science, vol. 5714, Springer, 2009, pp. 473–474. DOI: 10.1007/978-3-642-04346-8.

[4] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and L. Bolikowski, "Cermine: Automatic extraction of structured metadata from scientific literature," *International Journal on Document Analysis and Recognition (IJDAR)*, 2015. [Online]. Available: https://doi.org/10.1007/s10032-015-0249-8.

[5] L. Foppiano, L. Romary, M. Ishii, and M. Tanifuji, "Automatic identification and normalisation of physical measurements in scientific literature," in *Proceedings of the ACM Symposium on Document Engineering 2019, Berlin, Germany, September 23-26, 2019*, S. Schimmler and U. M. Borghoff, Eds., ACM, 2019, 24:1–24:4. DOI: 10.1145/3342558.3345411.

[6] Q. Liu, Z. Gao, B. Liu, and Y. Zhang, "Automated rule selection for aspect extraction in opinion mining," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, Eds., AAAI Press, 2015, pp. 1291–1297. [Online]. Available: http://ijcai.org/Abstract/15/186.

[7] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31, 1998*, R. Agrawal, P. E. Stolorz, and G. Piatetsky-Shapiro, Eds., AAAI Press, 1998, pp. 80–86. [Online]. Available: http://www.aaai.org/Library/KDD/1998/kdd98-012.php.

[8] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Double embeddings and cnn-based sequence labeling for aspect extraction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, I. Gurevych and Y. Miyao, Eds., Association for Computational Linguistics, 2018, pp. 592–598. DOI: 10.18653/v1/P18-2094.

[9] G. Karamanolakis, D. Hsu, and L. Gravano, "Leveraging just a few keywords for fine-grained aspect detection through weakly supervised co-training," 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 2019, Hong Kong, China,*

[11] According to https://towardsdatascience.com/benchmarking-language-detection-for-nlp-8250ea8b67c

November 3-7, 2019, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Association for Computational Linguistics, 2019, pp. 4610–4620. DOI: 10.18653/v1/D19-1468.

[10] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015. arXiv: 1503.02531. [Online]. Available: http://arxiv.org/abs/1503.02531.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781.

[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *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)*, J. Burstein, C. Doran, and T. Solorio, Eds., Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. [Online]. Available: https://doi.org/10.18653/v1/n19-1423.

[13] S. Angelidis and M. Lapata, "Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Association for Computational Linguistics, 2018, pp. 3675–3686. DOI: 10.18653/v1/d18-1403.

[14] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1373–1378. [Online]. Available: https://aclweb.org/anthology/D/D15/D15-1162.

[15] J. D. Choi, J. Tetreault, and A. Stent, "It depends: Dependency parser comparison using a web-based evaluation tool," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 387–396.

[16] E. Duizer and M. Koopmans, "Tracking emerging pathogens: The case of noroviruses," in. May 2006, pp. 77–110, ISBN: 9781855739635. DOI: 10.1533/9781845691394.1.77.

[17] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, ser. Data-Centric Systems and Applications. Springer, 2007, ISBN: 978-3-540-37881-5. DOI: 10.1007/978-3-540-37882-2.

[18] T. L. Weissgerber, N. M. Milic, S. J. Winham, and V. D. Garovic, "Beyond bar and line graphs: Time for a new data presentation paradigm," Plos Biology, 2015. DOI: 10.1371/journal.pbio.1002128.

[19] T. L. Weissgerber, O. Garcia-Valencia, V. D. Garovic, N. M. Milic, and S. J. Winham, "Meta-research: Why we need to report more than 'Data were Analyzed by t-tests or ANOVA'," eLife, 2018. DOI: 10.7554/eLife.36163.