FrESH – Feedback-reliant Enhancement of Subjective Content Descriptions by Humans

Magnus Bender[1,*], Kira Schwandt[1], Ralf Möller[1] and Marcel Gehrke[1]

[1]University of Lübeck, Institute of Information Systems, Ratzeburger Allee 160, 23562 Lübeck, Germany

Abstract

An agent in pursuit of a task may work with a corpus containing text documents. To perform information retrieval on the corpus, the agent internally maintains a model of the documents in the corpus. This model may contain annotations such as Subjective Content Descriptions (SCD)—additional data associated with different sentences of documents. In a our scenario, a human interacts with the information retrieval agent: The human sends a query to the agent, the agent uses its internal model to calculate a response and returns this response. However, the response may contain erroneous parts. Such errors, like faulty SCDs, may be send back to the agent by the human as feedback. Then, the agent can incorporate the feedback to improve its internal model. However, removing a faulty association of a sentence with an SCD in a previously trained model is a difficulty task—often the model needs to be retrained from scratch. To circumvent this, this paper presents FrESH an approach for Feedback-reliant Enhancement of Subjective Content Descriptions by Humans. Using FrESH the model keeps fresh and maintained with human feedback.

Keywords

Subjective Content Descriptions (SCDs), Text Annotation, Information Retrieval Agent, Incorporate Human Feedback, Incremental Model Adjustment, Information System

1. Introduction

Typically, in machine learning a model is trained to then perform tasks on the model instead on the data. Training a model is an expensive task. However, if data points later on have to be removed, we want to discreetly manipulate the model instead of retraining it from scratch. For example, it may turn out that an item from the data is erroneous or there may be privacy or copyright related problems with an item. Additionally, the model may produce wrong results because some item may be associated incorrectly. In all these cases, the item needs to be removed from the data and the model. However, removing an item from a model is a difficult task because the model only encodes the data and does not contain the data in such a way that individual elements can be distinguished and removed. Therefore, models are often retrained from scratch after items are removed from the data.

3nd Workshop on Humanities-Centred Artificial Intelligence (CHAI)

*Corresponding author.

bender@ifis.uni-luebeck.de (M. Bender); kira.schwandt@student.uni-luebeck.de (K. Schwandt); moeller@ifis.uni-luebeck.de (R. Möller); gehrke@ifis.uni-luebeck.de (M. Gehrke)

https://www.ifis.uni-luebeck.de/~bender/ (M. Bender); https://www.ifis.uni-luebeck.de/~moeller/ (R. Möller); https://www.ifis.uni-luebeck.de/~gehrke/ (M. Gehrke)

0000-0002-1854-225X (M. Bender); 0000-0002-1174-3323 (R. Möller); 0000-0001-9056-7673 (M. Gehrke)

For different models, there are different approaches to avoid retraining from scratch, e.g., for k-Means [1, 2] or linear and logistic regression [3]. The common idea is to avoid retraining the model and instead updating the model by applying an inverse operation that removes an item of the data from the model.

In this paper, we consider the scenario of a information retrieval agent [4], which is an is a rational and autonomous unit acting in a world fulfilling a defined task. In our case the agent works with a corpus of text documents and provides an information retrieval service for human users. Internally, the agent uses Subjective Content Descriptions (SCD), whereas each SCD represents a concept in the corpus and is associated to similar sentences across the corpus. When a human sends a query to the agent, the agent determines similar SCDs representing similar concepts and returns the sentence associated with these SCDs to the human. The SCDs of a corpus are modeled by an SCD matrix which can be trained by the Supervised Estimator of SCD Matrices (SEM) [5] or the UnSupervised Estimator of SCD Matrices (USEM) [6].

As the name *subjective* content description already states, it is *subjective* which sentences are associated with which description. From the perspective of a human users of the agent, there may be a faulty association between a sentence and an SCD. If the agent uses such a faulty association while computing the response for a user, the response will be determined erroneous by the user. In this situation, the human user may give feedback to the agent and the agent needs to update its SCD matrix by removing the faulty association between sentence and SCD. We are now back to the point, where the model of the agent needs to be updated. Additionally, the agent should not need to retrain the SCD matrix from scratch after the sentence has been removed from the corpus.

To solve the problem, this paper provides FrESH, an approach for Feedback-reliant Enhancement of Subjective Content Descriptions. The agent may use FrESH to process feedback by removing a faulty sentence from the SCD matrix and only needs to update the SCD the faulty sentence has been associated with. Furthermore, FrESH can be used to remove sentences from the SCD matrix if their content should be removed entirely, i.e., to keep *Fake-News* out of the corpus or to remove copyright or privacy protected content.

An agent as described above may be integrated in an information system working with SCDs and providing a human friendly way of interacting with SCD based models [7]. Adding a feedback mechanism to the agent allows the users of the information system to build their own subjective model by enhancing an initial SCD matrix learned by USEM on their corpus.

In the humanities, corpora of text documents are often rather small and need to be annotated by scientists. An information system using SCDs helps theses scientists doing their work, e.g., SCDs are used to identify comments and actual content in Tamil and Greek language corpora [8]. Furthermore, SCDs provide techniques for (i) estimating SCDs for a single previously unseen text document using the Most Probably Suited SCD (MPS^{2}CD) algorithm [9], (ii) classifying a text document as related, extended, revised, or unrelated to a corpus [9], (iii) moving the SCDs from one corpus to another similar corpus by adapting the SCDs' domain [10], (iv) enriching SCDs in a corpus already sparsely associated with SCDs [11], or (v) detecting complementary documents to a corpus [12].

The remainder of this paper is structured as follows: First, we recap the basics of SCDs, the SCD matrix and the estimation of SCD matrices. Second, we formalize the problem of incorporating feedback by removing false associations of sentences with SCDs in an SCD matrix.

Afterwards, we provide three consecutive methods to solve the problem and evaluate each. Finally, we conclude with a summary and short outlook.

2. Preliminaries

This section specifies notations, recaps the basics of SCDs and describes the estimation of SCD matrices.

2.1. Notations

First, we formalize our setting of a corpus.

- A word w_i is a basic unit of discrete data from a vocabulary $\mathcal{V} = \{w_1, \ldots, w_L\}$, $L \in \mathbb{N}$.
- A sentence s is defined as a sequence of words $s = (w_1, \ldots, w_N)$, $N \in \mathbb{N}$, where each word $w_i \in s$ is an element of vocabulary $\mathcal{V}$. Commonly, a sentence is terminated by punctuation symbols like ".", "!", or "?".
- A document d is defined as a sequence of sentences $d = (s_1^d, ..., s_M^d)$, $M \in \mathbb{N}$.
- A corpus $\mathcal{D}$ represents a set of documents $\{d_1, \ldots, d_D\}$, $D \in \mathbb{N}$.
- An SCD t is a tuple of the SCD's additional data $\mathcal{C}$ and the referenced sentences $\{s_1, ..., s_S\}$, $S \in \mathbb{N}$. Thus, each SCD references sentences in documents of $\mathcal{D}$, while in the opposite direction a sentence is associated with an SCD.
- A sentence associated with an SCD is called SCD window, inspired by a tumbling window moving over the words of a document. Generally, an SCD window might not be equal to a sentence and may be a subsequence of a sentence or the concatenated subsequences of two sentences, too. Even though, in this paper, an SCD window always equals a sentence.
- For a corpus $\mathcal{D}$ there exists a set g called SCD set containing K associated SCDs $g(\mathcal{D}) = \left\{t_j = \left(\mathcal{C}_j, \bigcup_{d \in \mathcal{D}} \{s_1^d, ..., s_S^d\}\right)\right\}_{j=1}^{K}$. Given a document $d \in \mathcal{D}$, the term $g(d)$ refers to the set of SCDs associated with sentences from document d.
- Each word $w_i \in s^d$ is associated with an influence value $I(w_i, s^d)$ representing the relevance of w_i in the sentence s^d. For example, the closer w_i is positioned to the object of the sentence s^d, the higher its corresponding influence value $I(w_i, s^d)$. The influence value is chosen according to the task and might be distributed binomial, linear, or constant.

2.2. Subjective Content Descriptions

SCDs provide additional location-specific data for documents [5]. The data provided by SCDs may be of various types, like additional definitions or links to knowledge graphs.

Kuhr et al. use an SCD-word distribution represented by a matrix when working with SCDs [5]. The SCD-word distribution matrix, in short SCD matrix, can be interpreted as a generative model. A generative model for SCDs is characterized by the assumption that the SCDs generate the words of the documents. We assume that each SCD shows a specific distribution of words of the referenced sentences in the documents.

The SCD matrix $\delta(\mathcal{D})$ models the distributions of words for all SCDs $g(\mathcal{D})$ of a corpus $\mathcal{D}$ and

Algorithm 1 Supervised Estimator of SCD Matrices $\delta(\mathcal{D})$

function SEM($\mathcal{D}$, $g(\mathcal{D})$)
  Input: Corpus $\mathcal{D}$; Set of SCDs $g(\mathcal{D})$
  Output: SCD-word distribution matrix $\delta(\mathcal{D})$
  Initialize an $K \times L$ matrix $\delta(\mathcal{D})$ with zeros
  for each document $d \in \mathcal{D}$ **do**
    for each SCD $t = (\mathcal{C}, \{s_1^d, ..., s_S^d\}) \in g(d)$ **do**
      for $j = 1, ..., S$ **do** ▷ Iterate over sentences
        for each word $w_i \in s_j^d$ **do**
          $\delta(\mathcal{D})[t][w_i] \mathrel{+}= I(w_i, s_j^d)$
  return $\delta(\mathcal{D})$

is structured as follows:

$$\delta(\mathcal{D}) = \begin{array}{c c} & \begin{array}{ccccc} w_1 & w_2 & w_3 & \cdots & w_L \end{array} \\ \begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_K \end{array} & \begin{pmatrix} v_{1,1} & v_{1,2} & v_{1,3} & \cdots & v_{1,L} \\ v_{2,1} & v_{2,2} & v_{2,3} & \cdots & v_{2,L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{K,1} & v_{K,2} & v_{K,3} & \cdots & v_{K,L} \end{pmatrix} \end{array}$$

The SCD matrix consists of K rows, one for each SCD in $g(\mathcal{D})$. Each row contains the word probability distribution for an SCD. Therefore, the SCD matrix has L columns, one for each word in the vocabulary of the corpus $\mathcal{D}$.

2.3. Supervised and UnSupervised Estimator for SCD Matrices

The SCD matrix can be estimated in a supervised manner given the set $g(\mathcal{D})$ for a corpus $\mathcal{D}$. SEM is described in Algorithm 1. Given a corpus $\mathcal{D}$, the algorithm iterates over each document d in the corpus and the document's SCDs. For each associated SCD t, the referenced sentences $s_1^d, ..., s_S^d$ are used to update the SCD matrix. Thereby, the row of the matrix representing SCD t gets incremented for each word in each sentence by each word's influence value.

Finally, the SCD matrix needs to be normalized row-wise to meet the requirements of a probability distribution. However, the normalization is often skipped because later the cosine similarity is often used with the rows of the matrix and the cosine similarity does a normalization by definition.

Unlike SEM, USEM estimates an SCD matrix $\delta(\mathcal{D})$ without needing the SCD set $g(\mathcal{D})$. USEM initially starts by associating each sentence to one unique SCD, which leads to an initial SCD matrix consisting of a row for each sentence in the document's corpus. Then USEM finds the sentences in the corpus that represent the same concept and groups them into one SCD.

However, the SCDs estimated by USEM and SEM may contain faulty sentences which have to be removed if a human user gives feedback about such errors.

		SCD matrix	
		Frequecies	Distribution
Errors/ Feedback	Faulty sentences and their SCDs	Method 1	Method 3
	Only faulty sentences	Method 2	Method 4

Figure 1: The four different cases regarding the input data of FrESH.

Algorithm 2 FrESH Method 1 (M1)

1: **function** FRESH-M1(SCD Matrix $\delta(\mathcal{D})$, Set of faulty Sentences with SCDs p)
2: **for** each $(s, t) \in p'$ **do** ▷ Iterate over faulty sentences with SCDs
3: **for** each $w_i \in s$ **do** ▷ Iterate over words
4: $\delta(\mathcal{D})[t][w_i] \mathrel{-}= I(w_i, s)$
5: **return** $\delta(\mathcal{D})$

3. Incorporate Feedback

In this section, we present FrESH and thereby how to incorporate human feedback to enhance an SCD matrix. For our method, we assume that the feedback exactly states which sentences is falsely associated with its SCD, i.e., the human user may click on a button *remove sentence*. Therefore, we consider the problem of removing a faulty sentence from an SCD matrix. This also removes the sentences from the corpus, making FrESH useful for removing *Fake-News*, copyright, and privacy protected content, too.

We differentiate between two types of SCD matrices $\delta(\mathcal{D})$: The values contain the frequencies of the words without normalization or a normalized version containing row-wise distributions of words. Additionally, we differentiate between the way the errors are contained in the feedback: Only a faulty sentences is given as input or a faulty sentence together with it's SCD is given.

In total, we have four different cases regarding the input data of FrESH, which are also shown in Figure 1. Next, we will consider each case and develop method 1 to 4 to solve each.

3.1. Method 1

The input consists of an SCD matrix containing the frequencies of the words and a set of faulty sentences with their associated SCDs p. It is then possible to revert the operations of SEM (Algorithm 1). SEM adds in Line 9 for each word in a sentence the frequency of the word weighted by an influence value to the row of the matrix representing the SCD. The first method (M1) reverses this addition by subtracting the same value in Line 4 of Algorithm 2.

M1 inverts the operations of SEM. Therefore, an SCD matrix learned by SEM on the corpus without p will be identical to an SCD matrix from which p was removed by M1.

3.2. Method 2

The input consists of an SCD matrix containing the frequencies of the words and a set of faulty sentences p'. Hence, first the SCD for each faulty sentence needs to be found, afterwards the

Algorithm 3 FrESH Method 2 (M2)

function FRESH-M2(SCD Matrix $\delta(\mathcal{D})$, Set of faulty Sentences p')
  for each $s \in p'$ **do** ▷ Each faulty sentence
    $t = \text{MPS}^2\text{CD}(\delta(\mathcal{D}), s)$ ▷ Get SCD of sentence, use MPS²CD [9]
    for each word $w_i \in s$ **do** ▷ Iterate over words
      $\delta(\mathcal{D})[t][w_i] \mathrel{-}= I(w_i, s)$
  return $\delta(\mathcal{D})$

Algorithm 4 FrESH Method 3 (M3)

function FRESH-M3(SCD Matrix $\delta'(\mathcal{D})$, Set of faulty Sentences with SCDs p')
  for each $(s, t) \in p'$ **do** ▷ Iterate over faulty sentences with SCDs
    $m = \min_{j=1,\ldots,L;\ \delta'(\mathcal{D})[t][j]>0} \delta'(\mathcal{D})[t][j]$ ▷ Minimal value in SCD's row
    for each word $w_i \in s$ **do** ▷ Iterate over words
      $\delta'(\mathcal{D})[t][w_i] \mathrel{-}= I(w_i, s) \cdot m$
  return NORMALIZEROWS($\delta'(\mathcal{D})$)

sentence can be removed from the matrix as in M1. The MPS²CD algorithm finds a most suitable SCD for a sentence, and is therefore used with this second method (M2) in Algorithm 3.

Unlike M1, M2 does not guarantee to produce identical matrices if a matrix is first trained on a corpus with faulty sentences that are then removed, compared to a matrix trained directly on a corpus without faulty sentences. This is caused by the fact, that MPS²CD is used to find a most suitable SCD, which may not be the SCD used initially. However, MPS²CD works quite accurately, so we expect only a small difference, which should not affect an agent using SCDs.

3.3. Method 3

The input consists of an SCD matrix containing row-wise distributions of words $\delta(\mathcal{D})'$ and a set of faulty sentences with their associated SCDs p. The difficulty is to revert the operations of SEM on the distributions of each SCD. During normalization of the SCD matrix, all frequencies in each matrix's row are divided by a divisor. To approximate this divisor with M3, we assume that in each row at least one value had the frequency one, i.e., there was a word that occurred only once in the referenced sentences of the SCD. This word will have the minimal value in the distribution and we assume it had frequency one. So the normalized value is equal to one divided by the divisor, which in turn is a factor that we can use to decrease the values that are subtracted from the matrix by the same amount as during the normalization.

Again, M3 is based on an approximation and may not produce identical matrices.

3.4. Method 4

The input consists of an SCD matrix containing row-wise distributions of words and a set of faulty sentences p'. For this input, the ideas from M2 and M3 can be combined. First, the SCD for each faulty sentences needs to be determined, which can be done via MPS²CD. Second, the minimal value of this SCD's distribution can be determined and used as factor while changing the SCD matrix.

We will not consider M4 in detail, as it is only a combination of M2 and M3. In most use-cases there is no benefit in normalizing the SCD matrix, because mostly the cosine similarity is used and it does a normalization by definition. Therefore, especially M2 is beneficial for our agent using SCDs and getting feedback from its users, e.g., to remove faulty sentences.

Next, we present an evaluation of M1, M2, and M3.

4. Evaluation

After we have introduced FrESH to remove sentences from an SCD matrix, we present an evaluation of the methods M1, M2, and M3. First, we describe the used corpus and evaluation workflow. Afterwards, we present the results of the evaluation.

4.1. Dataset

In this evaluation we use the 20 newsgroups[1] dataset. 20 newsgroups is a well-known corpus consisting of e-mails from 20 e-mail newsgroups. Thematically, the 20 newsgroups can be divided into six topics, *computer*, *sport*, *science*, *politics*, *religion* and *for sale*. The entire corpus consists of 18 828 text documents. The documents have between 1 and 39 682 words with a median of 160 words.

4.2. Workflow and Metrics

FrESH is implemented in Python and runs in a Docker container on a machine featuring 8 Intel 6248 cores at 2.50GHz (up to 3.90GHz) and 16GB RAM.

To evaluate each method, we first create the corpus $\mathcal{D}_f$ containing all documents from 20 newsgroups. We annotate the sentences with OpenIE [13] and use the extracted triples as data for the SCDs. Then, we split $\mathcal{D}_f$ into to subsets, $\mathcal{D}_s$ contains the sentences we assume to be faulty and $\mathcal{D}_k$ contains the non-faulty sentences. For each corpus we learn an SCD matrix using SEM, i.e., $\delta(\mathcal{D}_f)$ and $\delta(\mathcal{D}_k)$. Afterwards, we apply one of the three methods to $\delta(\mathcal{D}_f)$ and remove the sentences in $\mathcal{D}_s$. This step yields $\delta'(\mathcal{D}_f)$, which should be identical to $\delta(\mathcal{D}_k)$.

As a metric, we then calculate the difference of the matrices $\delta'(\mathcal{D}_f)$ and $\delta(\mathcal{D}_k)$ for each SCD t using the Hellinger distance [14]:

$$HD_t(\delta'(\mathcal{D}_f), \delta(\mathcal{D}_k)) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{L}\left(\sqrt{\delta'(\mathcal{D}_f)[t][i]} - \sqrt{\delta(\mathcal{D}_k)[t][i]}\right)^2}$$

4.3. Results

In this section, we present the results of the evaluation. In the left part of Figure 2, the duration of removing one sentence with the three different methods is shown. M1 is very fast, while M2

[1] http://qwone.com/~jason/20Newsgroups/

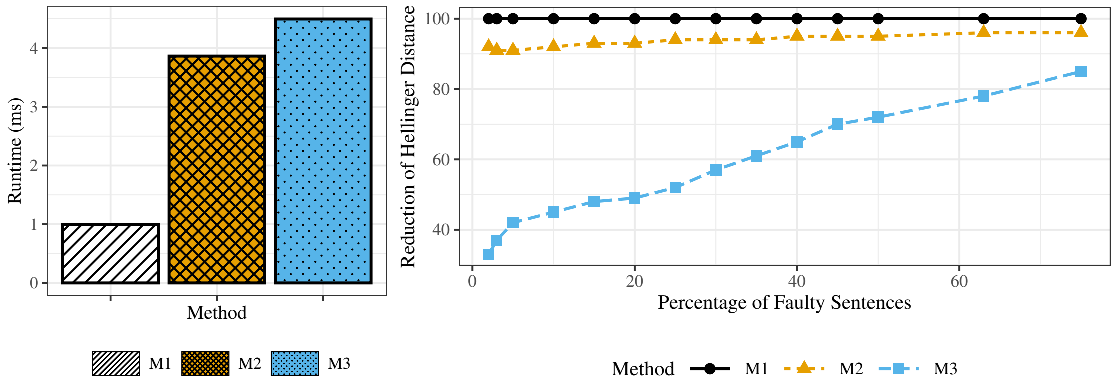

Figure 2: Left: Average runtime removing one sentence with the different methods. Right: Reduction of difference between the matrices $\delta'(\mathcal{D}_f)$ and $\delta(\mathcal{D}_k)$ for the three methods and different numbers of faulty sentences.

and M3 need more time. M2 needs to determine the SCD via MPS^{2}CD which takes time and M3 needs to calculate the factor needed to maintain the distribution in the matrix. However, all methods are reasonably fast, as it takes at most 4.5 ms to remove a sentence which will be clearly faster than retraining the matrix from scratch.

In the right part of Figure 2, the difference between the matrices $\delta'(\mathcal{D}_f)$ and $\delta(\mathcal{D}_k)$ is shown. The difference is shown as percentage of the reduction of the Hellinger distance averaged for all SCDs: First, the Hellinger distance between $\delta(\mathcal{D}_f)$ (full corpus) and $\delta(\mathcal{D}_k)$ (non-faulty part) is calculated. When removing the faulty sentences from $\delta(\mathcal{D}_f)$ to get $\delta'(\mathcal{D}_f)$ the distance should become smaller, which is shown as reduction. A reduction of 100 % implies that $\delta(\mathcal{D}_k)$ equals $\delta'(\mathcal{D}_f)$. The reductions are shown for all three methods and different numbers of faulty sentences in $\mathcal{D}_s$. The x-axis represents the size of $\mathcal{D}_s$ as percentage of the entire corpus $\mathcal{D}_f$.

M1 inverts the operations of SEM, therefore, after removing all faulty sentences, both SCD matrices are equal and the difference is reduced completely. M2 reaches very high reductions around 95% and thus provides a reliable technique to remove faulty sentences. The results of M3 are significant below M1 and M2. Especially, if only a few faulty sentences are deleted, the difference keeps high. This implies that our assumption used in M3, to approximate the factor used to normalize the matrix, does not hold.

Therefore an agent maintaining an SCD matrix should store it using frequencies instead of distributions. Doing so, the agent may use M1 and M2 to incorporate feedback.

5. Conclusion

This paper presents FrESH, a technique to enhance SCD matrices using human feedback. Humans indicate faulty sentences, which then can be removed from the matrix without retraining the matrix. This allows human users of an agent working with SCDs to build their own subjective model for the corpus the agent works with. Additionally, it allows the operators of the agent to remove erroneous, copyright, or privacy protected data quickly and efficiently. Our evaluation shows, that FrESH works quite reliable when the SCD matrix contains the word frequencies.

Currently, a faulty sentence is removed entirely from the SCD matrix and corpus. Furthermore, dependencies and relations to other SCDs and other sentences are not maintained, if a sentences is removed. Future work will focus on maintaining these dependencies.

References

[1] S. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (1982) 129–137.

[2] A. A. Ginart, M. Y. Guan, G. Valiant, J. Zou, Making AI forget you: Data deletion in machine learning, 2019.

[3] Z. Izzo, M. A. Smart, K. Chaudhuri, J. Y. Zou, Approximate data deletion from machine learning models, in: International Conference on Artificial Intelligence and Statistics, 2021.

[4] S. Russell, P. Norvig, Artificial Intelligence, Global Edition A Modern Approach, Pearson Deutschland, 2021.

[5] F. Kuhr, T. Braun, M. Bender, R. Möller, To Extend or not to Extend? Context-specific Corpus Enrichment, Proceedings of AI: Advances in Artificial Intelligence (2019) 357–368.

[6] M. Bender, T. Braun, R. Möller, M. Gehrke, Unsupervised estimation of subjective content descriptions, Proceedings of the 17th IEEE International Conference on Semantic Computing (ICSC-23) (2023).

[7] M. Bender, T. Braun, R. Möller, M. Gehrke, Unsupervised estimation of subjective content descriptions in an information system, Int. Journal of Semantic Computing (2023).

[8] M. Bender, T. Braun, M. Gehrke, F. Kuhr, R. Möller, S. Schiff, Identifying and translating subjective content descriptions among texts, Int. J. Semantic Computing 15 (2021) 461–485.

[9] F. Kuhr, M. Bender, T. Braun, R. Möller, Augmenting and automating corpus enrichment, Int. J. Semantic Computing 14 (2020) 173–197.

[10] F. Kuhr, M. Bender, T. Braun, R. Möller, Context-specific adaptation of subjective content descriptions, 15th IEEE International Conference on Semantic Computing, (ICSC 2021), Laguna Hills, CA, USA, January 27-29 (2021) 134–139.

[11] F. Kuhr, B. Witten, R. Möller, Corpus-Driven Annotation Enrichment, 13th IEEE International Conference on Semantic Computing, (ICSC 2019), Newport Beach, CA, USA, January 30 - February 1, (2019) 138–141.

[12] M. Bender, F. Kuhr, T. Braun, To extend or not to extend? complementary documents, 16th IEEE International Conference on Semantic Computing, (ICSC 2022), Virtual, January 26-28 (2022) 17–24.

[13] G. Angeli, M. J. Johnson Premkumar, C. D. Manning, Leveraging linguistic structure for open domain information extraction, Proceedings of the Association of Computational Linguistics (ACL) (2015) 344–354.

[14] E. Hellinger, Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen, Journal für die reine und angewandte Mathematik (1909) 210–271.