${}_t\gamma$ – Inter-annotator agreement for categorization with simultaneous segmentation and transcription-error correction

Fabian Barteld **Ingrid Schröder** **Heike Zinsmeister**
Institut für Germanistik
Universität Hamburg
firstname.lastname@uni-hamburg.de

Abstract

When annotating non-standard texts such as historical texts or spoken language, tasks that are normally considered to be pure categorization tasks such as part-of-speech tagging are often combined with correcting errors in the tokenization and even the transcribed text itself along the way. As a consequence, inter-annotator agreement measures are needed that measure agreement for categorization by also taking changes in segmentation and the underlying text into account. In this paper, we present the first inter-annotator measure of this kind, text-gamma (${}_t\gamma$). Based on γ (Mathet et al., 2015), the inter-annotator agreement is measured using an alignment of the annotations. For this, we consider alignments of the annotations that follow from optimal alignments of the underlying text sequences. Furthermore, we use a specialized function to measure the disorder of the alignment. For chance-correction, we introduce a method that takes the annotation bias introduced by pre-annotation into account when estimating the expected (dis)agreement between annotators.

1 Introduction

The annotation of non-standard texts such as historical texts, spoken language, or user-generated content poses specific problems for the annotation process. Even tasks as basic as segmenting a text into tokens for subsequent part-of-speech (POS) tagging become considerably harder for such data than for standard text since whitespace often does not coincide with the boundaries of syntactical words (Barteld et al., 2014). As a consequence, human annotators are sometimes asked to check and correct the underlying tokenization along the way when annotating this kind of data. Examples for annotation guidelines that address this task explicitly are Čibej et al. (2016), giving guidelines for the normalization of Slovene Tweets and the guidelines for HiTS (Dipper et al., 2013), a POS tagset developed for historical variants of German. Furthermore, when working with data that is not born-digital such as historical texts or data that is not written in nature such as spoken language, the textual representation of the data that is annotated is already an interpretation of the original data and might contain errors. This adds the necessity of correcting the text during the process of tagging. As an example two different transcriptions of the same text are shown in (1) where an "i" followed by an "n" was corrected to "m".

(1) sambt aller v**in̄**stendicheit vthgelacht /
sambt aller v**m̄**stendicheit vthgelacht /
with all circumstances construed
'construed extensively'
(Source: Verl. Sohn)

Annotation tools developed for the annotation of non-standard text such as CoBaLT (Kenter et al., 2012) and CorA (Bollmann et al., 2014) consequently allow the annotators to change the underlying text and the segmentation into tokens during the annotation process. Effectively, this is turning the annotation from a categorization task into a combination of string editing, segmentation, and categorization.

While the annotation tools exist, there is no chance-corrected inter-annotator agreement measure for this setting available. We address this issue by presenting *text-gamma* (${}_t\gamma$) the first measure for categorization that takes into account the possibility of correcting the segmentation and the text along the way. As the quality of the transcription and the segmentation presented to the annotators affects the expected number of corrections, we also introduce a method for determining

chance correction that takes the annotation bias introduced by pre-annotations into account when estimating the expected (dis)agreement between annotators.

While ${}_t\gamma$ is usable for all kinds of segments and categories – even multiple categorizations of a segment, e.g. assigning POS tags and lemmas to tokens – with simultaneous correction of the segmentation and the underlying text done by an arbitrary number of annotators, we exemplify and evaluate this measure on data as created in a setting of tokenization and POS tagging of an historical text by two annotators.

2 The annotation task

In this section, we present a formalization of the different types of categorization tasks: (a) *pure categorization*, the traditional task, where predefined segments are labeled with a category, (b) *categorization with segmentation correction*, the extension of pure categorization to born-digital, non-standard texts such as computer-mediated communication, where the segmentation is corrected by the annotators, and (c) *categorization with segmentation and text correction*, the extension of categorization to non-standard texts that are not born-digital such as historical texts where the digitized text might contain errors that are corrected by the annotators as well as the segmentation.

For the formalization, we combine the quite similar concepts that are introduced by Mathet et al. (2015) and used in GATE (Cunningham et al., 2014).[1] We define an **annotation** as an entity that has been created by a (human or automatic) annotator, that has a type (e.g. *token*, *sentence*) and a feature set realized as a set of attribute-value pairs (e.g. *POS=noun*). An annotation has a position on a continuum in terms of start and end offsets. The **continuum** can be continuous, e.g. in the case of a timeline where the offsets represent the points in time where an annotation starts and ends. We look at cases where the continuum is a text represented by a character string and the start and end points of annotations are given by character offsets, therefore the continuum is discrete. Furthermore, annotations that are attached to the same continuum can be combined in an **annotation set**. When the continuum is text, i.e., a character string, we mark this with the subscript *text* (**annotation set$_{\text{text}}$**).

Using this terminology, the traditional task of POS tagging – an example of pure categorization – can be modeled as an iterative creation of annotation sets on the same continuum. The first iteration, which is usually done automatically by a tokenizer, creates annotations of the type *token* with non-overlapping start and end offsets. The annotations cover the continuum completely, only whitespace characters may be left uncovered.[2] The resulting annotation set$_{\text{text}}$ is the input to the second iteration of the annotation procedure – this phase is traditionally seen as the annotation proper: In this second iteration, annotators are presented with the annotated text resulting from the first iteration and add new feature-value pairs (for POS) to the annotations of type *token*. For inter-annotator agreement experiments, iteration 2 is done independently by multiple annotators, resulting in multiple annotation sets$_{\text{text}}$. Fig. 1 illustrates the three types of categorization tasks introduced above.

Fig. 1a shows the traditional setting, pure categorization. In this setting, the annotators do not change the text or the token segmentation, i.e., in our terminology, the continuum and the offsets of the annotations, respectively. In this case, each resulting annotation set$_{\text{text}}$ contains the same number of annotations and for each annotation there is exactly one corresponding annotation in the other sets, which are easily identified by the offsets. The only possible difference is in the POS values. This setting allows for a straightforward comparison of the assigned categories.

Fig. 1b shows categorization with simultaneous segmentation correction, i.e., the annotators occasionally change the start and end offsets of annotations by merging or splitting them. This results in annotation sets$_{\text{text}}$ derived from the same input, which possibly differ in the number of annotations which again might also differ in their positions on the continuum. Therefore, it is not as straightforward to identify corresponding annotations for which the assigned categories have to be

[1] Both introduce similar concepts, treating annotations as spans over a continuum. However, there are differences. For example, the annotations as used in GATE are more general than the units introduced by Mathet et al. (2015), as annotations are typed and allow for more than one category by using feature sets.

[2] Note that in our formalization tokens are independent of whitespace in the underlying texts. E.g. the string 'New York' can be treated as a multi-word unit by creating an annotation that covers the whole sequence or as two tokens by creating two annotations that cover the first and the second part respectively, leaving the whitespace uncovered. Therefore, changing the segmentation does not affect the underlying continuum.

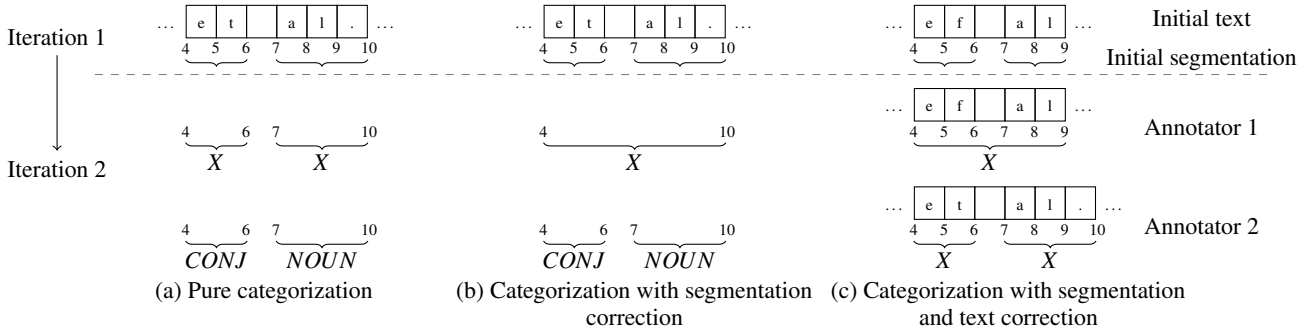

Figure 1: Different types of categorization tasks, using the universal POS tagset (Petrov et al., 2012)

compared. Still, the annotations are all attached to the same continuum.

Fig. 1c shows the case when the data is not born-digital and annotators are allowed to change the textual representation, i.e., the underlying continuum, as well as the segmentation. Textual changes can also affect the annotations, e.g., when inserting a character into the text, the offsets of all subsequent annotations need to be adapted. This is exemplified by the "." that the second annotator inserted. Therefore the last offset in the example is 10, while it is 9 for the first annotator. In the end, the resulting annotation sets might differ regarding the contained annotations. Furthermore, the annotations are attached to different continua.

This third annotation task could be split into three separate annotation processes where first, the text is corrected, then this text is segmented and in a third step the segments are labeled. Such a pipelined annotation setting would allow us to compute the inter-annotator agreement for each of the three steps independently using existing measures. However, it would introduce the need to fix the result of each step, e.g., errors in the segmentation and the transcription cannot be corrected when assigning labels. Our experience with the creation of a corpus with Middle Low German texts shows that many segmentation and/or transcription errors only become apparent while assigning POS tags. Consequently, we present the first measure for inter-annotator agreement that can be used when categorization is combined with segmentation and transcription-error correction.

3 Related work

There exist inter-annotator agreement (IAA) measures for each of the tasks described in Fig. 1 when performed individually. In wide use are measures like α (Krippendorff, 1980) and κ (Cohen, 1960) for categorization tasks. Artstein and Poesio (2008) give an overview of these and other measures for categorization tasks.

A commonly-used measure for the quality of a segmentation is *WindowDiff* (Pevzner and Hearst, 2002). However, this and related measures, are geared toward comparing an automatically created segmentation with a reference segmentation and therefore do not apply chance correction. For manually created segmentation, it is preferable to use measures that take chance correction into account like α_U (Krippendorff, 1995) that measures the degree to which segments overlap or B-based π^* (Fournier, 2013) that is designed for complete segmentation tasks where the annotations cover the whole continuum.

For a setting in which the two tasks of detecting units and categorizing them are combined, there exist only a few measures, among them different versions of ${}_u\alpha$ (Krippendorff, 2013; Krippendorff, 2015) and γ (Mathet et al., 2015). The latter is based on finding an optimal alignment between the annotations from a set of annotators, i.e., identifying the annotations that are most similar, aligning them, and then calculating the mean dissimilarity between them. For the task considered here such a measure has to be combined with a measure quantifying the dissimilarity between texts. There are a few attempts to measure the quality of transcriptions, e.g., Munyaradzi and Suleman (2013) using a normalized variant of the Levenshtein distance for manuscript transcriptions and Valenta et al. (2014) using word accuracy for speech transcription. Both do not apply chance correction. As using chance correction for IAA is considered state of the art (Artstein and Poesio, 2008), we aim to apply chance correction in our measurements. For our task the chance correction has to account

for the fact that, at least with transcription and segmentation, the annotators do not start from scratch but are presented with pre-annotations, i.e., they start with a tokenized transcription. When the impact of these pre-annotations on IAA (Fort and Sagot, 2010) is not considered, the actual agreement would be overestimated.

In the next section, we present a method to create alignments between the annotations from different annotators. Using these alignments the disagreement between aligned annotations can be calculated similarly to the way in which it is calculated in γ. However, differences in the underlying texts have to be included in the disagreement. Subsequently, we propose a method to estimate the expected agreement taking the pre-annotations into account. Finally, we evaluate our measure using corpus shuffling (Mathet et al., 2012).

4 Aligning annotations from different continua using sequence alignment

Gamma (γ) (Mathet et al., 2015) is calculated using the mean dissimilarity between aligned annotations taking the category and the position into account. The alignment used is the alignment with the lowest mean dissimilarity. For this, all possible alignments are considered in the original computation. Using this method directly is not possible in our situation, as differences in the position of units may result from different textual bases. For instance, the insertion of one letter by only one of the annotators shifts all following offsets of her annotations to the right. As a consequence, annotations that span only one letter would not overlap when comparing the texts of different annotators, leading to artificially high dissimilarities.

In example (1), the same part of the original texts is transcribed with two letters ("iñ") and with one letter ("ñ") in the two transcriptions. This influences the characeter offsets of all the following characters, e.g., the "/" starts at the position 38 in the first transcription and at position 37 in the second transcription. As it only has a length of one, there is no overlap between these two tokens when only considering their positions in the corresponding transcription.

Ignoring the position of annotations is not a solution here, since it would allow the alignment of annotations spanning the same sequence of characters even if they were from different ends of the text. Therefore, we apply a different method to find optimal alignments between annotations by using (multiple) sequence alignment (MSA). MSA is a common technique in analyzing genome sequences and an active research topic in bioinformatics (Chatzou et al., 2015). MSA has been used in natural language processing as well (Barzilay and Lee, 2002; Prokić et al., 2009; List, 2012; Kirschenbaum, 2013). Given n input sequences the result of a MSA is a set of n aligned sequences, i.e., the resulting sequences all have the same length and the characters at a given position in the sequences are aligned with each other. To accomodate differing lengths between the input sequences, gaps (represented by "_" in the examples) are inserted (cf. example 2).

(2) v o _ r w a h r
v o u r w a _ r
f _ u r w a h r

An optimal sequence alignment is one that minimizes the costs introduced by matches, mismatches and gaps. The basic algorithm to find an optimal alignment is a specialization of the algorithm described by Needleman and Wunsch (1970). Normally, the alignment of mismatches is allowed. However, then it is not always possible to perfectly align the annotations on the new sequences as can be seen from the following example:

(3) h a t s
h a t t

In (3), it is not possible to positionally align an annotation corresponding to *hat* in the first sequence with an annotation corresponding to *hatt* in the second sequence on the continuum created by sequence alignment. As such this is not a problem for aligning these annotations for the calculation of γ. However, when textual and positional dissimilarity are both integrated into the calculation of the alignments' dissimilarity, the dissimilarity between *hatt* and *hat* will be artificially high as the annotations differ both positionally and textually.

To not over-punish such settings, we do not include the position in the dissimilarity measure of ${}_t\gamma$. Furthermore, we do not consider all possible alignments of annotations but only alignments of annotations that have the same position in an optimal sequence alignment. This avoids the problem of aligning two annotations from different regions of the continuum as described above.

So far, we would not allow the alignment of *hat* and *hatt* in (3). To make this alignment pos-

sible, we only allow matches and gaps in the sequence alignment, e.g., by setting the cost for mismatches such that introducing gaps will always be preferred.

To create possible alignments of annotations, we introduce boundaries as elements into the sequences (denoted by "{" and "}" below). Now, aligned annotations can be read off directly from the aligned sequences.[3] The strings from example (3) lead to the following optimal sequence alignments in (4) and (5):

(4) { h a t } { s _ }
{ h a t _ _ _ t }

(5) { h a t _ } { s }
{ h a t t } _ _ _

Both alignments ore optimal sequence alignments even if in (4) no annotations are aligned and in (5) the annotations covering *hat* and *hatt* are aligned. In our approach, all alignments of annotations that result from an optimal sequence alignment are considered for finding the best alignment of annotations.

We want to point out the behavior of this alignment method for adjacent annotations that only partially overlap comparing two annotation sets. Take the artificial example of $\{a\}\{bbc\}$ and $\{abb\}\{c\}$. Examples (6) and (7) show two optimal alignments of these sequences:

(6) { a } { b b _ _ c }
{ a _ _ b b } { c }

(7) { a _ _ } { b b c }
{ a b b } { _ _ c }

In this case, aligning the annotations or not aligning them both result in optimal sequence alignments (both with a cost of $4 \times c_g$, where c_g denotes the cost of inserting a gap). However, in the examples (8) and (9) with the sequences $\{a\}\{bc\}$ and $\{ab\}\{c\}$, variant (9), in which the annotations are aligned is "cheaper" and hence is the only optimal sequence alignment:

(8) { a } { b _ _ c }
{ a _ _ b } { c }

(9) { a _ } { b c }
{ a b } { _ c }

In the examples (10) and (11) with sequences $\{a\}\{bbbc\}$ and $\{abbb\}\{c\}$, it is the other way round. Here the annotations are not aligned as only option (10) is an optimal sequence alignment.

(10) { a } { b b b _ _ c }
{ a _ _ b b b } { c }

(11) { a _ _ _ } { b b b c }
{ a b b b } { _ _ _ c }

For these examples, we assumed that gaps at textual positions (gap_t) have the same cost as gaps at boundary positions (gap_b). If we allow the setting of gap_b independently of gap_t a preference for or against aligning annotations that partially overlap can be chosen. Supposing that gap_t is set to 1, the following cases apply: (i) when two boundaries are less than $2 \times gap_b$ characters apart, they are always aligned, (ii) when two boundaries are exactly $2 \times gap_b$ characters apart, they can be aligned and, (iii) when two boundaries are more than $2 \times gap_b$ characters apart are never aligned. In our experiments, we set $gap_t = gap_b$.

There exist many algorithms for MSA differing in the computational complexity and the accuracy of the produced alignments. In principle all of these methods are usable to induce possible alignments of annotations. For the evaluation, where we aligned two versions of one text consisting of about 3,700 characters, we used the algorithm by Needleman and Wunsch (1970) but followed more than one path in the backtracking phase in order to obtain the different possible alignments.

Simply following all possible paths leading to optimal alignments of the sequences may be computationally intractable as the simple difference in example (1) already allows the three optimal alignments shown in (12).

(12) a. v _ i n s
v m _ _ s

b. v i _ n s
v _ m _ s

c. v i n _ s
v _ _ m s

As we are only interested in inducing alignments of annotations, the above differences do not influence the result. Hence, we only follow alternative paths when annotation boundaries are involved. Furthermore, we exploit inequality (1) (see Section 5) that holds for the dissimilarity measure that we use, and bias the alignments towards aligning annotations by aligning boundaries if possible. In (13) only the second alignment is produced.

[3] Note that this method requires the annotations of one annotator to be non-overlapping. Otherwise, the character denoting the end of an annotation can be ambiguous.

(13) a. { n e } { m a g }
{ - - - - m a g }

b. { n e } { m a g }
- - - - { m a g }

Aligning the text used for our experiments with its shuffled version (see Section 7), where the text, the segmentation and the categories are changed, and the magnitude was set to 1, leads to only one annotation alignment in the mean produced by this method (out of ten runs, only in one run two alignments were produced).

5 Calculating the observed disorder

As γ (Mathet et al., 2015), ${}_t\gamma$ is calculated based on the disorder of an optimal alignment ($\delta(a)$) between the annotations from different annotators. An alignment $\bar{a}$ is considered optimal when it minimizes the disorder. Unlike Mathet et al. (2015), we do not consider all possible alignments between annotations when looking for the optimal alignment but only the alignments that result from an optimal sequence alignment as described in the previous section. Annotations from different annotators are aligned when they cover the same span in the aligned sequences. Therefore, for each of the optimal sequence alignments exactly one alignment of annotations is defined consisting of unitary alignments ($\breve{a}$) between annotations or annotations and empty elements ($\emptyset$).

Following Mathet et al. (2015), the disorder of an alignment is defined as

$$\bar{\delta}(\bar{a}) = \frac{1}{\bar{x}} \sum_{i=1}^{|\bar{a}|} \breve{\delta}(\breve{a}_i)$$

where $\breve{\delta}$ is the **dissimilarity** between the aligned annotations. An alignment of an annotation with the empty element has a dissimilarity of $\Delta_\emptyset$ (cf. Mathet et al. (2015)).

We define the dissimilarity of an alignment of two annotations u and v as

$$d_t\gamma(u,v) = \frac{1}{n+1}(d_t(text(u),text(v)) + \sum_{i=1}^{n} d_i(feat_i(u),feat_i(v)))$$

where n is the number of features of the annotations (cf. Section 2). d_t is a dissimilarity measure between the texts covered by the annotations and the d_i are dissimilarity measures between the feature values. For the evaluation, we use the simple nominal dissimilarity measure which is 0 in the case of equality and $\Delta_\emptyset$ in the case of inequality for all d_x. Other d_x are usable as well, e.g., d_{cat} as described by Mathet et al. (2015), that takes overlaps between categories into account, or a string similarity metric such as the Levenshtein distance (Levenshtein, 1966) for textual differences.

Note that when using the dissimilarity measure exactly as described above, the following inequality holds:

$$d_t\gamma(u,v) \leq \Delta_\emptyset = \frac{1}{2}(d_t\gamma(u,\emptyset) + d_t\gamma(v,\emptyset)) \quad (1)$$

Therefore, the dissimilarity of an alignment is at least as high as the dissimilarity of an alignment where fewer annotations are aligned (i.e., it has more alignments with $\emptyset$). This means that many alignments created by optimal sequence alignments can be removed from the set of possible alignments for the calculation of ${}_t\gamma$.

As pointed out above, we do not consider positional differences in our dissimilarity measure. This is unproblematic since we do not align tokens that are not mapped to the same position by the sequence alignment process.

6 Calculating the expected disorder

For state-of-the-art IAA metrics, it is expected to take chance agreement into account (Artstein and Poesio, 2008). Our new measure ${}_t\gamma$ – like the original γ – measures disagreement between aligned annotations. The standard way of incorporating chance-correction to disagreement based measures is to use the ratio between the observed disagreement (D_o) and the expected disagreement (D_e), i.e. the disagreement that is expected when both of the annotators are guessing. Therefore, we define ${}_t\gamma$ exactly as γ as $1 - \frac{D_o}{D_e}$.

We follow Mathet et al. (2015) and compute D_e by sampling randomly generated annotation $\text{sets}_{\text{text}}$. Mathet et al. (2015) randomly create sets for which (i) the number of units per annotators, (ii) the categories, (iii) the length of the units of a given category, (iv) the length of gaps, and (v) overlaps between units of given categories are distributed as in the observed annotation set. Then they use these samples to estimate D_e.

This, however, estimates D_e when annotations are created without any pre-annotation which is

not the case for text corrections and tokenization in our case. Therefore, calculating D_e in this way would underestimate the actually expected disorder. Take for example two annotators annotating a text that was automatically tokenized with an error rate of 4% (Jurish and Würzner, 2013). In this case, only a small fraction of tokens needs to be changed. The expected agreement for two annotators highly disagreeing will still be substantially higher than the agreement to be expected when two tokenizations are created randomly. Therefore, we do not sample annotation sets that are randomly generated, but we create annotation sets by applying changes randomly to the pre-annotation.

Given the situation where tokenized transcriptions are annotated with POS tags, the creation of random annotation sets consists of three steps: Firstly, the text is changed, secondly the segmentation is changed, and thirdly the segments are annotated with POS tags. When modeling random annotations, we assume that all three steps are independent of each other. Further, we assume that the amount of changes (c_t and c_s for text and segmenation) the annotators perform follows a binomial distribution with the parameter n being the number of annotations. The parameter p can either be derived from the (known) quality of the pre-annotation, e.g., set to 0.04 for segmentation changes when the error rate of the tokenizer is 4%. Alternatively, it can be estimated from the observed differences between the annotation sets and the pre-annotation. Both methods can also be combined using maximum a posteriori (MAP) estimates for p (Manning and Schütze, 1999). Like Mathet et al. (2015), we use the annotations from all annotators for estimating distributions, i.e., treating annotators as interchangeable (Krippendorff, 2011).

Given the tokenized transcription, in the first step, we apply c_t text changes. For this c_t distinct annotations are chosen according to a uniform distribution. Then one of the three types of textual changes (insertion, deletion and substitution) is chosen from an equal distribution. For insertion and substitution a character is chosen based on the distribution of characters in the observed annotation set.

In a second step, the segmentation is changed by applying splits and mergers, i.e., adding or removing boundaries. This is done c_s times. For each change, one of the three operations (split, merge with left, merge with right) is chosen from a uniform distribution. Afterwards a segment is chosen again from an equal distribution, excluding the first segment for merge with left and the last segment for merge with right. Note that the annotations resulting from a split or a merger can be chosen for a subsequent change.

For the third step, i.e., labelling the tokens, in our case no pre-annotation is assumed. Therefore, we simply add labels to the tokens following the distribution of the labels in the observed annotation sets.

Using this method to create random annotation sets, we can estimate D_e by applying the same sampling method as Mathet et al. (2015).

7 Evaluation

For evaluating ${}_t\gamma$, we use the corpus shuffling method (Mathet et al., 2012). With this method a given reference annotation is changed randomly with a given magnitude m. Following Mathet et al. (2012), the shuffling is repeated with differing values of m (ranging between 0 and 1 with a stepsize of 0.05). For each of these values, the inter-annotator agreement is measured. These values show how the measure reacts to differences in two annotation sets of a specified magnitude. The values taken by the inter-annotator agreement measure should be (i) strictly decreasing with increasing magnitude m – i.e. reflect the increasing difference of the annotation sets and (ii) use the full range of possible values (Mathet et al., 2015).

We use a reference annotation set_{text} for the evaluation. The text has a length of 3,706 characters. The annotation set contains 608 tokens labeled with POS tags. We simulate a second annotation set_{text} by applying shuffling to this reference annotation. For the shuffling, three methods are applied: (i) textual change, (ii) segmentation change and (iii) label change. As shuffling all three types with the same magnitude is unrealistic (due to the pre-annotation bias), we calculate m_t for the magnitude of text changes, m_s for the magnitude of segmentation changes and m_l for the magnitude of labeling changes from a given m as follows: $m_t = 0.5 \times m$, $m_s = 0.1 \times m$ and $m_l = m$.

In each of the three steps, given a magnitude $0 \leq m_i \leq 1$, $c = m_i \times n$ changes are applied. For textual and category changes, the changes are applied to distinct annotations. As our parametrization of γ only measures if two aligned annotations

have the same text or not, each token is only considered once for text changes. The shuffling itself follows the same procedure as for the calculation of the expected disagreement.

We test three different settings that correspond to the three types of categorization tasks given in Fig. 1: (a) only categories are shuffled, (b) categories and segmentations are shuffled, and (c) categories, segmentations and the text is shuffled.

For the calculation of the expected disorder, we do not estimate the probabilities for the changes from the data to benchmark the influence of these parameters on the final agreement value. We evaluate three parameter settings: Firstly, we set the probability for text (p_t) and for segmentation changes (p_s) both to 0, i.e., the expected disorder is calculated for pure categorization (Cat). Secondly, we simulate the situation, where a text that is born-digital is automatically tokenized with an error rate of 4% (Jurish and Würzner, 2013), consequently p_t is set to 0 and p_s to 0.04 (Cat + Seg). Thirdly, we simulate the situation, where a text is automatically transcribed and tokenized afterwards with 25% of the tokens needing a textual correction. p_t and p_s are therefore set to 0.25 and 0.04 respectively (Cat + Seg + Text). Note, that the values for m_t and m_s limit the magnitude of the shuffling to approximately twice the expected error rate.

As both the shuffling and the calculation of the expected disorder is randomized, we repeat each step ten times. Figure 2 gives the mean values. The error bars denote the standard error.

For comparison, we used the DKPro Agreement package (Version 2.1.0) (Meyer et al., 2014) to compute α for the pure categorization setting and α_U with aggregation over categories for the categorization and segmentation setting. We also used the software supplied by the authors of γ[4] to calculate gamma for the categorization and segmentation settting. We only calculated γ for one shuffling, and only for magnitudes 0, 0.25, 0.5, 0.75 and 1.

As can be seen from Fig. 2, ${}_t\gamma$ shows an almost perfectly linear response to the increasing magnitude of the shuffling. The parametrizations expecting less change are always lower than the other parametrizations (except in the case of perfect agreement). This is expected as more agreement is attributed to chance.

[4] `https://gamma.greyc.fr` (Version 1.0).

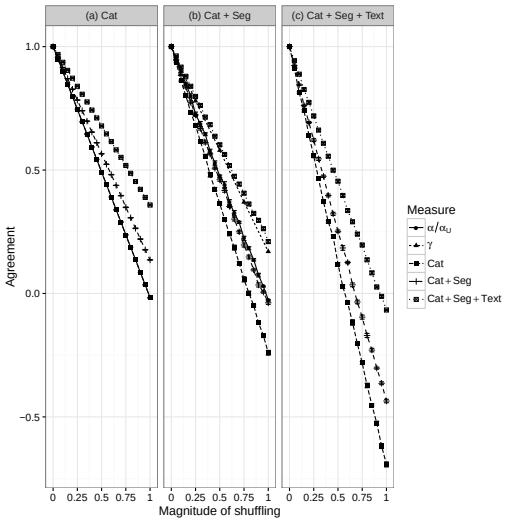

Figure 2: Evaluation results

When only the categories are shuffled, the parametrization of ${}_t\gamma$ for pure categorization covers the full range between 1 and 0, i.e., between perfect agreement and chance agreement. In this setting it behaves indistinguishably from α. When expecting errors in the transcription and segmentation, the agreement values stay above 0, reflecting the fact that the perfect agreement concerning the text and the segmentation is better than expected by chance. Consequently, the values of ${}_t\gamma$ can go below 0 in the other settings – as there are disagreements in the segmentation and/or the text not expected by chance. This differs from what Mathet et al. (2015) expect and is due to the fact that the parameters for the expected disorder calculation are not estimated from the observed annotation sets but are fixed. When expecting categorization and segmentation changes, ${}_t\gamma$ behaves similarly to α_U when categories and segments are shuffled. As expected, the original γ overestimates the amount of agreement as it does not take the pre-annotation into account.

The agreement value with settings for the expected agreement corresponding to the shuffling scenario is close to 0 when m is close to 1. The fact that it is slightly below 0 is due to the fact that $m_s = 0.1$ is slightly higher than $2 \times p_s = 0.08$.

8 Conclusion and further work

We presented text-gamma ${}_t\gamma$, a derivation of γ (Mathet et al., 2015), to measure inter-annotator agreement for categorization tasks where the annotators are allowed to change the underlying text and the segmentation during the annotation pro-

cess as it is done when annotating non-standard data that is not born-digital. The basis of our method is to align the texts using sequence alignment to create alignments of the annotations. The best of these alignments is chosen using a special dissimilarity measure. The inter-annotator agreement is measured on the basis of the mean dissimilarity between the aligned annotations. A practical point not addressed so far is that the resulting optimal alignment between the annotations can be used to show the annotators cases where they disagree and to analyze these deviations between the annotators.

For chance correction, we introduced a simple model to obtain expected disorders. To take the influence of the pre-annotation into account, our model does not model the creation of an annotation from scratch but starting with a given annotation set, random changes are applied.

Our evaluation using corpus shuffling showed that ${}_t\gamma$ reacts with a linear decrease to deviations between two annotation sets with increasing magnitude.

In its current form, ${}_t\gamma$ has some limitations. It assumes that the annotations cover the whole text as, e.g., with tokenization (with the possible exception of whitespace) and are not overlapping. While ${}_t\gamma$ is usable with annotation sets that do not cover the whole text, it is important to bear in mind that only annotations are compared. Textual changes outside of annotations have no influence on the agreement value. For non-overlapping unitizations, one possible way to take such changes into account would be to transform them into segmentations by treating gaps as annotations with the special type *gap* and ensure that gaps are not aligned with annotations of other types.

Changing the order of segments in the text is another point that ${}_t\gamma$ in its current form does not handle. This can appear, for example, when annotators disagree on the location where interlinear additions are added. The global sequence alignment used to infer possible alignments does not allow alignments between identical text segments to appear in different positions or – in other words – edges aligning annotations do not cross.

In the case of overlapping annotations of the same type, aligning annotations by inserting the annotation boundaries into the texts and aligning the text does not work as is since closing boundaries may be ambiguous in the case of overlaps.

Furthermore, our evaluation only took one type of annotation (tokens), categorization with one set of categories (POS) and two annotators into account and used a basic dissimilarity metric for nominal categories. It will be interesting to see how ${}_t\gamma$ behaves with more than two annotators, other dissimilarity metrics that take overlaps between categories into account, and with annotation sets containing multiple types of segments (e.g. *tokens* and *sentences* as in the annotation task described by Čibej et al. (2016)) and/or multiple labels for annotations (e.g. POS tag and lemma).

Regarding the chance correction, we introduced a simple model to randomly change the annotation. This model introduced some simplifications, for example, the three parts of the annotation process are modelled independently and only one edit operation is allowed for each token. Further work could introduce a more detailed model for chance correction, for example introducing further edit operations for a token with a decreasing probability.

Resources

We provide the following resources together with the paper:

(i) An implementation of the IAA measure described in the paper. The program takes CorA-XML-files, the output format of the annotation tool CorA[5], as input and outputs the IAA value and an alignment of the annotations for further analysis. It can be found at `https://github.com/fab-bar/TextGammaTool`.

(ii) An org-file[6] containing the paper and the complete code that was used to run the experiments, making the work reproducible. It can be found at `https://github.com/fab-bar/paper-KONVENS2016`.

Acknowledgements

The work of the first and second authors has been funded by the DFG. We would like to thank the anonymous reviewers for their helpful remarks and Piklu Gupta for improving our English. All remaining errors are ours.

[5] `https://www.linguistics.ruhr-uni-bochum.de/comphist/resources/cora/index.html`

[6] `http://orgmode.org/`

Primary data

Verl. Sohn *De parabell vam vorlorn Szohn*. Printed 1527 in Magdeburg by Burchard Waldis. Transcribed in the DFG-funded project "Referenzkorpus Mittelniederdeutsch / Niederrheinisch (1200 - 1650)".

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Fabian Barteld, Renata Szczepaniak, and Heike Zinsmeister. 2014. The definition of tokens in relation to words and annotation tasks. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 250–258, Tübingen, Germany.

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 164–171, Stroudsburg, PA. Association for Computational Linguistics.

Marcel Bollmann, Florian Petran, Stefanie Dipper, and Julia Krasselt. 2014. CorA: A web-based annotation tool for historical and other non-standard language data. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH) @ EACL 2014*, pages 86–90, Gothenburg, Sweden. Association for Computational Linguistics.

Maria Chatzou, Cedrik Magis, Jia-Ming Chang, Carsten Kemena, Giovanni Bussotti, Ionas Erb, and Cedric Notredame. 2015. Multiple sequence alignment modeling: methods and applications. *Briefings in Bioinformatics*, pages 1–15.

Jaka Čibej, Darja Fišer, and Tomaž Erjavec. 2016. Normalisation, Tokenisation and Sentence Segmentation of Slovene Tweets. In *Proceedings of the LREC-Workshop on Normalisation and Analysis of Social Media Texts (NormSoMe)*, pages 5–10, Portorož, Slovenia.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, Wim Peters, and Leon Derczynski. 2014. Developing Language Processing Components with GATE Version 8. University of Sheffield Department of Computer Science.

Stefanie Dipper, Karin Donhauser, Thomas Klein, Sonja Linde, Stefan Müller, and Klaus-Peter Wegera. 2013. HiTS: ein Tagset für historische Sprachstufen des Deutschen. *JLCL*, 28(1):1–53.

Karën Fort and Benoît Sagot. 2010. Influence of Pre-annotation on POS-tagged Corpus Development. In *Proceedings of the Fourth Linguistic Annotation Workshop, ACL 2010*, pages 56–63, Uppsala, Sweden. Association for Computational Linguistics.

Chris Fournier. 2013. Evaluating Text Segmentation using Boundary Edit Distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1702–1712, Sofia, Bulgaria. Association for Computational Linguistics.

Bryan Jurish and Kay-Michael Würzner. 2013. Word and Sentence Tokenization with Hidden Markov Models. *JLCL*, 28(2):61–83.

Tom Kenter, Tomaž Erjavec, Darja Fišer, and others. 2012. Lexicon construction and corpus annotation of historical language with the CoBaLT editor. In *roceedings of the 6th EACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 1–6, Avignon, France. Association for Computational Linguistics.

Amit Kirschenbaum. 2013. Unsupervised Segmentation for Different Types of Morphological Processes Using Multiple Sequence Alignment. In Adrian-Horia Dediu, Carlos Martín-Vide, Ruslan Mitkov, and Bianca Truthe, editors, *Statistical Language and Speech Processing*, number 7978 in LNAI, pages 152–163. Springer, Berlin, Heidelberg.

Klaus Krippendorff. 1980. Chapter 12. In *Content Analysis: An Introduction to Its Methodology*, pages 129–154. Sage, Beverly Hills, CA, 1 edition.

Klaus Krippendorff. 1995. On the reliability of unitizing continuous data. *Sociological Methodology*, 25(47):47–76.

Klaus Krippendorff. 2011. Agreement and Information in the Reliability of Coding. *Communication Methods and Measures*, 5(2):93–112.

Klaus Krippendorff. 2013. Chapter 12. In *Content Analysis: An Introduction to Its Methodology*, pages 267–328. Sage, Thousand Oaks, CA, 3 edition.

Klaus Krippendorff. 2015. Replacement of Section 12.4 (revised version 2015.9.23). In *Content Analysis: An Introduction to Its Methodology*, pages 309–319. Sage, Thousand Oaks, CA, 3 edition. Available at http://web.asc.upenn.edu/usr/krippendorff/dogs.html (2015-08-05).

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Johann-Mattis List. 2012. Multiple sequence alignment in historical linguistics. In Enrico Boone, Kathrin Linke, and Maartje Schulpen, editors, *Proceedings of ConSOLE XIX*, pages 241–260.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, London, England.

Yann Mathet, Antoine Widlöcher, Karën Fort, Claire François, Olivier Galibert, Cyril Grouin, Juliette Kahn, Sophie Rosset, and Pierre Zweigenbaum. 2012. Manual corpus annotation: Giving meaning to the evaluation metrics. In *Proceedings of COLING 2012: Posters*, pages 809–818, Mumbai, India.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The Unified and Holistic Method Gamma (γ) for Inter-Annotator Agreement Measure and Alignment. *Computational Linguistics*, 41(3):437–479.

Christian M. Meyer, Margot Mieskes, Christian Stab, and Iryna Gurevych. 2014. DKPro Agreement: An Open-Source Java Library for Measuring Inter-Rater Agreement. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 105–109, Dublin, Ireland.

Ngoni Munyaradzi and Hussein Suleman. 2013. Quality Assessment in Crowdsourced Indigenous Language Transcription. In Trond Aalberg, Christos Papatheodorou, Milena Dobreva, Giannis Tsakonas, and Charles J. Farrugia, editors, *Research and Advanced Technology for Digital Libraries*, number 8092 in LNCS, pages 13–22. Springer, Berlin, Heidelberg.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 23–25, Istanbul, Turkey.

Lev Pevzner and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Jelena Prokić, Martijn Wieling, and John Nerbonne. 2009. Multiple sequence alignments in linguistics. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*, pages 18–25. Association for Computational Linguistics.

Tomáš Valenta, Luboš Šmídl, Jan Švec, and Daniel Soutner. 2014. Inter-Annotator Agreement on Spontaneous Czech Language. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*, number 8655 in LNCS, pages 390–397. Springer International Publishing, Switzerland.