UNIVERSITY OF TWENTE.

Faculty of Behavioural Management and Social Sciences

Linguistic Accommodation between Leaders and Followers

Jakob A. Buske
B.Sc. Thesis
July 2019

Supervisors:
A.M.G.M. Hoogeboom MSC
Prof. dr. C.P.M. Wilderom

Human Resource Development Group
Faculty of Behavioural Management
and Social Sciences
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Recent research has highlighted the need for studies investigating leaders and followers not only as isolated individuals but as interacting parties. Drawing on the Communication Accommodation Theory (CAT), this study uses the concept of linguistic accommodation, the adaptation of one's linguistic style to become more or less similar towards another person's linguistic style to research leader-follower interactions. Specifically, we investigated the relations of follower-to-follower, follower-to-leader, and leader-to-follower linguistic accommodation with team and leader effectiveness. To that means, a large data set including 75 teams in Dutch public sector staff meetings was acquired. Moreover, we extend recent developments in the measurements of accommodation by proposing a new valid and accurate method to measure linguistic accommodation. As such, this study yielded a new perspective on the interactions between leaders and their followers by applying the CAT to a leadership context. No relationship between accommodation of any aforementioned direction and effectiveness ratings was found, which could be attributed to a general lack of accommodative behaviors. Several explanations for the lack of accommodation were identified and the need for further research on the nature of accommodation as well as its measurement was established.

Recently, more and more scholars in the leadership literature voiced the need to investigate the specific interactions between leaders and their followers (Ford & Harding, 2018). This could yield further insight into the underlying social dynamics that drive effective workplace behaviors. However, studies applying this more interactional focus have been sparse (Ford & Harding, 2018). This could be due to a lack of objective and valid measures as well as theories on the organizational context in which to carry out interactional studies. One theory capable of providing insight into effective leader-follower interactions is the so-called *Communication Accommodation Theory* (CAT; Giles, Coupland, & Coupland, 1991), which has been applied to a broad range of communication settings, such as psychotherapy (Lord, Sheng, Imel, Baer, & Atkins, 2015), education (Chen, 2019), and hostage negotiation (Taylor & Thomas, 2008). An advantageous characteristic of the CAT is that it does not merely describe the individual's behavior, but also the interactions between two and more people. Thus, CAT offers more insight into how both leaders and followers interact. This study draws on the CAT to investigate leader-follower interactions and their relations with team and leader effectiveness in a Dutch, public-sector workplace context.

Communication Accommodation Theory

The CAT is grounded in the *social identity theory,* which entails that people's social identity is created by one's perception of unity with a group and that people aim to behave coherently with the identity adopted (Ashforth & Mael, 1989). The CAT builds upon the social identity theory by predicting that humans adapt their communication based on their interlocutor’s style and their “own desire to establish and maintain their (...) social identity” (Dragojevic, Gasiorek, & Giles, 2015, p. 3). Accordingly, the communication process between two or more persons is a trade-off between the maintenance of one’s own identity versus the creation of (perceived) social affiliation within a group. The process of adapting one's communication style towards others is called accommodation. Accommodation can be examined by observing the number of occurrences of interacting people's behaviors.

Accommodation among people who interact with each other at work can take three different directions. First, people can converge towards each other. With *convergence,* speakers change their own communication styles to become more similar to their interlocutors. As a result, converging speakers are often evaluated more positively on a broad range of social dimensions (e.g. Soliz & Giles, 2014). Second, *diverging* speakers adapt their communication style to be more different compared to the communication style of their interlocutors. Third, speakers who neither converge nor diverge are considered to express *maintenance.*

Accommodation can occur on different dimensions, such as communication strategies, gazing patterns, use of accents and dialects, posture, or the topic addressed (for an overview, see Dragojevic et al., 2015). One dimension commonly used is linguistic accommodation, where the interlocutors adapt their choice of words to each other. Linguistic accommodation between two persons has been found to be related to a wide range of constructs. Converging confederates led people to admit to more socially undesirable behaviors (Guéguen, 2013); converging sales clerks were perceived as being more competent and generated more sales (Jacob, Guéguen, Martin, & Boulbry, 2011); and converging hostage negotiators have been found to be more successful (Taylor & Thomas, 2008). Moreover, research indicates that linguistic convergence in communication is positively related to empathy ratings of both leaders (Meinecke & Kauffeld, 2018) and therapists (Lord et al., 2015). Linguistic convergence also led to reduced social distance (Chung & Pennebaker, 2007) and increased ratings of quality of contact as well as relational solidarity (Soliz & Giles, 2014). Interactions that are characterized by divergence or maintenance are generally perceived as less harmonious and are related to lower conversational effectiveness (Soliz & Giles, 2014).

Accommodation between Team Members

Besides the effects of accommodation of dyads, convergence and divergence can also occur in teams. Converging communication between team members was related to improved team cohesion and team effectiveness (Gonzales, Hancock, & Pennebaker, 2010; Scissors, Gill, & Gergle, 2008; Yilmaz, 2016). In general, convergence in communication between team members might facilitate a shared mental model within the team, which would, in turn, lead to higher levels of team effectiveness (Cannon-Bowers, Salas, & Converse, 1993; Gonzales et al., 2010). Other research proposed that convergence in teams might also improve the clarity of language and quality of comprehension between the communicating parties, which could also have a positive impact on team effectiveness (Pickering & Garrod, 2004). In general, convergence between all team members in a team has been shown to invoke favorable results. Accordingly, we expect to replicate the findings and predict:

H1: *High follower-to-follower linguistic convergence is positively related to team effectiveness.*

While follower-to-follower accommodation could indicate that the followers share one social common identity, follower-to-leader accommodation should indicate that the followers not merely share one social identity, but that they share their social identity with their leader. Hence, followers would accommodate towards their leaders to adopt their social identities. Based on the previously discussed literature, this shared social identity might positively impact the team's effectiveness. One could argue that, despite their special statuses, leaders are still members of their teams. Then, the followers' convergence towards their leaders should also facilitate a shared mental model (Gonzales et al., 2010) and foster the overall quality of communication (Pickering & Garrod, 2004) and, thus, increase the team's effectiveness (Cannon-Bowers et al., 1993; Yilmaz, 2016). Moreover, followers' linguistic convergence towards their leaders might have a larger impact on team effectiveness than towards other followers as the leaders and, by extension, leader-follower interactions could be more influential on the overall team dynamics than a regular follower and the corresponding follower-follower interactions (Zaccaro, Rittman, & Marks, 2001). As such, the previously discussed findings could also extend to follower-leader relationships.

H2A*: High follower-to-leader linguistic convergence is positively related to team effectiveness.*

Accommodation in Leader-to-Follower Communication

Despite the extensive research conducted on the effects of accommodation between team members, studies considering the particular role of the leader in the accommodation process have been sparse. However, studies already proposed that differences in accommodation prevalence can be due to differences in status. Generally, individuals low in status or power tend to converge more towards individuals high in status or power (Danescu-Niculescu-Mizil, Lee, Pang, & Kleinberg, 2012; Jones, Cotterill, Dewdney, Muir & Joinson, 2014; Liao, Bazarova, & Yuan, 2016; Muir, Joinson, Cotterill & Dewdney, 2016). This phenomenon, often called asymmetric convergence, can be explained by the communication accommodation theory because lower status individuals seek acceptance, whereas higher status individuals do not perceive the need to do so (Giles et al., 1991). Grounded in these findings, leaders should be expected to engage in less convergence than their followers.

The expectation that leaders should converge less towards their followers does not entail that leader-to-follower accommodation is less important than follower-to-follower accommodation. Contrary, the leader has a significant influence on the team dynamics (e.g. Zaccaro et al., 2001). Hence, the leader's adoption of the team's social identity should not only improve the social climate in the team but should also serve as an indicator of it. Accommodation of leaders towards their followers might not only be as important as the follower-to-follower accommodation but also could have a larger impact due to the special role of the leader.

H2B: *High leader-to-follower linguistic convergence is positively related to team effectiveness.*

Although leaders are expected to show less convergence in general due to their higher formal status, prior research hints to high leader-to-follower communication convergence being positively related to leadership effectiveness. Many constructs that have been related to communication accommodation directly or indirectly impact leadership effectiveness as well. For example, a high need for affiliation (Dragojevic et al., 2015; Steinmann, Dörr, Schultheiss, & Maier, 2015), empathy (Mumford, Todd, Higgs, & McIntosh, 2017) and affective trust (Miao, Newman, & Huang, 2014) are all positively related to accommodation. Moreover, the only study conducted to establish the relation between leader-to-follower accommodation and leader effectiveness found that presidential candidates who engaged in convergence in presidential debates had higher success rates in later polls (Romero, Swaab, Uzzi, & Galinsky, 2015). Only one study researched the link between leader accommodation and effectiveness. However, accommodation already has been found to relate to a series of correlates of leader effectiveness. Based on these tentative results, the authors predict:

H2C: *Leaders displaying more linguistic convergence towards followers are more effective.*

This study adds to the literature by following Ford and Harding's (2018) call for research investigating the interactions between leaders and their followers and hence, observing the team as such instead of a mere collection of isolated individuals. To that means, CAT is used to gain insight into the underlying social processes between leaders and their teams as well as the explanatory factors of leader and team effectiveness. To the researchers' knowledge, no prior study has investigated the relationship between linguistic accommodation and leader effectiveness before. Hence, this study expands the sparse literature on linguistic accommodation and leader effectiveness while building on prior findings on linguistic accommodation between followers.

Method

The study at hand uses video capture and text mining approaches to investigate the relations between team and leader effectiveness and intra-group and leader-to-follower accommodation.

Participants

75 work teams consisting of 5 to 34 members (mean = 13.4, sd = 5.72) were randomly sampled in a large Dutch government institution. Of the team leaders, 18 were female and 54 male, with an average age of 51.36 (min = 27, max = 62, sd = 7.78). The leaders worked 23.66 years (sd = 13.73) for the institution. The 917 followers (522 females, 319 males) were, on average, 48.55 (sd= 10.81) years old. During the staff meetings, the leaders voiced on average 6129.7 words (sd = 3318.6), while each follower used, on average, 514.2 words (sd = 773.5). All participants signed an informed consent. This study was approved by the BMS ethics board of the University of Twente.

For each team, a random staff meeting was selected. The meeting room was fitted with wide-angle web-cameras directed at the followers as well as one camera directed at the leader. The researchers paid attention to select technical equipment as small as possible to ensure that the obtrusiveness of the video cameras is limited. During the meeting itself, the researchers left the room in order to avoid being distracting to the participants. After the video data were collected, the recorded meetings were transcribed by native Dutch speakers. Here, a manual for transcription of the videos was used to ensure that all transcripts complied with the same standard. The resulting transcripts were then used for the data analysis.

Measures

Markers used to Measure Accommodation.

The measurement of accommodation relies on the use of marker words. Marker words are an explicitly defined group of words whose occurrence is observed within an utterance. This study used 14 different marker categories of function words. Function words do not have a semantic meaning on their own. Instead, their meaning arises out of the context of the conversation. This allowed the measurement of accommodation to be independent of the topic of a conversation and thus enables researchers to compare communication of different meetings.

The marker groups used were identified by Danescu-Niculescu-Mizil, Gamon, and Dumais (2011) within the dictionary used by LIWC, the standard software package used for analyzing accommodation (Pennebaker, Boyd, Jordan, & Blackburn, 2015). The 14 marker groups, as well as examples in both English and Dutch, can be inspected in Table 1. Despite the, compared to other text analysis methods, low volume of words used as markers with a total of 1.165 words, prior research indicated that these words make up about 55 percent of one's day-to-day language use (Tausczik & Pennebaker, 2010). As the meetings were recorded for this study were held in Dutch, the Dutch translation of the LIWC dictionary was used (Boot, Zijlstra, & Geenen, 2017).

Table 1
Marker Word Categories developed by Pennebaker et al. (2015), translated by Boot et al. (2017)

Category	**Description**	**Number of words in this category**	**Example (Dutch)**	**Example (English)**
Articles	Adjective that is only used before a noun.	11	de, het	the
Certainty	Indicating certainty regarding a statement.	194	onmiskenba*, immer	distinctively, always
Conjunctions	A word used for connecting clauses or sentences.	44	aangezien, zodat, teneinde	because, so that, in order to
Discrepancy	Indicating a discrepancy between two entities.	135	onvoldoende, hoeft, onmogelijk*	insufficient, left out, impossible
Exclusive	Indicating that something is excluded.	35	buitensluiten, zonder, tenzij	(to) exclude, without, unless
Inclusive	Indicating that something is included.	54	en, inbegrepen, optellen	and, included, (to) add
Impersonal Pronouns	Pronoun used without definite reference.	87	dit, iemands, eenieder	this, someone’s, everyone
Negations	Indicating that something is not true or contained in an entity.	24	desondanks, nee, niente	nevertheless, no, nothing
Prepositions	Word indicating the relation of several nouns.	83	niettegenstaande, uit, ingevolge	nevertheless, out of, in response to
Quantifiers	Indicating a quantity of an entity.	182	reeks*, velen, gedeeltes	series, many, excerpts
Tentative	Indicating that somebody is not sure about something.	279	eigenlijk	actually

1st Person Singular	Indicating that a person is referring to him-/herself.	12	ikzelf, mijne	myself, my
1st Person Plural	Indicating that a person is referring to a group of which that person is part of.	7	onszelf, ons	ourselves, us
2nd Person	Referring to others as oneself.	18	jou, jouwe	you, your

Measure of Linguistic Accommodation.

Several statistical procedures with varying degrees of complexity are available to compute accommodation scores based on the counts of occurrences of the marker words in the utterances. The most commonly used method is the *Linguistic Style Matching* (LSM; e.g. Ireland et al., 2011). LSM is implemented in the often-used software LIWC (Pennebaker et al., 2015) and compares the total marker use per category of two speakers. Ireland and colleagues (2011) defined LSM for any category marker:

$$LSM_{marker} = 1 - \frac{|\, marker_1 - \, marker_2 \,|}{marker_1 + \, marker_2 + 0.0001}$$

Here, $marker_1$ and $marker_2$ represent the percentage of words that were markers used throughout the whole conversation for person one and two, respectively. A small number (i.e. 0.0001) is added to ensure that the divisor does not become zero even if none of the marker words have been used. Critics have been noting that LSM merely compares the overall marker word usage of the persons of interest and does not take into account *when* these words were used (Doyle, Yurovsky, & Frank, 2016). As a result, the LSM cannot indicate whether two persons indeed adapted their language (i.e., directly matching their communication after a team member spoke) to each other or whether they shared the same communication style from the beginning on due to cultural similarities or random chance. Hence, LSM serves as a measure of stylistic cohesion or homophily rather than as an actual measure of accommodation (Danescu-Niculescu-Mizil et al., 2011; Doyle et al., 2016).

In order to separate actual accommodation from mere homophily, Danescu-Niculescu-Mizil et al. (2012) formulated a probabilistic framework, later called the *Subtractive Conditional Probability* (SCP; Doyle et al., 2016), as a new measure for accommodation:

$$SCP = P(B \mid A) - P(B)$$ [1]

where accommodation is defined as the conditional probability that B uses a word of the marker category given that A used a word of the same category minus the overall probability of B using the marker. With this formula, deviances from B’s marker usage, as a reaction to A’s marker usage, could be identified, which in turn could be interpreted as accommodation. By using conditional probabilities, the SCP did not merely compare the overall marker usage of two persons but introduced a temporal aspect: B's marker usage would only be considered accommodation if A also used the marker in the preceding utterance. However, the boundaries (i.e., the minimum and maximum scores) of the SCP depend on A’s marker usage baseline (i.e. the overall probability of a person using the marker word in an utterance), meaning that the resulting scores are not necessarily comparable to each other. Moreover, the SCP’s accuracy in estimating accommodation decreases as the baseline of A approaches one. This is because P(B) approaches P(B|A) as the baseline of A (P(A)) increases. As a result, the SCP scores approach zero independently of the actual accommodation. This notion is

1 Using the formalization of Doyle et al. (2016)

supported by a simulation conducted prior to this study (see Appendix A) as well as Doyle et al. (2016). Hence, a new measure was needed that maintained the conditional character of the SCP while also ensuring accurate and comparable measures.

To overcome the problems posed by the SCP, Doyle et al. (2016) proposed the *Hierarchical Alignment Model* (HAM). The HAM estimates accommodation by subtracting the probability that A does not but B does use the marker from the probability that both A and B use the marker in a hierarchically nested inverse-logit space. This eliminates the problems of the SCP by replacing the baseline correction P(B), which was influenced by P(A), with P(B | ┐ A). However, both the HAM and the SCP assume that the usage of a marker is binary (i.e. either an utterance contains a marker or not). As such, both models ignore the fact that the more words an utterance contains, the more likely it becomes that any marker word is used. As a result, both models could misinterpret differences in utterance lengths as accommodation. To solve this potential source of bias, Doyle & Frank (2016) developed the *Word-based Hierarchical Alignment Model* (WHAM). Instead of examining marker usage of B per utterance, the WHAM observes B's marker usage per word used. This way, it corrects for the utterance length of B. However, there remains to be no correction for the utterance length of A. Nonetheless, the formula proposed in the WHAM constitutes the most valid method of operationalizing linguistic accommodation and is, hence, used in this study[2]. Here, the model will be referred to as the *Simplified Accommodation Model* (SAM).

Preparation of the Data.

Before accommodation scores could be computed with the SAM, the data needed to be prepared. The single utterances in the transcripts were sorted into target-reply pairs where the reply utterance directly followed the target utterance. Then, the frequencies of the

[2] Doyle and Frank (2016) make use of hierarchical and Bayesian elements in the WHAM. These elements have been left out as they were not needed in this study.

markers established by the LIWC dictionary (Pennebaker et al., 2015) as well as the word count were computed for each utterance. After the completion of these preparatory steps, the SAM could be applied.

Application of the SAM Formula.

The SAM defines accommodation as the inverse logit conditional probability that a word in the reply is a marker given that the target utterance contains a marker of the same category minus the inverse logit conditional probability that a word in the reply is a marker given that the target utterance does not contain a marker. This can be formally expressed as:

$$\text{Accommodation} = \mathit{logit}^{-1}\ P(m \in \text{words of reply} \mid m \in \text{target}) - \mathit{logit}^{-1}\ P(m \in \text{words of reply} \mid m \notin \text{target})$$

As both elements of the term are probabilities conditional on the marker prevalence in the target utterance, the target's baseline usage of the marker does not affect the final accommodation score. Furthermore, the second element serves as a control for the respondent's baseline usage of the marker. Since one accommodation score is computed per marker category, the mean accommodation is calculated for all categories. Accordingly, the accommodation score describes the difference in probabilities of the respondent's marker use dependent on the target's marker usage for all categories.

Simulating the SAM.

A simulation study was conducted to evaluate the characteristics of the SAM. To that means, utterance pairs were generated from random baseline and accommodation parameters. Then, the SAM was used to estimate accommodation. Furthermore, the SCP introduced by Danescu-Niculescu-Mizil and colleagues (2011) was also used for comparative purposes. The simulation study supported the notion that the latter method was biased by the marker usage baseline of the target. Contrary, the SAM provided an unbiased estimate of accommodation.

Scatterplots showing the relation between true and estimated accommodation values for both models can be inspected in Appendix A. Lastly, the minimum amount of utterances required for an accurate estimation was established. The results indicated that a total of 100 observed utterance pairs was sufficient to estimate accommodation with moderate accuracy. However, this threshold relates to the total amount of utterance pairs, meaning that the 100 observations can be distributed upon the 14 categories given that all categories are manifestations of one common latent factor, accommodation. As a result, the SAM proves to be suitable also for sparse data sets. Both the ability to provide an unbiased estimate as well as the robustness towards sparse data sets makes the SAM a suitable method for operationalizing accommodation.

Accommodation in teams.

All common models that operationalize accommodation assume conversations with two participants. While this dyadic structure is convenient for mathematical reasons, it does not apply to the study at hand, where conversations of teams were recorded. However, this problem can be circumvented by the appropriate selection and aggregation of the textual data of more persons comprising one group. The accommodation of the leaders towards their followers constitutes a case where accommodation of one individual towards the remaining group needs to be measured. This was done by following Danescu-Niculescu-Mizil et al. (2012), who aggregated all data of the remaining group assuming that the followers constituted one person. Accordingly, all utterances pairs where the leader responded to any given follower were selected. Thus, interactions between the followers were ignored for this measurement.

While the accommodation of a leader towards a group constitutes a one-to-many problem, the measurement of the follower-to-follower accommodation can be referred to as a many-to-many problem. This problem was resolved by the same approach as for leader-to-follower accommodation. Hence, all utterances where one follower replied to another follower were collected and supplied to the SAM. The result then was used as the team's follower-to-follower accommodation score.

Leader Effectiveness.

To assess of leader effectiveness, the followers of each team were asked to respond to four items of the Multifactor Leadership Questionnaire 5X (Bass & Avolio, 1995). The MLQ is a leadership assessment questionnaire commonly used by researchers. For four leadership effectiveness items, an inter-item correlation of $\alpha = .89$ was obtained. Hence, these four items serve as a reliable measure of leadership effectiveness. A sample item is: "[The leader] is effective in meeting my job-related needs". The aggregated leader effectiveness scores ranged between 3.79 and 6.22 ($\textit{leader effectiveness}_{mean} = 5.37$, $\textit{leader effectiveness}_{sd} = .49$). Detailed descriptive statistics can be inspected in Table 2.

Table 2
Descriptive Statistics of Leader Effectiveness

	Minimum	Mean	SD	Maximum	Lambda 2
Leader Eff. - Individual Ratings	2.00	5.12	.87	7	.89
Leader Eff. - Aggregated to Group-Level	3.79	5.37	.49	6.22	

Team Effectiveness.

Four items developed by Gibson, Cooper, and Conger (2009) were given to the followers as a measure for team effectiveness. The inter-item reliability for these items was $\alpha = .89$. Due to these psychometric properties as well as their brevity, the items are a valid and suitable measure of team effectiveness. A sample item is: "[The team] continuously delivers high performance". The team effectiveness aggregated on a team level ranged from 3.83 to 6.2 ($\textit{team effectiveness}_{mean} = 5.03$, $\textit{team effectiveness}_{sd} = .57$). Detailed descriptive statistics can be inspected in Table 3.

Table 3
Descriptive Statistics of Team Effectiveness

	Minimum	Mean	SD	Maximum	Lambda 2
Team Eff. - Individual Ratings	1.5	4.96	1.03	7	.89
Team Eff. - Aggregated to Group-Level	3.83	5.03	.57	6.21	

Analysis

After the data collection, the data were analyzed with the statistical programming language R (R Core Team, 2018). Descriptive statistics of all accommodation values were computed. Inter-item reliability was established for the single accommodation scores per category. This step ensured that all marker categories were indeed measuring the common construct accommodation. Instead of the commonly used Cronbach’s alpha, Guttman’s lambda 2 is used, as this serves as a more accurate estimate for inter-item reliability (Sijtsma, 2009).

To test the four hypotheses, four separate Ordinary Linear Regression models were formulated. In the first three models, team effectiveness was used as the outcome and follower-to-follower, follower-to-leader, and leader-to-follower accommodation were used as the predictor variable, respectively. The fourth model predicted leadership effectiveness based on the leaders’ accommodation scores. The assumptions for all models were tested. Outliers were identified as data points with large standardized residuals above two as well as Cook’s Distance based on the degree of visual deviation in a plot (Fox, 1991), which presents the degree of influence an outlier has on the overall model (Field, Miles, & FIeld, 2012) and consequently removed.

Results

992 participants in 75 teams were observed during staff meetings and responded to surveys assessing team and leader effectiveness. The accommodation scores had a mean of $accommodation_{mean}$ = .00065 ($accommodation_{min}$ = -.0039, $accommodation_{max}$ = .00462, $accomodation_{sd}$ = .001). Moreover, the 14 categories used yielded an internal consistency of lambda 2 = .29 . Within the three directions of accommodation, differences in internal consistency became apparent: While both follower-to-follower and leader-to-follower accommodation shared low internal consistency (lambda 2 = .36), follower-to-leader accommodation yielded none (lambda 2 = -.03). Detailed accommodation scores by direction can be inspected in Table 4, correlations between the accommodation and effectiveness scores can be found in Table 5.

The summary statistics of the four regression models to test the hypotheses can be inspected in Table 6. The explained variance of all models was close to zero. Accordingly, none of the four models significantly predicted their respective dependent variable. As a result, the data did not support any of the hypotheses.

Table 4
Descriptive Statistics of Accommodation Scores

	Min	Mean	SD	Max	Lambda 2
Follower to Follower	-.0039	.001	.0014	.0046	.36
Follower to Leader	-.0001	.0000	.0006	.0001	-.03
Leader to Follower	-.0001	.0000	.0007	.0003	.36
Total	-.0039	.0007	.001	.0046	.29

Note. Total represents all communication measures aggregated independent of direction.

Table 5
Correlations between Effectiveness and Accommodation Scores

	Team Effectiveness	Leader Effectiveness	Leader to Follower Acc.	Follower to Leader Acc.	Follower to Follower Acc.
Team Effectiveness	1	.51	.00	-.15	-.07
Leader Effectiveness		1	.08	-.09	.04
Leader to Follower Acc.			1	.21	.00
Follower to Leader Acc.				1	.16
Follower to Follower Acc.					1

Ancillary analyses were conducted to investigate whether homophily might explain leader and team effectiveness rather than true accommodation. LSM (M = .06, SD = .01) values were computed and regressed on the effectiveness data. LSM could neither explain team ($F(1,73) < .01$, $p = .99$, adj. $R^2 = -.01$) nor leader ($F(1, 73) = 0.1$, $p = .75$, adj. $R^2 = -.01$) effectiveness.

Lastly, a simulation study was conducted to investigate how the internal consistencies of the SAM measures per category would react to changes in parameters. In line with the data

Table 6
Statistics on the four regression models used

Dependent Variable	Independent Variable	Adj. R^2	F-Statistic	p
Team Effectiveness	Follower-to-Follower Acc.	-.01	.36(1, 73)	.55
	Follower-to-Leader Acc.	-.01	.00(1, 73)	.95
	Leader-to-Follower Acc.	.01	1.67(1, 73)	.20
Leader Effectiveness	Leader-to-Follower Acc.	-.01	.56(1, 73)	.46

set used in this study, 5 utterances per case were used in this simulation. Assuming there was accommodation, the lambda 2 of the 14 categories was .88 . Assuming the underlying accommodation was equal to zero, the internal consistencies were close to zero. Moreover, the distribution of total SAM scores resembled the distribution of the data collected.

Discussion

This study investigated the relationships of team and leader effectiveness with accommodation in regular staff meetings. The descriptive statistics of the accommodation measures indicate that no true accommodation occurred, as all values were close to zero. The low internal consistencies of the 14 categories used tentatively supported this notion, indicating that the single measurements were impacted by chance rather than accommodation. In that case, accommodation could not influence the measures as there was no accommodation. A simulation study further yielded twofold support for this interpretation. First, it was shown that if there had been any accommodation, the internal consistencies of the 14 categories would have been higher. Second, applying the SAM to a dataset generated with the assumption of no accommodation and otherwise parameters paralleling the study at hand, the resulting distribution of accommodation scores resembled the accommodation scores observed in the study at hand. As such, it can be concluded that no accommodation as modeled in the SAM occurred in the data collected.

No significant relationship was found for either of the constructs. Hence, all hypotheses had to be rejected. Further analyses showed that LSM, a measure of homophily, also related to neither team nor leader effectiveness. As a result, none of the measures collected from the meetings significantly predicted effectiveness ratings collected for this study.

Four possibilities could cause the findings of this study. Firstly, the participants could have, indeed, not accommodated to each other. Here, a lack of accommodation could be explained by a different sample used in this study compared to the remaining literature researching accommodation. Commonly, research on the CAT, including linguistic accommodation, draws on samples with participants that did not know each other before the study was conducted. In the paper at hand, however, participants have been with their respective team 23.11 years, on average. The members of each group could have created a strong shared social identity, which would have eliminated the trade-off between one's own and the group's identity. Then, there would be no need for accommodation to occur.

Secondly, culture might have diminished the participants' perceived need to accommodate. The Dutch culture is characterized by a high degree of individualism and low collectivism (Hofstede, Hofstede, & Minkov, 2010; Vu, Finkenauer, Huizinga, Novin, & Krabbendam, 2017). Individuals of collectivistic cultures focus more on their in- and out-groups, while individuals of individualistic cultures experience group affiliation as less important (Hofstede, 2010). As a result, the participants of this study might have perceived less of a need to demonstrate group affiliation. Then, the participants also would have had no motivation to accommodate to their colleagues. This possible explanation fits the CAT as the CAT assumes the need for group affiliation to be the driving motivator for accommodative behaviors (Giles et al., 1991).

Third, linguistic convergence is merely one potential dimension of accommodation. The CAT applies to a series of dimensions at which people could accommodate to each other, including the pitch of voice, body posture, and speaking rate (Dragojevic et al., 2015). As such, the participants could have accommodated towards each other on a level different than linguistic style. This limitation of the CAT is also noted by one of its grounding fathers: The CAT predicts accommodation itself but fails to predict on what dimension the accommodation is about to occur (Dragojevic et al., 2015). To ensure that all processes of accommodation are captured, a study would have to observe the communication processes on all dimensions known to relate to CAT.

Fourth, the nature of accommodation could be different than assumed. The computational method of the SAM implied that speakers adapt their function word usage on an utterance-to-utterance level. Hence, speakers would have to change their wordings as a direct response to another's utterance. However, accommodation could also occur at a slower rate. Then, the use of a function word might influence not only the directly following utterance but also affect the usage in the following minutes. Not only would the SAM not detect this potential type of accommodation, but it would further suppress the resulting scores, as the usage of the marker words in latter utterances would be noted as an increased baseline of the respondent. Thus, accommodation could have occurred with different structural properties than assumed in this study.

Strengths & Limitations

This study benefited from both its large data set and a new method for computing linguistic accommodation. A large data set of 44,000 utterances was used to investigate accommodation. No other study known to the authors draws on a data set that large in the CAT, workplace, and leadership context. Furthermore, the naturalistic nature of this research adds to the validity of the findings. Besides the quality of the data set, this study benefited from the use of a new unbiased and valid, yet approachable measure. As a result, the accommodation scores computed in this study could be considered more accurate than in other studies in the field.

Despite the large data set and new measure of accommodation, two limitations follow from the possible explanations of the findings at hand. Firstly, only one level of accommodation was observed, leaving the possibility that the participants accommodated on other levels. Hence, no conclusive statement could be made regarding accommodation itself, but merely one facet of it. Due to this, the findings provide conclusive evidence of neither the lack of accommodation nor its presence. Secondly, although the SAM is a valid and accurate measure of linguistic accommodation, its functionality is dependent on the underlying assumption that accommodation occurs within one statement-response set. However, if accommodation were to occur slower than assumed here, the SAM could not serve as a suitable measure and would have to be adapted.

Future Studies & Implications

Future research should address the two limitations previously noted. As such, accommodation should be observed on a broad range of factors related to the CAT to ensure that accommodation does not occur at a facet not observed. Then, a lack of accommodation would yield stronger evidence that there is indeed no accommodation occurring. Furthermore, the SAM should be adapted to also detect lagging effects of accommodation, meaning the influence of the usage of a function word on not only the first following but also the remaining utterances of the respondent. Then, more confident conclusions regarding the exact nature of linguistic accommodation could be drawn.

This study also highlighted some shortcomings of the CAT. The CAT proved useful in a broad range of fields due to its generalizability. However, the general nature of the CAT leads to impractical characteristics when applied in research. As Dragojevic et al. (2015) noted, the CAT successfully predicts when accommodation occurs, but fails to predict on what dimension people accommodate. Moreover, no common understanding prevails in the literature regarding the temporal specifics of accommodative processes (i.e., whether accommodation can be observed within minutes rather than seconds). As such, future studies should investigate whether the exact nature of accommodation can be predicted to provide valuable guidelines in researching accommodation. Here, researchers could focus on the influence of the familiarity between the participants on the dimension of accommodation.

Conclusion

This study investigated the relations of follower-to-follower, follower-to-leader, and leader-to-follower accommodation with team and leader effectiveness in Dutch public sector staff meetings. To that means, a large data set was acquired in a field setting. Moreover, recent developments in the measurements of accommodation were built upon. No relationship between accommodation and effectiveness ratings was found, which could be attributed to a general lack of accommodative behaviors. Several explanations for the lack of accommodation were identified and the need for further research on the nature of accommodation as well as its measurement was established.

References

Ashforth, B. E., & Mael, F. (1989). Social identity theory and the organization. *Academy of management review*, *14*(1), 20-39. doi:10.5465/amr.1989.4278999

Bass, B. M., & Avolio, B. J. (1995). MLQ Multifactor Leadership Questionnaire, Leader Form, Rater Form, and Scoring. California. Palo Alto, CA: Mind Garden

Boot, P., Zijlstra, H., & Geenen, R. (2017). The Dutch translation of the Linguistic Inquiry and Word Count (LIWC) 2007 dictionary. *Dutch Journal of Applied Linguistics*, *6*(1), 65-76.

Cannon-Bowers, J. A., Salas, E., & Converse, S. (1993). Shared mental models in expert team decision making. In N. J. Castellan, Jr. (Ed.), *Individual and group decision making: Current issues* (pp. 221-246). Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

Chen, C. H. (2019). Exploring teacher–student communication in senior-education contexts in Taiwan: A communication accommodation approach. *International Journal of Ageing and Later Life*, 1-47. doi:10.3384/ijal.1652-8670.17369

doi: 10.1075/dujal.6.1.04boo

Chung, C., & Pennebaker, J. (2007). The Psychological Functions of Function Words. In K. Fiedler (Ed.), *Frontiers of social psychology. Social communication* (pp. 343-359). New York, NY, US: Psychology Press.

Danescu-Niculescu-Mizil, C., Gamon, M., & Dumais, S. (2011). Mark my words!: linguistic style accommodation in social media. In *Proceedings of the 20th international conference on World wide web* (pp. 745-754). ACM. doi:10.1145/1963405.1963509

Danescu-Niculescu-Mizil, C., Lee, L., Pang, B., & Kleinberg, J. (2012, April). Echoes of power: Language effects and power differences in social interaction. In *Proceedings of the 21st international conference on World Wide Web* (pp. 699-708). ACM. doi:

10.1145/2187836.2187931

Dragojevic, M., Gasiorek, J., & Giles, H. (2015). Communication accommodation theory. *The international encyclopedia of interpersonal communication*, 1-21. doi:10.1002/9781118540190.wbeic006

Doyle, G., & Frank, M. C. (2016). Investigating the sources of linguistic alignment in conversation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 526-536).

Doyle, G., Yurovsky, D., & Frank, M. C. (2016). A robust framework for estimating linguistic alignment in Twitter conversations. In *Proceedings of the 25th international conference on world wide web* (pp. 637-648). International World Wide Web Conferences Steering Committee. doi: 10.1145/2872427.2883091.

Field, A., Miles, J., & Field, Z. (2012). Discovering statistics using R. London: SAGE Publications Ltd.

Ford, J., & Harding, N. (2018). Followers in leadership theory: Fiction, fantasy and illusion. *Leadership*, *14*(1), 3-24. doi:10.1177/1742715015621372

Fox, J. (1991). Regression diagnostics: An introduction (Vol. 79). Sage.

Gonzales, A. L., Hancock, J. T., & Pennebaker, J. W. (2010). Language style matching as a predictor of social dynamics in small groups. *Communication Research*, *37*(1), 3-19. doi:10.1177/0093650209351468

Gibson, C. B., Cooper, C. D., & Conger, J. A. (2009). Do you see what we see? The complex effects of perceptual distance between leaders and teams. *Journal of Applied Psychology*, *94*(1), 62. doi: 10.1037/a0013073

Giles, H., Coupland, N., & Coupland, J. (1991). Accommodation theory: Communication, context, and consequence. In H. Giles, J. Coupland, & N. Coupland (Eds.), *Studies in emotion and social interaction. Contexts of accommodation: Developments in applied* *sociolinguistics* (pp. 1-68). New York, NY, US: Cambridge University Press; Paris, France: Editions de la Maison des Sciences de l'Homme. doi:CBO9780511663673.001

Guéguen, N. (2013). Mimicry and honesty: People give more honest responses to their mimicker. *International Journal of Psychological Research*, *6*(1), 53-57.

Hofstede, G. (2011). Dimensionalizing cultures: The Hofstede model in context. *Online readings in psychology and culture*, *2*(1), 8. doi:10.9707/2307-0919.1014

Hofstede, G., Hofstede, G. J. & Minkov, M. (2010). Cultures and Organizations: Software of the Mind (Rev. 3rd ed.). New York: McGraw-Hill.

Ireland, M. E., Slatcher, R. B., Eastwick, P. W., Scissors, L. E., Finkel, E. J., & Pennebaker, J. W. (2011). Language style matching predicts relationship initiation and stability. *Psychological science*, *22*(1), 39-44. doi: 10.1177/0956797610392928

Jacob, C., Guéguen, N., Martin, A., & Boulbry, G. (2011). Retail salespeople's mimicry of customers: Effects on consumer behavior. *Journal of Retailing and Consumer Services*, *18*(5), 381-388. doi:10.1016/j.jretconser.2010.11.006

Jones, S., Cotterill, R., Dewdney, N., Muir, K., & Joinson, A. (2014). Finding Zelig in text: A measure for normalising linguistic accommodation. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 455-465).

Liao, W., Bazarova, N. N., & Yuan, Y. C. (2018). Expertise judgment and communication accommodation in linguistic styles in computer-mediated and face-to-face groups. *Communication Research, 45*(8), 1122-1145. doi:10.1177/0093650215626974

Lord, S. P., Sheng, E., Imel, Z. E., Baer, J., & Atkins, D. C. (2015). More than reflections: Empathy in motivational interviewing includes language style synchrony between therapist and client. *Behavior Therapy, 46*(3), 296-303.

doi:10.1016/j.beth.2014.11.002

Meinecke, A. L., & Kauffeld, S. (2018). Engaging the hearts and minds of followers: Leader empathy and Language style matching during appraisal interviews. *Journal of Business and Psychology*, 1-17. doi:10.1007/s10869-018-9554-9

Miao, Q., Newman, A., & Huang, X. (2014). The impact of participative leadership on job performance and organizational citizenship behavior: Distinguishing between the mediating effects of affective and cognitive trust. *The International Journal of Human Resource Management*, *25*(20), 2796-2810. doi:10.1080/09585192.2014.934890

Muir, K., Joinson, A., Cotterill, R., & Dewdney, N. (2016). Characterizing the linguistic chameleon: Personal and social correlates of linguistic style accommodation. *Human Communication Research*, *42*(3), 462-484. doi:10.1111/hcre.12083

Mumford, M. D., Todd, E. M., Higgs, C., & McIntosh, T. (2017). Cognitive skills and leadership performance: The nine critical skills. *The Leadership Quarterly*, *28*(1), 24-39. doi:10.1016/j.leaqua.2016.10.012

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*. Retrieved from: https://repositories.lib.utexas.edu/handle/2152/31333

Pickering, M. J., & Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and brain sciences*, *27*(2), 169-190. doi:10.1017/S0140525X04000056

R Core Team (2018). R: A language and environment for statistical computing. *R Foundation for Statistical Computin*g, Vienna, Austria. Retrieved from https://www.R-project.org/

Romero, D. M., Swaab, R. I., Uzzi, B., & Galinsky, A. D. (2015). Mimicry is presidential: Linguistic style matching in presidential debates and improved polling numbers. *Personality and Social Psychology Bulletin*, *41*(10), 1311-1319. doi:10.1177/0146167215591168

Scissors, L. E., Gill, A. J., & Gergle, D. (2008). Linguistic mimicry and trust in text-based CMC. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (pp. 277-280). ACM. doi:10.1145/1460563.1460608

Sijtsma, K. (2009). On the use, the misuse, and the very limited usefulness of Cronbach's alpha. *Psychometrika*, *74*(1), 107. doi: 10.1007/s11336-008-9101-0

Soliz, J., & Giles, H. (2014). Relational and Identity Processes in Communication: A Contextual and Meta-Analytical Review of Communication Accommodation Theory. Annals of the International Communication Association, 38(1), 107-144. doi:10.1080/23808985.2014.11679160

Steinmann, B., Dörr, S. L., Schultheiss, O. C., & Maier, G. W. (2015). Implicit motives and leadership performance revisited: What constitutes the leadership motive pattern?. *Motivation and Emotion*, *39*(2), 167-174. doi:10.1007/s11031-014-9458-6

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC And computerized text analysis methods. *Journal of language and social psychology*, *29*(1), 24-54. doi:10.1177/0261927X09351676

Taylor, P. J., & Thomas, S. (2008). Linguistic style matching and negotiation outcome. *Negotiation and Conflict Management Research*, *1*(3), 263-281. doi:10.1111/j.1750-4716.2008.00016.x

Vu, T. V., Finkenauer, C., Huizinga, M., Novin, S., & Krabbendam, L. (2017). Do Individualism and collectivism on three levels (country, individual, and situation) influence theory-of-mind efficiency? A cross-country study. *PloS one*, *12*(8), e0183011. doi:10.1371/journal.pone.0183011

Yilmaz, G. (2016). What you do and how you speak matter: Behavioral and linguistic determinants of performance in virtual teams. *Journal of Language and Social Psychology*, *35*(1), 76-97. doi:10.1177/0261927X15575772

Zaccaro, S. J., Rittman, A. L., & Marks, M. A. (2001). Team leadership. *The leadership quarterly*, *12*(4), 451-483. doi:10.1016/S1048-9843(01)00093-5

Appendix A

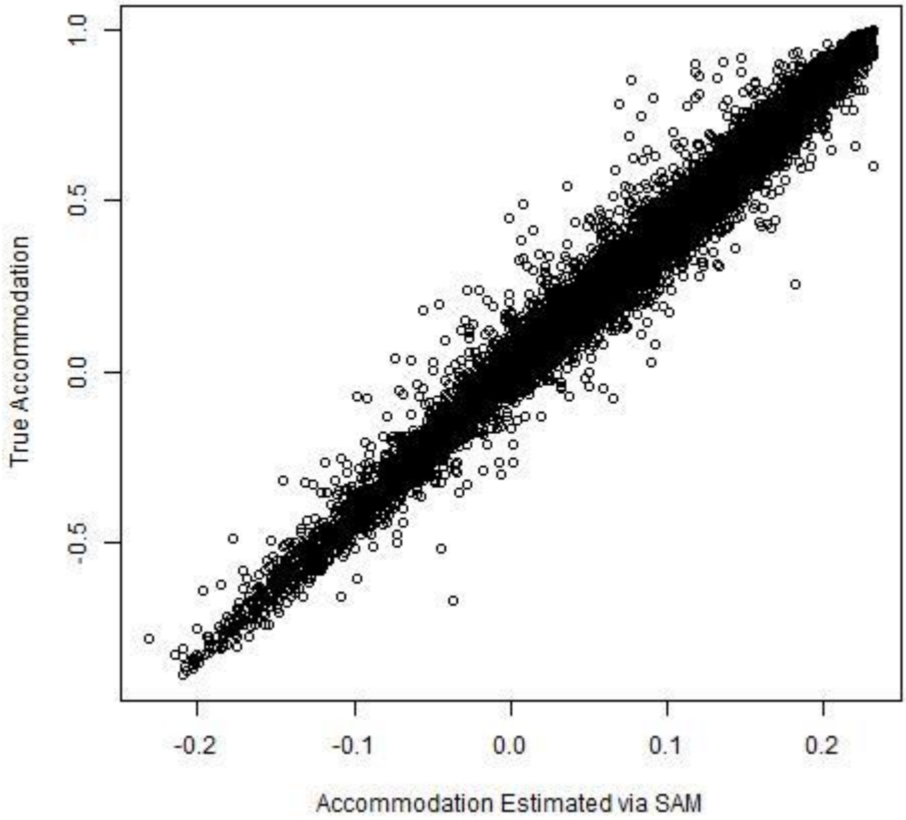

Figure 1.

True vs Estimated (SAM) Accommodation.

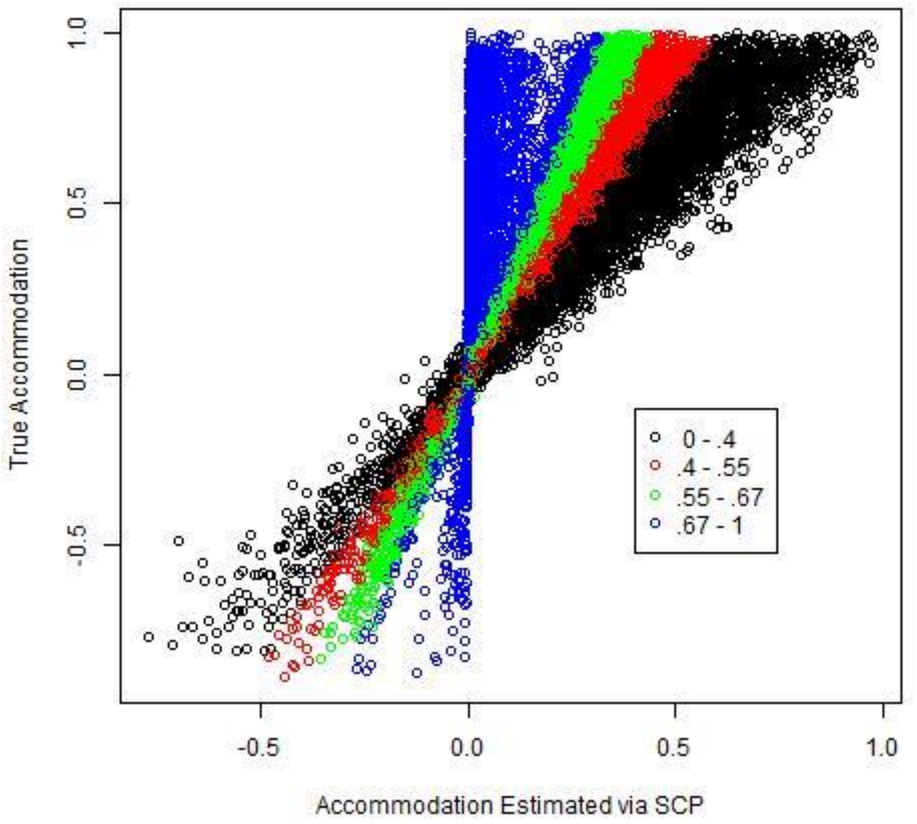

Figure 2.

True vs Estimated (SCP) Accommodation.

Appendix B

/R/helpers.R

```
library(textreadr)
library(dplyr)

get_team <- function(x){
  # Simple function that parses file name for the team identification. Assumes that pilots have
word
  # 'Pilot' in them.

  pilot <- grepl("Pilot", x)

  number <- gsub("[^0-9]", "", x)
  if(pilot){
    id <- paste("P", number, sep = "")
  } else id <- number

  return(id)
}

get_identifiers <- function(x, group_followers = T, group_all = F) {
  # Scans a vectors of documents for identifiers.
  # Arguments:
  # x -- Vector containing documents as a character string.
  # group_followers -- Boolean indicating whether specific follower IDs should be saved
(FALSE).
  # group_all -- Boolean. If true, a logical vector indicating whether an identifier exists or not
is returned.
  #
  # Returns a vector if group_followers = T, returns a list of two vectors otherwise, where
  # in role -- 0 ~ not detected, 1 ~ leader, 2 ~ follower and
  # in id -- 0 ~ leader speaking, 9999 ~ not detected, NA ~ no role detected.
  # The later ensures that just id is all infomation needed to identify a person within a team.

  #' Let's talk Regex!
  #'
  #' Our regular expression to identify leaders and followers is (with some adaptions):
  #'
  #' "^[[:space:]]{0,2}L.{0,11}:|^[[:space:]]{0,2}F.{0,11}:"
  #'
  #' ... which means ...
  #'
  #'                        Indicate a hit when...
  #' ^                       ...at the beginning of the string...
  #' [[:space:]]{0,2}          ...there are 0-2 spaces...
  #' L                        ...followed by the letter 'L'...
  #' .{0,11}                    and then anything 0 - 11 times...
  #' :                      and a colon...
```

```
  #' |                        OR
  #' ^                         ...at the beginning of the string...
  #' [[:space:]]{0,2}            ...there are 0-2 spaces...
  #' F                          ...followed by the letter 'F'...
  #' .{0,11}                      and then anything 0 - 11 times...
  #' :                         and a colon...

  if(group_all==TRUE){
    return(grepl("^[[:space:]]{0,2}L.{0,11}:|^[[:space:]]{0,2}F.{0,11}:", x))
  }

  role <- vector(length = length(x))
  id <- vector(length = length(x))
  for(u in 1:length(x)){
    if(grepl("^[[:space:]]{0,2}L.{0,11}:", x[u])){
      role[u] <- 1
      id[u] <- 0
    } else if(grepl("^[[:space:]]{0,2}F.{0,11}:", x[u])){
      role[u] <- 2

      if(group_followers == FALSE){
        # Delete everything that is not a number.
        identifier <- gsub("[^0-9]", " ", x[u])
        identifier <- unlist(strsplit(identifier, " "))
        identifier <- identifier[nchar(identifier) > 0]
        if(length(identifier) >= 1) id[u] <-as.numeric(identifier[1]) else id[u] <- 9999

      }
    } else {
      role[u] <- NA
      id[u] <- 9999
    }
  }

  if(group_followers == T) {
    return(role)
  } else{
    out_list <- list()
    out_list$role <- role
    out_list$id <- id
    return(out_list)
  }
}

merge_non_identified_utterances <- function(x) {
  # Merges utterance wihtout identifier with last utterance.

  with_identifier <- which(get_identifiers(x, group_all = TRUE))

  without_identifier <- which(!get_identifiers(x, group_all = TRUE))
```

```
  for (u in without_identifier) {
    # Find last utterance with identifier
    last_id <- max(with_identifier[with_identifier < u])

    if(last_id == TRUE) {
      x[last_id] <- paste(x[last_id], x[u])
    }

    # If there is no id before, just delete the line (-> Headers)
    x[u] <- NA
  }

  x <- na.omit(x)

  return(x)
}

delete_brackets <- function(x, br_type = c(1, 2, 3, 4)) {

  brackets <- list(c("\\(", "\\)"), c("<", ">"), c("\\{", "\\}"), c("\\[", "\\]"))
  brackets <- brackets[br_type]

  for(br in 1:length(brackets)){
    # For every utterance...
    for (u in 1:length(x)) {
      if(grepl(paste(brackets[[br]][1], "|", brackets[[br]][2], sep=""), x[u])){
        # Get all brackets.
        bracket_split <- unlist(base::strsplit(x[u], ""))
        open_index <- grep(brackets[[br]][1], bracket_split)
        closed_index <- grep(brackets[[br]][2], bracket_split)

        if(length(open_index) != length(closed_index)) stop(
          paste("Error in delete_brackets(). Number of opening and closing brackets does not
match!", "Text:", x[u])
                )

        # For each pair of brackets...
        # The sequence is reversed to not influence the remaining indecies.
        for (i in length(open_index):1) {
          bracket_split <- bracket_split[-(open_index[i]:closed_index[i])]

        }

        bracket_split <- paste(bracket_split, collapse = "")
        x[u] <- bracket_split

      }
    }
  }
```

```
  return(x)
}

delete_empty_utt <- function(utt, ids, roles){
  for (u in 1:length(utt)) {
    if(!grepl("[a-z]", utt[u])) {
      utt[u] <- NA
      ids[u] <- NA
      roles[u] <- NA
    }
  }
  utt <- utt[!is.na(utt)]
  ids <- ids[!is.na(ids)]
  roles <- roles[!is.na(roles)]

  return(list(utterances = utt, ids = ids, roles = roles))
}

clean_doc <- function(x, ids = NA, roles = NA, operations = c(1, 2, 3, 4, 5, 6)) {
  # A catch all function performing simple transformations on the text.
  # Arguments:
  # x -- Vector containing utterances as character string.
  # operations -- Vector indicating operations to be taken, where
  # 1 ~ delete all digits, 2 ~ delete all punctuation, 3 ~ make all text lower case, 4 ~ delete
identifiers, 5 ~ delete all brackets

  if(5 %in% operations) x <- delete_brackets(x) # moved to top to make errors more
informative
  if(4 %in% operations) x <- gsub("^[[:space:]]{0,2}L.{0,11}:|^[[:space:]]{0,2}F.{0,11}:",
"", x)
  if(1 %in% operations) x <- gsub("[0-9]+", "", x)
  if(2 %in% operations) x <- gsub("(*UCP)(*UTF)[[:punct:]]", "", x, perl = T) # Added the
encodings to delete some punct that is encoded specially by Word.
  if(3 %in% operations) x <- tolower(x)
  if(6 %in% operations){
    temp <- delete_empty_utt(utt = x, ids = ids, roles = roles)
    x <- temp$utterances
    ids <- temp$ids
    roles <- temp$roles
  }
  if(7 %in% operations) {
    temp_2 <- merge_double_speakers(x, ids, roles)
    x <- temp_2$utterances
    ids <- temp_2$ids
    roles <- temp_2$roles
  }

  return(list(utterances = x, ids = ids, roles = roles))
}
```

```
merge_double_speakers <- function(utterances, ids, roles) {
  # Merges two or more successive utterances if spoken by same person. This normally
occurs when a person in between
  # speaks inaudibly, leading to a deletion of the intermediary utterance.
  #
  # This function should be called as late as possible in the data wrangling to ensure all
deletions were already
  # executed.

  for (id in length(ids):2) {

    if(ids[id] == ids[id-1]){
      utterances[id-1] <- paste(utterances[id-1], utterances[id])
      utterances[id] <- NA

      ids[id] <- NA
      roles[id] <- NA


    }

  }
  utterances <- utterances[!is.na(utterances)]
  ids <- ids[!is.na(ids)]
  roles <- roles[!is.na(roles)]

  return(list(ids=ids, utterances=utterances, roles = roles))
}



doc_as_table <- function(utterances, roles, ids, team, tab = 0, df=NA) {
  # Converts vectors of utterances, roles, and ids to a table.
  # tab -- If zero, a table will be created. Otherwise, an already existing table can be inserted
to append new data.

  if(tab == 0) df <- data.frame(team=NA, roles=NA, ids=NA, SeqID=NA, utterances=NA)

  # Input Validation
  if(length(utterances) != length(roles) | length(roles) != length(ids)) stop("Input vectors need
to be of same length!")

  names <- c("Team", "Role", "ID", "SeqID", "Utterance")
  for (u in 1:length(utterances)) {
    row <- c(team, roles[u], ids[u], u, utterances[u])
    names(row) <- names
    df <- rbind(df, row)

  }
```

```
  return(df)
}


DNM_coordination <- function(b, a) {
  # Calculates the linguistic coordination metric according to Danescu-Niculescu-Mizil et al.
(2012)
  #
  # Arguments:
  # a, b -- Vectors containing binary (!) descriptions of the utterances of a and b.
  #        This function will assess to what degree b coordinated towards a. Hence,
  #        the order of inputs influence the results. Also the assumption that a
  #        starts speaking first follows. Accordingly, the data have to be
  #        pre-processed, ideally with the DNM_wrapper() function.

  if(length(a) != length(b)) stop("Input vectors need to be of same length!")
  a <- as.numeric(a > 0)
  b <- as.numeric(b > 0)

  # ou = original utterance, m = marker
  m_in_ou <- sum(a) / length(a)
  m_in_reply <- sum(b) / length(b)
  m_in_both <-sum(a == 1 & b == 1) / length(a)

  coord <- m_in_both / m_in_ou - m_in_reply

  return(list(LSM = coord, limit = m_in_ou))

}



DNM_wrapper <- function(x_df, a_id, b_id, team, type = "regular", cols = c("Team", "Role",
"ID", "SeqID", "Utterance", "WC", "article", "certain", "conj",
                                                     "discrep", "excl", "incl", "ipron", "negate",
"preps", "quant", "tentat", "i",
                                                     "we", "you")) {
  # Transforms a doc_df into a data frame suitable for analysing linguistic
  # coordination with the DNM_coordiation() and mDNM_coordination() functions.
  #
  # Arguments:
  # x_df      --  Data frame as created by doc_as_table()
  # a_id, b_id--  A vector of identifying strings in the format of
  #              c("Role", "Id"). If id == NA, only roles will be considered.
  # type      --  A character string indicating what DNM function will be used
  #              later. Here: "regular" ~ DNM_coordination(),
  #              "modified" ~ mDNM_coordination().

  # Select relevant team
```

```
  x_df <- filter(x_df, Team == team)

  # Select relevant columns
  x_df <- select(x_df, cols)

  # Make LIWC entries binary for regular DNM coordination.
  if(type == "regular") {
    x_df[,7:ncol(x_df)] <- as.numeric(x_df[,7:ncol(x_df)] > 0)
  }


  # Getting utterances of a
  if(is.na(a_id[2])){
    a_utt <- filter(x_df, Role == a_id[1])
  } else if(!is.na(a_id[2])) {
    a_utt <- filter(x_df, Role == a_id[1] & ID == a_id[2])
  }

  # Getting utterances that followed a. This way, we delete irrelevant utterances.
  # However, these 'irrelevant' utterances might still be usable for a determination of the
baseline.
  seq_ids <- a_utt$SeqID + 1
  b_utt <- filter(x_df, SeqID %in% seq_ids)

  if(is.na(b_id[2])){
    b_utt <- filter(b_utt, Role == b_id[1])
  } else if(!is.na(b_id[2])) {
    b_utt <- filter(b_utt, Role == b_id[1] & ID == b_id[2])
  }

  # Deleting all entries from a_utt where b is not the next speaker.
  seq_ids <- b_utt$SeqID - 1
  a_utt <- filter(a_utt, SeqID %in% seq_ids)

  return(list(a=a_utt, b=b_utt))
}

absolut <- function(percent, word_count) {
  return(round(percent * word_count / 100))
}

SAM <- function(x) {
  #' Accepts one data.frame with columns A, B, and WC, where A is the speaker,
  #' B is the person aligning, and WC is the word count of utterance B.
  #'
  #' Each row represents a dyad of utterances.
  #'
  #' Prevalence of m in utterances is represented binary (0 ~ not present,
  #' 1 ~ present) in A and in count form in B. Column WC represent the
  #' word count in the given utterance of B.
```

```
  #'
  #' Update: The input of A can now also be count data.
  #'

  N_align <- filter(x, A > 0)
  N_base <- filter(x, A == 0)

  C_align <- sum(N_align$B) / sum(N_align$B_WC)
  C_base <- sum(N_base$B) / sum(N_base$B_WC)

  return(list(Align = logit_inv(C_align) - logit_inv(C_base), C_align = C_align, C_base =
C_base))

}

SAM_wrapper <- function(x_df, a_id, b_id, team, cats =c("article", "certain", "conj",
                                             "discrep", "excl", "incl", "ipron", "negate", "preps",
"quant", "tentat", "i",
                                             "we", "you")){
  #' A wrapper for SAM().
  #'
  #' Arguments:
  #' x_df       -- Data frame as doc_table(). Commonly referred to as doc_df.
  #' a_id, b_id -- A vector of identifying strings in the format of
  #             c('Role', 'Id'). If id == NA, only roles will be considered
  #' team       -- String indicating team number. Has to follow convenctions of doc_df.
  #' cats       -- Vector indicating what categories should be used for alignment.
  #'
  #' Returns:
  #'
  #'


  # Select relevant team
  x_df <- filter(x_df, Team == team)

  # Getting utterances of a
  if(is.na(a_id[2])){
    a_utt <- filter(x_df, Role == a_id[1])
  } else if(!is.na(a_id[2])) {
    a_utt <- filter(x_df, Role == a_id[1] & ID == a_id[2])
  }


  # Getting utterances that followed a. This way, we delete irrelevant utterances.
  # However, these 'irrelevant' utterances might still be usable for a determination of the
baseline.
  seq_ids <- a_utt$SeqID + 1

  if(is.na(b_id[2])){
```

```
    b_utt <- filter(x_df, SeqID %in% seq_ids & Role == b_id[1])
  } else if(!is.na(b_id[2])) {
    b_utt <- filter(x_df, SeqID %in% seq_ids & Role == b_id[1] & ID == b_id[2])
  }

  # Deleting all entries from a_utt where b is not the next speaker.
  seq_ids <- b_utt$SeqID - 1
  a_utt <- filter(a_utt, SeqID %in% seq_ids)

  # If alignment scores have to be computed for several categories,
  # the SAM() function has to be called several times with different
  # arguments. Results are stored in a vector.
  results <- vector(mode = "numeric", length = length(cats)+1)

  for(c in 1:length(cats)){

    # Construct a data.frame
    temp_df <- data.frame("A" = a_utt[,cats[c]],
                          "B" = b_utt[,cats[c]],
                          "B_WC" = b_utt[,"WC"])


    # Transform data
    temp_df$B <- absolut(temp_df$B, temp_df$B_WC)

    # Run analysis
    results[c] <- SAM(temp_df)$Align


  }

  # Find the number of statement-response pairs
  results[length(results)] <- nrow(a_utt)

  names(results) <- c(cats, "n_utt")
  return(results)
}

logit <- function(p) {
  return(log(p / (1 - p), exp(1)))
}

logit_inv <- function(p){
  return(exp(p) / (exp(p) + 1))
}

SAM_sim <- function(A_base = 0.05, B_base = 0.05, coord = 0.3, N_utt = 1000,
N_len_decay = 0.05, WC = NA, A_WC = NA) {
  SAM_df <- data.frame(A = vector(length = N_utt), B = vector(length = N_utt), B_WC =
vector(length = N_utt), A_WC = vector(length = N_utt))
```

```
  if(is.na(WC)) WC <- ceiling(rexp(N_utt, N_len_decay))
  if(is.na(A_WC)) A_WC <- ceiling(rexp(N_utt, N_len_decay))

  A <- vector(length = N_utt)

  for(i in 1:N_utt){
    A[i] <- rbinom(1, A_WC[i], A_base)
  }


  B <- vector(length = N_utt)

  for (i in 1:N_utt) {
    if(A[i] > 0){
      B[i] <- rbinom(1, 1, B_base + coord)
    } else {
      B[i] <- rbinom(1, 1, B_base)
    }
  }

  B <- B * WC

  SAM_df$A <- A
  SAM_df$B <- B
  SAM_df$B_WC <- WC
  SAM_df$A_WC <- A_WC

  return(SAM_df)

}


complete_case_wrapper <- function(data, desiredCols) {
    # Copied from BenBarnes at https://stackoverflow.com/a/11258247
    completeVec <- complete.cases(data[, desiredCols])
    return(data[completeVec, ])
}


LSM <- function(A, B){
  #' Calculates LSM as presented in Gonzales,
  #' Hancock, & Pennebaker, 2010.
  #'
  #' Arguments:
  #' A, B ~ Data frames giving the LIWC output for A and B per
  #'        utterance. Numbers are given as percentages.
  #'
```

```
  if((FALSE == is.data.frame(A)) | (FALSE == is.data.frame(B))) {
    stop("Input must be a data frame!")
  }
  if(ncol(A) != ncol(B)) stop("Data frame must have same width!")

  results <- vector(mode="numeric", length = ncol(A))
  for (c in 1:ncol(A)) {
    pA <- mean(A[,i], na.rm = T)
    pB <- mean(B[,i], na.rm = T)

    results[i] <- 1 - (abs(pA - pB) / (pA + pB))
  }

  return(mean(results, na.rm = T))
}
```

/docs/creating_doc_df.R

```
setwd("C:/Users/jakob/Documents/Text Mining/transcripts/Trans_Checked")


source("C:/Users/jakob/Documents/Text Mining/helpers.R")

available_files <- list.files()
doc_df <- data.frame(team=NA, roles=NA, ids=NA, SeqID=NA, utterances=NA)


for (i in 1:length(available_files)) {
  print(i)
  document <- read_docx(available_files[i])
  team_id <- get_team(available_files[i])
  print(team_id)

  document <- merge_non_identified_utterances(document)

  identifiers <- get_identifiers(document, group_followers = F)
  roles <- identifiers$role
  ids <- identifiers$id

  cleaned <- clean_doc(document, ids = ids, roles = roles)
  utterances <- cleaned$utterances
  ids <- cleaned$ids
  roles <- cleaned$roles

  doc_df <- doc_as_table(utterances = utterances, team = team_id, roles = roles, ids = ids,
tab=1, df = doc_df)
}
```

```
doc_df <- doc_df[2:nrow(doc_df),]

write.csv(doc_df, "C:/Users/jakob/Documents/Text Mining/doc_df.csv")
```

/docs/spss_prep.Rmd

```
---
title: "Preparing the SPSS-Survey File"
author: "Jakob Buske"
date: "23 Mai 2019"
output: pdf_document
---



```{r warning=F}
library(haven)
library(dplyr)
source("./R/helpers.R")

spss_file <- read_sav(file.choose())
```


Rearranging the Team-Numbers

The team numbers in the sav file were named in a non-conventional way.

Numbers below 15 are Pilots (e.g. "1" -> "Pilot 1"). All other team numbers have a one
added to the front of their actual ID (e.g. "121" -> "21").

```{r}
for (r in 1:nrow(spss_file)) {
  if(is.na(spss_file$TeamNumber[r])) next
  if(as.numeric(spss_file$TeamNumber[r]) < 15){
    spss_file$TeamNumber[r] <- paste("P", as.character(spss_file$TeamNumber[r]), sep = "")
  } else if(as.numeric(spss_file$TeamNumber[r]) < 99){
    next
  } else{
    spss_file$TeamNumber[r] <-
as.character(as.numeric(substring(as.character(spss_file$TeamNumber[r]), 2)))
  }
}

length(levels(as.factor(spss_file$TeamNumber)))
```

Deleting teams that are not in doc_df
```

There may not be transcripts for every team due to bad audio quality. To delete the teams whose meetings were not transcribed, doc_df is read and checked for available team numbers.

```{r}
doc_df <- read.csv("./data_gen/doc_df.csv", stringsAsFactors = F, dec = ",")
doc_df <- doc_df[2:nrow(doc_df),2:ncol(doc_df)]
names(doc_df)[1:5] <- c("Team", "Role", "ID", "SeqID", "Utterance")

for(i in c(2, 3, 4)){
  doc_df[,i] <- as.numeric(doc_df[,i])
}
```

Finding the teams that are in the SPSS file but not in the transcipts.

```{r}
spss_del <-
levels(as.factor(spss_file$TeamNumber))[!(levels(as.factor(spss_file$TeamNumber)) %in%
levels(as.factor(doc_df$Team)))]
```

```{r}
print(paste("Teams in SPSS file:",
as.character(length(levels(as.factor(spss_file$TeamNumber))))))
print(paste("Files to be deleted due to lack of transcript:", as.character(length(spss_del))))
print(paste("Remaining files:", length(levels(as.factor(spss_file$TeamNumber))) -
length(spss_del)))
```

Deleting all rows related to teams that ought to be deleted.

```{r}
for (r in 1:nrow(spss_file)) {
  if(spss_file$TeamNumber[r] %in% as.character(spss_del)) spss_file[r,] <- rep(NA,
ncol(spss_file))
}
```

It turns out that for team 31, two persons are coded as a leader. The comments indicate that one of them is the real leader. Accordingly, the other one is re-coded as a follower.

```{r}
spss_file[which(spss_file$ID == 231), "LeaderFollower"] <- 2
```

To make the data set more managable, only relevant columns are selected. Also, observations without a TeamNumber are deleted.

Relevant columns:
```{r}
names(spss_file)[c(2, 9, 11, 76, 77, 78, 86, 87, 88, 89, 122, 123, 124, 125)]
```

```{r}
spss_small <- spss_file %>%
  select(c(2, 9, 11, 76, 77, 78, 86, 87, 88, 89, 122, 123, 124, 125)) %>%
  filter(! is.na(TeamNumber))

head(spss_small)
```

Lastly, the team and leader effectiveness scores are aggregated for the analysis.
```{r}
spss_small <- cbind(spss_small, rowMeans(spss_small[,c("LeEff1",
                                                        "LeEff2",
                                                        "LeEff3",
                                                        "LeEff4")], na.rm = T))
names(spss_small)[ncol(spss_small)] <- "LeEff"

spss_small <- cbind(spss_small, rowMeans(spss_small[,c("TeamEff1",
                                                        "TeamEff2",
                                                        "TeamEff3",
                                                        "TeamEff4")], na.rm = T))

names(spss_small)[ncol(spss_small)] <- "TeamEff"

```

```{r}
write.csv(spss_small, "./data_gen/survey.csv", row.names = F)
```

/docs/accommodation.Rmd

```
---
title: "Accommodation Scores"
author: "Jakob Buske"
date: "10 Juni 2019"
output: pdf_document
---
```

In this document, the accommodation scores for the whole data set are computed. Since we are not only interested in the data needed for the actual analysis but also background information, some variations of the accommodation scores are computed. Specifically, the following data sets are generated:

* Follower-Follower Accommodation (via SAM, H1A & LSM, exploratory)

* Leader-Follower Accommodation (via SAM, H1B & LSM, expl.)

* Follower-Leader Accommodation (via SAM, H1C & LSM, expl.)

* Person-Person Accommodation (via SAM, DNM & LSM, all expl.)

For the last computation, all scores are generated on an dyadic level (i.e. one-to-one). The remaining scores are group calculations, where the accommodation of one person towards the remaining group is calculated (i.e. one-to-many).

```{r warning=FALSE}
source("./R/helpers.R")
library(dplyr)
doc_df <- read.csv("./data_gen/doc_df.csv", stringsAsFactors = F, dec = ",")
doc_df <- doc_df[2:nrow(doc_df),2:ncol(doc_df)]
names(doc_df)[1:5] <- c("Team", "Role", "ID", "SeqID", "Utterance")

for(i in c(2, 3, 4)){
  doc_df[,i] <- as.numeric(doc_df[,i])
}

# Utterances with unknown speakers are filtered out.
doc_df <- filter(doc_df, ID < 99)
```

Implementation of SAM

The SAM is implemented in SAM().

```{r}
SAM
```

As the data from the doc_df have to be transformed before the SAM() can computed scores, a wrapper is used.

```{r}
SAM_wrapper
```

Follower-Follower Accommodation

SAM

```{r}
f_f_acc <- as.data.frame(matrix(ncol = 16))
names(f_f_acc) <- c("Team", "article", "certain", "conj", "discrep",
                  "excl", "incl", "ipron", "negate", "preps", "quant",
                  "tentat", "i", "we", "you", "n_utt")

# For each team...
teams <- unique(doc_df$Team)
for (t in 1:length(teams)) {
  doc_df_t <- dplyr::filter(doc_df, Team == teams[t])


  SAM_temp <- SAM_wrapper(x_df = doc_df_t, a_id = c("2",  NA),
                        b_id = c("2", NA), team = teams[t])
  temp <- c(teams[t],  SAM_temp)
  names(temp) <-  c("Team", "article", "certain", "conj", "discrep",
                  "excl", "incl", "ipron", "negate", "preps", "quant",
                  "tentat", "i", "we", "you", "n_utt")
  f_f_acc <- rbind(f_f_acc, temp)

}



# Transforming columns to numeric again
for (c in 2:16) {
  f_f_acc[,c] <- as.numeric(f_f_acc[,c])
}

# Attaching the rowMeans to the data frame as aggregate scores. NAs are ignored.
f_f_acc <- cbind(f_f_acc, rowMeans(f_f_acc[,2:15], na.rm = T))
names(f_f_acc)[ncol(f_f_acc)] <- "Acc"

# Deleting the first row of NAs
f_f_acc <- f_f_acc[2:nrow(f_f_acc),]

head(f_f_acc)
```
LSM

LSM cannot be computed on a group level in a meaningful way. Follower-Follower LSM can be inspected in the section regarding Person-Person accommodation.

Leader-Follower Accommodation

In this section, the degree to which the followers accommodate towards their leader is calculated. The process is equal to the process of the previous section. Again, the group is already being aggregated in the analysis.

Attention: As of now, sometimes two utterances directly following each other are from the same person! Fix this!

SAM

```{r}
# Creating an empty data frame to store the data in.
l_f_acc <- as.data.frame(matrix(ncol = 16))
names(l_f_acc) <- c("Team", "article", "certain", "conj", "discrep",
                    "excl", "incl", "ipron", "negate", "preps", "quant",
                    "tentat", "i", "we", "you", "n_utt")

# For each team...
teams <- unique(doc_df$Team)
for (t in 1:length(teams)) {
  doc_df_t <- dplyr::filter(doc_df, Team == teams[t])


  SAM_temp <- SAM_wrapper(x_df = doc_df_t, a_id = c("1",  NA),
                          b_id = c("2", NA), team = teams[t])
  temp <- c(teams[t], SAM_temp)
  names(temp) <-  c("Team", "article", "certain", "conj", "discrep",
                    "excl", "incl", "ipron", "negate", "preps", "quant",
                    "tentat", "i", "we", "you", "n_utt")

  l_f_acc <- rbind(l_f_acc, temp)


}

# Transforming columns to numeric again
for(c in 2:16){
  l_f_acc[,c] <- as.numeric(l_f_acc[,c])
}

# Attaching the rowMeans to the data frame as aggregate scores. NAs are ignored.
l_f_acc <- cbind(l_f_acc, rowMeans(l_f_acc[,2:15], na.rm = T))
names(l_f_acc)[ncol(l_f_acc)] <- "Acc"

# Deleting first row of NAs
l_f_acc <- l_f_acc[2:nrow(l_f_acc),]

head(l_f_acc)
```

LSM

LSM is not directional. Leader-Follower scores are computed in the Follower-Leader section.

Follower-Leader Accommodation

SAM

```{r}
# Creating an empty data frame to store the data in.
f_l_acc <- as.data.frame(matrix(ncol = 16))
names(f_l_acc) <- c("Team", "article", "certain", "conj", "discrep", "excl", "incl", "ipron",
"negate", "preps", "quant", "tentat", "i", "we", "you", "n_utt")

# For each team...
teams <- unique(doc_df$Team)
for (t in 1:length(teams)) {
  doc_df_t <- dplyr::filter(doc_df, Team == teams[t])


  SAM_temp <- SAM_wrapper(x_df = doc_df_t, a_id = c("2",  NA), b_id = c("1", NA), team
= teams[t])
  temp <- c(teams[t], SAM_temp)
  names(temp) <-  c("Team", "article", "certain", "conj", "discrep", "excl", "incl", "ipron",
"negate", "preps", "quant", "tentat", "i", "we", "you", "n_utt")

  f_l_acc <- rbind(f_l_acc, temp)


}

# Transforming columns to numeric again
for(c in 2:16){
  f_l_acc[,c] <- as.numeric(f_l_acc[,c])
}

# Attaching the rowMeans to the data frame as aggregate scores. NAs are ignored.
f_l_acc <- cbind(f_l_acc, rowMeans(f_l_acc[,2:15], na.rm = T))
names(f_l_acc)[ncol(f_l_acc)] <- "Acc"

# Deleting first row of NAs
f_l_acc <- f_l_acc[2:nrow(f_l_acc),]

head(f_l_acc)
```

LSM

```{r}
f_l_LSM <- as.data.frame(matrix(ncol = 2))
names(f_l_LSM) <- c("Team", "LSM")
```

```
teams <- unique(doc_df$Team)
for(t in 1:length(teams)){
  A_temp <- doc_df %>%
      filter(Team == teams[t] & Role == 1) %>%
      select(c("article", "certain", "conj", "discrep", "excl", "incl",
               "ipron", "negate", "preps", "quant", "tentat", "i", "we",
               "you"))

  B_temp <- doc_df %>%
      filter(Team == teams[t] & Role == 2) %>%
      select(c("article", "certain", "conj", "discrep", "excl", "incl",
               "ipron", "negate", "preps", "quant", "tentat", "i", "we",
               "you"))

  temp <- c(teams[t], LSM(A_temp, B_temp))
  names(temp) <- c("Team", "LSM")
  f_l_LSM <- rbind(f_l_LSM, temp)
}

f_l_LSM[,2] <- as.numeric(f_l_LSM[,2])

head(f_l_LSM)
```

Person-Person Accommodation

Person-Person Accommodation via SAM

```{r}
P_P_acc <-  as.data.frame(matrix(ncol = 18))
names(P_P_acc) <- c("Team", "ID_A", "ID_B", "article", "certain", "conj", "discrep",
                    "excl", "incl", "ipron", "negate", "preps", "quant", "tentat",
                    "i", "we", "you", "n_utt")

# For each team...
teams <- unique(doc_df$Team)

for (t in 1:length(teams)) {
  doc_df_t <- dplyr::filter(doc_df, Team == teams[t])

  for (f1 in 1:length(unique(doc_df_t$ID))) {
    for(f2 in 1:length(unique(doc_df_t$ID))){

      # Generating id-vectors for both speakers
      if(unique(doc_df_t$ID)[f1] == 0){
        a_id_temp <- c("1", unique(doc_df_t$ID)[f1])
      } else{
        a_id_temp <- c("2", unique(doc_df_t$ID)[f1])
```

```
      }


      if(unique(doc_df_t$ID)[f2] == 0){
        b_id_temp <- c("1", unique(doc_df_t$ID)[f2])
      } else{
        b_id_temp <- c("2", unique(doc_df_t$ID)[f2])
      }

      SAM_temp <- SAM_wrapper(x_df = doc_df_t, a_id = a_id_temp, b_id = b_id_temp,
team = teams[t])
      temp <- c(teams[t], unique(doc_df_t$ID)[f1], unique(doc_df_t$ID)[f2], SAM_temp)
      names(temp) <-  c("Team", "ID_A", "ID_B", "article", "certain", "conj", "discrep",
"excl",
                    "incl", "ipron", "negate", "preps", "quant", "tentat", "i", "we", "you",
                    "n_utt")

      P_P_acc <- rbind(P_P_acc, temp)



    }

  }

}

# Transforming the columns to mode numeric again.
for(c in 2:18){
  P_P_acc[,c] <- as.numeric(P_P_acc[,c])
}

# Attaching the rowMeans to the data frame as aggregate scores. NAs are ignored.
P_P_acc <- cbind(P_P_acc, rowMeans(P_P_acc[,4:17], na.rm = T))
names(P_P_acc)[ncol(P_P_acc)] <- "Acc"

P_P_acc <- filter(P_P_acc, n_utt > 1)

head(P_P_acc)
```

Person-Person Accommodation via DNM

```{r}
P_P_DNM <- as.data.frame(matrix(ncol = 17))
names(P_P_DNM) <- c("Team", "ID_A", "ID_B", "article", "certain", "conj",
                    "discrep", "excl", "incl", "ipron", "negate", "preps",
                    "quant", "tentat", "i", "we", "you")
```

```
# For each team...
teams <- unique(doc_df$Team)

for (t in 1:length(teams)) {
  doc_df_t <- dplyr::filter(doc_df, Team == teams[t])

  for (f1 in 1:length(unique(doc_df_t$ID))) {
    for(f2 in 1:length(unique(doc_df_t$ID))){

      # Generating id-vectors for both speakers
      if(unique(doc_df_t$ID)[f1] == 0){
        a_id_temp <- c("1", unique(doc_df_t$ID)[f1])
      } else{
        a_id_temp <- c("2", unique(doc_df_t$ID)[f1])
      }


      if(unique(doc_df_t$ID)[f2] == 0){
        b_id_temp <- c("1", unique(doc_df_t$ID)[f2])
      } else{
        b_id_temp <- c("2", unique(doc_df_t$ID)[f2])
      }

      DNM_prep <- DNM_wrapper(x_df = doc_df_t, a_id = a_id_temp, b_id = b_id_temp,
team = teams[t])

      DNM_temp <- vector(mode = "numeric", length = 14)

      for (cat in 1:14) {
        DNM_temp[cat] <- DNM_coordination(DNM_prep$b[,cat+6],
DNM_prep$a[,cat+6])$LSM
      }


      temp <- c(teams[t], unique(doc_df_t$ID)[f1], unique(doc_df_t$ID)[f2], DNM_temp)
      names(temp) <-  c("Team", "ID_A", "ID_B", "article", "certain", "conj",
                  "discrep", "excl", "incl", "ipron", "negate", "preps",
                  "quant", "tentat", "i", "we", "you")

      P_P_DNM <- rbind(P_P_DNM, temp)



    }

  }

}

# Transforming the columns to mode numeric again.
```

````
for(c in 2:17){
  P_P_DNM[,c] <- as.numeric(P_P_DNM[,c])
}

# Attaching the rowMeans to the data frame as aggregate scores. NAs are ignored.
P_P_DNM <- cbind(P_P_DNM, rowMeans(P_P_DNM[,4:17], na.rm = T))
names(P_P_DNM)[ncol(P_P_DNM)] <- "Acc"


head(P_P_DNM)
```
### Person-Person Accommodation via LSM

```{r}
P_P_LSM <- as.data.frame(matrix(ncol = 4))
names(P_P_LSM) <- c("Team", "ID_A", "ID_B", "LSM")

# For each team...
teams <- unique(doc_df$Team)

for (t in 1:length(teams)) {
  doc_df_t <- dplyr::filter(doc_df, Team == teams[t])

  for (f1 in 1:length(unique(doc_df_t$ID))) {
    for(f2 in 1:length(unique(doc_df_t$ID))){
      A_temp <- doc_df_t %>%
        filter(ID == unique(doc_df_t$ID)[f1])

      B_temp <- doc_df_t %>%
        filter(ID == unique(doc_df_t$ID)[f2])

      temp <- c(teams[t], unique(doc_df_t$ID)[f1], unique(doc_df_t$ID)[f2], LSM(A_temp, B_temp))
      names(temp) <-  c("Team", "ID_A", "ID_B", "LSM")

      P_P_LSM <- rbind(P_P_LSM, temp)



    }

  }

}

# Transforming the columns to mode numeric again.
for(c in 2:4){
  P_P_LSM[,c] <- as.numeric(P_P_LSM[,c])
}
````

````
head(P_P_LSM)
```


## Writing files to disk

```{r}
# SAM files
f_f_acc <- cbind(f_f_acc$Team, rep(2, nrow(f_f_acc)),
              rep(2, nrow(f_f_acc)), f_f_acc[,2:ncol(f_f_acc)])
names(f_f_acc)[1:3] <- c("Team", "Role_A", "Role_B")

f_l_acc <- cbind(f_l_acc$Team, rep(2, nrow(f_l_acc)),
              rep(1, nrow(f_l_acc)), f_l_acc[,2:ncol(f_l_acc)])
names(f_l_acc)[1:3] <- c("Team", "Role_A", "Role_B")

l_f_acc <- cbind(l_f_acc$Team, rep(1, nrow(l_f_acc)),
              rep(2, nrow(l_f_acc)), l_f_acc[,2:ncol(l_f_acc)])
names(l_f_acc)[1:3] <- c("Team", "Role_A", "Role_B")

SAM_df <- rbind(f_f_acc, f_l_acc, l_f_acc)

write.csv(SAM_df, "./data_gen/SAM_df.csv", row.names = F)

write.csv(P_P_acc, "./data_gen/P_P_SAM.csv", row.names = F)

# LSM files

write.csv(f_l_LSM, "./data_gen/F_L_LSM.csv", row.names = F)
write.csv(P_P_LSM, "./data_gen/P_P_LSM.csv", row.names = F)

# DNM file
write.csv(P_P_DNM, "./data_gen/P_P_DNM.csv", row.names = F)

```
````

/docs/data_analysis.Rmd

````
---
title: "Data Analysis"
author: "Jakob Buske"
date: "14 June 2019"
output:
  pdf_document:
    toc: true
    number_sections: true
---

\newpage
````
````

```{r echo=FALSE, warning=FALSE, message=FALSE}
library(dplyr)
library(psych)
library(car)
library(ggplot2)
library(ggpubr)
SAM_df <- read.csv("./data_gen/SAM_df.csv")
survey <- read.csv("./data_gen/survey.csv")
P_P_SAM <- read.csv("./data_gen/P_P_SAM.csv")
P_P_DNM <- read.csv("./data_gen/P_P_DNM.csv")
F_L_LSM <- read.csv("./data_gen/F_L_LSM.csv")
doc_df <- read.csv("./data_gen/doc_df.csv", stringsAsFactors = F, dec = ",")
doc_df <- doc_df[2:nrow(doc_df),2:ncol(doc_df)]
names(doc_df)[1:5] <- c("Team", "Role", "ID", "SeqID", "Utterance")

for(i in c(2, 3, 4)){
  doc_df[,i] <- as.numeric(doc_df[,i])
}

# Utterances with unknown speakers are filtered out.
doc_df <- filter(doc_df, ID < 99)
```

Overview of Available Data Sets

In previously deployed scipts, two data sets were prepared for analysis: *SAM_df* and *survey*. The *SAM_df* data set contains all accommodation values for the sample computed with the SAM method. It is structured as following:

```{r}
head(SAM_df)
```

Here, Role A and B indicate who is accommodating to whom. Following the literature, A indicates the person speaking first, while B is the person that responds and (potentially) accommodates. Hence, all accommodation values refer to B accommodating *towards* A. The variable *n_utt* represents the number of utterances used to compute the accommodation values. Lastly, the variable *Acc* is the mean (with NAs removed) of all accommodation scores per row and can be thus seen as a total accommodation score.

The *survey* data set contains the demographics of the sample (used in Participants.pdf) as well as Leader and Team Effectiveness Scores.

```{r}
head(survey)
```

Three data sets have been created showing the Person to Person accommodation between individuals. That is, these data sets include accommodation scores for all possible dyads in the sample. This type of data set has been generated for the SAM, the DNM, and LSM.

Descriptives

Accommodation

The distribution of the SAM scores can be inspected in the following histogram. The data seem to be somewhat normally distributed. Notably, all scores are close to zero and do not make use of the whole span of the scale, which could (theoretically) range from -1 to 1. To a certain degree, the scores were expected to be lower than other measures of accommodation as this score eliminates the elements of homophily. However, these numbers are still lower than expected.

```{r, echo=FALSE}
hist(SAM_df$Acc, main="Accommodation Scores in Whole Sample", xlab =
"Accommodation")
```

Precise numbers can be retrieved from the summary statstics calculated below.

```{r, echo=FALSE}
summary(SAM_df$Acc)
sd(SAM_df$Acc)
```

Follower-to-Follower Accommodation

```{r}
F_F_Acc <- SAM_df %>%
  filter(Role_A == 2 & Role_B == 2)

summary(F_F_Acc$Acc)

sd(F_F_Acc$Acc)
```

Follower-to-Leader Accommodation

```{r}
# Note that Leader-Follower Accommodation == Follower-to-Leader Acc.!
L_F_Acc <- SAM_df %>%
  filter(Role_A == 1 & Role_B == 2)

summary(L_F_Acc$Acc)
sd(L_F_Acc$Acc)
```

Leader-to-Follower Accommodation

```{r}
F_L_Acc <- SAM_df %>%
  filter(Role_A == 2 & Role_B == 1)

summary(F_L_Acc$Acc)
sd(F_L_Acc$Acc)
```

Team Effectiveness

The distribution of the Team Effectiveness scores can be inspected in the histogram below.

```{r, echo=FALSE}
hist(survey$TeamEff, main="Individual Ratings of Team Effectiveness", xlab="Team Effectiveness")
```

```{r, echo=FALSE}
summary(survey$TeamEff)
sd(survey$TeamEff, na.rm = T)
```

As of now, the individual ratings of Team Effectivness were used. These ratings are now aggregated into team-level measures.

```{r}
TeamEff_df <- survey %>%
  group_by(TeamNumber) %>%
  summarise(TeamEff_mean = mean(TeamEff, na.rm = T),
            TeamEff_sd = sd(TeamEff, na.rm = T),
            TeamEff_n = n())

names(TeamEff_df)[1] <- "Team"
```

The aggregated Team Effectiveness scores are similiarly distributed. However, less low ratings can be found on the team-level.

```{r, echo=FALSE}
hist(TeamEff_df$TeamEff_mean, main="Team Effectiveness per Team", xlab="Team Effectiveness")
```

The summary statistics support this notion:

```{r echo=FALSE}
summary(TeamEff_df$TeamEff_mean)
sd(TeamEff_df$TeamEff_mean, na.rm = T)
```

```

### Leader Effectiveness

```{r, echo=FALSE}
hist(survey$LeEff, main="Individual Ratings of Leader Effectiness", xlab = "Leader Effectiveness")
```

```{r, echo=FALSE}
summary(survey$LeEff)
sd(survey$LeEff, na.rm = T)
```

Again, these individual ratings are aggregated to find team-level measurments.

```{r}
LeEff_df <- survey %>%
  group_by(TeamNumber) %>%
  summarise(LeEff_mean = mean(LeEff, na.rm = T),
            LeEff_sd = sd(LeEff, na.rm = T),
            LeEff_n = n())
names(LeEff_df)[1] <- "Team"

```

The distribution of Leader Effectiveness ratings is positively skewed. This seems to be as expected for situations where followers rate their leaders.

```{r echo=FALSE}
hist(LeEff_df$LeEff_mean, main="Leader Effectiveness per Team", xlab = "Leader Effectiveness")
```

```{r echo=FALSE}
summary(LeEff_df$LeEff_mean)
sd(LeEff_df$LeEff_mean, na.rm = T)
```

Internal Consistencies

A high internal consistency can indicate that all measures of one instrument (for the lack of a better term) indeed measure the same thing. Commonly, Cronbach's alpha is used for this. However, Gutmans Lambda 2 (Or ten Berge's mu 1; they are the same) is more accurate. Accordingly, Gutmans Lambda 2 will be used as a method to establish internal consistency.

Internal Consistiency of the SAM Measures
```

The internal consistency of the SAM measures can be inspected below. Again, the mu 1 is the relevant measure in this case.
```{r}
(total_tb <- tenberge(SAM_df[,4:17]))
```
```{r}
f_f_tb <- tenberge(SAM_df %>%
              filter(Role_A == 2 & Role_B == 2) %>%
              select(c(4:17)))
```

```{r}
f_l_tb <- tenberge(SAM_df %>%
              filter(Role_A == 2 & Role_B == 1) %>%
              select(c(4:17)))
```

```{r}
l_f_tb <- tenberge(SAM_df %>%
              filter(Role_A == 1 & Role_B == 2) %>%
              select(c(4:17)))
```

The total internal consistency is low with a lambda 2 of `r round(total_tb$mu1, digits=2)`. Interestingly, the consistency is not the same for all directions of measurement: While both Follower-Follower and Follower-Leader scores have a higher internal consistency (`r round(f_f_tb$mu1, digits=2)` and `r round(f_l_tb$mu1, digits=2)`), the internal consistency of the Leader-Follower measures approaches zero (`r round(l_f_tb$mu1, digits=2)`).

```{r}
P_P_SAM_tb <- tenberge(P_P_SAM[,4:17])
P_P_DNM_tb <- tenberge(P_P_DNM[,4:17])
```

The Person-Person scores found for the SAM support these numbers (`r round(P_P_SAM_tb$mu1, digits=2)`). The internal consistency of the Person-Person data set using the DNM method yielded a lambda 2 of `r round(P_P_DNM_tb$mu1, digits=2)`. This unexpected high lambda can be explained by the bias shared amongst the different measurements (see Simultaion.pdf).

Internal Consistencies of the Survey Questions

```{r}
LeEff_tb <- tenberge(survey[,7:10])
TeamEff_tb <- tenberge(survey[,11:14])
```

The lambda 2 of the four items regarding Leader Effectiveness is `r round(LeEff_tb$mu1, digits=2)`.

The lambda 2 of the four items regarding Team Effectiveness is `r round(TeamEff_tb$mu1, digits=2)`.

Internal Consistencies: Conclusion

The internal consistencies of the survey items are good. However, the internal consistencies of the accommodation measures are low. Simulations (see Simulation.pdf) show that if there was an underlying factor of accommodation, the SAM would have yielded a satisfactory internal consistency even with merely 5 utterances per dyad (although estimation accuracy would suffer). The data at hand seem to indicate that the levels of coordination are a) low and b) influenced by random error. Simulations with the accommodation paramenter set equal to zero and being randomly determined seem to support this notion.

Testing the Hypotheses

Assumptions of Regression Analyses

A few assumptions are used when performing a Regression Analysis. Most of them do not actually influence the model itself, but rather significance tests related to the model (e.g. the p-value of a coefficient). The general outline for checking these assumptions based on Field (2013, p. 311) is described here.

* Additivity and Linearity
  - The relationship between the variables is linear. If not met, the model is invalid.

* Independent Error
  - The residuals of the observations do not correlate (-> Autocorrelation). If not met, confidence intervals and significance tests will be invalid.
  - Can be tested via Durbin-Watson test. For values above 3 and below 1, check original paper.

* Homoscedasticity
  - The residual's variance stays the same at each level. If not met, confidence intervals and significance tests will be invalid.

* Normally Distributed Errors
  - Invalidates confidence intervals and significance tests in small samples if not met.

* Predictors are uncorrelated with external variables
  - Rather theoretical.

* No perfect Multicollinearity
  - Not relevant for simple linear regression.

* No Outlier Influences the Model

Hypothesis 1A: High follower-follower convergence is positively related to team effectiveness.

```{r}
h1A_df <- filter(SAM_df, Role_A == 2, Role_B == 2)
h1A_df <- inner_join(h1A_df, TeamEff_df, by = c("Team"))
```

Creating the Model

```{r}
fit_h1A <- lm(h1A_df$TeamEff_mean ~ h1A_df$Acc)
summary(fit_h1A)
```

The summary of the model indicates that there is not evidence for a relation between follower-follower convergence and team effectiveness (p ~ .55).

Checking the Assumptions of H1A

No relations is apparent between the Team Effectiveness and Accommodation scores. However, no non-linear relationship is apparent, either. Hence, the **assumption of additivity and linearity can regarded as a given**. Some data points, especially when looking at the Accommodation axis, could be regarded as outliers. These can be ignored (for now) as they could not hide any effect, meaning that the **assumption of non-existence of outliers is accepted**.

```{r}
plot(h1A_df$TeamEff_mean ~ h1A_df$Acc)
```

The Durbin-Watson test supports the notion that the **errors of the model are independent** of each other.

```{r}
durbinWatsonTest(fit_h1A)
```

The variance seems to become higher as Team Effectiveness approaches zero. This indicates a **violation of the assumption of heterogeneity of variance**.

```{r}
plot(rstandard(fit_h1A), fit_h1A$fitted.values, xlab = "Standardized Residuals", ylab = "Predicted Team Effectiveness")
```

The **errors of the model** seem to be **normally distributed**.

```{r}
hist(fit_h1A$residuals, main = "", xlab = "Residuals of Fit H1A")
```

Hypothesis 1B: High leader-follower convergence is positively related to team effectiveness.

```{r}
h1B_df <- filter(SAM_df, Role_A == 1, Role_B == 2)
h1B_df <- inner_join(h1B_df, TeamEff_df, by="Team")
```

Creating the Model

```{r}
fit_h1B <- lm(h1B_df$TeamEff_mean ~ h1B_df$Acc)
summary(fit_h1B)
```

The summary of the model indicates that there is not evidence for a relation between leader-follower convergence and team effectiveness (p ~ .95).

Checking the Assumptions of H1B

No relation is apparent between the Team Effectiveness and Accommodation scores. However, no non-linear relationship is apparent, either. Hence, the **assumption of additivity and linearity can regarded as given**. Some data points, especially when looking at the Accommodation axis, could be regarded as outliers. These can be ignored (for now) as is unlikely they hide any meaningfull effect, meaning that the **assumption of non-existence of outliers is accepted**.

```{r}
plot(h1B_df$TeamEff_mean ~ h1B_df$Acc, xlab = "Accommodation", ylab = "Team Effectiveness")
```

The Durbin-Watson test supports the notion that the **errors of the model are independent** of each other.

```{r}
durbinWatsonTest(fit_h1B)
```

There seems to be a change of variance throughout the following plot. This indicates a weak **violation of the assumption of heterogeneity of variance**.

```{r}
plot(rstandard(fit_h1B), fit_h1B$fitted.values, xlab = "Standardized Residuals", ylab =
"Predicted Team Effectiveness")
```

The **errors of the model** seem to be **normally distributed**.

```{r}
hist(fit_h1B$residuals, main = "", xlab = "Residuals of Fit H1B")
```

Hypothesis 1C: High follower-leader convergence is positively related to team effectiveness.

```{r}
h1C_df <- filter(SAM_df, Role_A == 2, Role_B == 1)
h1C_df <- inner_join(h1C_df, TeamEff_df, by="Team")
```

Creating the Model

```{r}
fit_h1C <- lm(h1C_df$TeamEff_mean ~ h1C_df$Acc)
summary(fit_h1C)
```

The summary of the model indicates that there is not evidence for a relation between follower-follower
convergence and team effectiveness (p ~ .2).

Checking the Assumptions of H1C

No relation is apparent between the Team Effectiveness and Accommodation scores. However, no non-linear relationship is apparent, either. Hence, the **assumption of additivity and linearity can regarded as given**. Some data points, especially when looking at the Accommodation axis, could be regarded as outliers. These can be ignored (for now) as is unlikely they hide any meaningfull effect, meaning that the **assumption of non-existence of outliers is accepted**.

```{r}
plot(h1C_df$TeamEff_mean ~ h1C_df$Acc, xlab = "Accommodation", ylab = "Team Effectiveness")
```

The Durbin-Watson test supports the notion that the **errors of the model are independent** of each other.

```{r}
durbinWatsonTest(fit_h1C)
```

The variance is the same throughout the plot. This indicates that the **assumption of heterogeneity of variance can be accepted**.

```{r}
plot(rstandard(fit_h1C), fit_h1C$fitted.values, xlab = "Standardized Residuals", ylab = "Team Effectiveness")
```

The **errors of the model** seem to be somewhat **normally distributed**.

```{r}
hist(fit_h1C$residuals, main = "", xlab = "Residuals of Fit H1C")
```

Hypothesis 2: Leaders displaying more converging behaviors are more effective.

```{r}
h2_df <- filter(SAM_df, Role_A == 2, Role_B == 1)
h2_df <- inner_join(h2_df, LeEff_df, by="Team")
```

Creating the Model

```{r}
fit_h2 <- lm(h2_df$LeEff_mean ~ h2_df$Acc)
summary(fit_h2)
```

The summary of the model indicates that there is not evidence for a relation between Follower-Leader
convergence and leader effectiveness (p ~ .46).

Checking the Assumptions of H1C

No relation is apparent between the Leader Effectiveness and Accommodation scores. However, no non-linear relationship is apparent, either. Hence, the **assumption of additivity and linearity can regarded as given**. Some data points, especially when looking at the Accommodation axis, could be regarded as outliers. These can be ignored (for now) as is unlikely they hide any meaningfull effect, meaning that the **assumption of non-existence of outliers is accepted**.

```{r}
plot(h2_df$LeEff_mean ~ h2_df$Acc, xlab = "Accommodation", ylab = "Leader Effectiveness")
```

The Durbin-Watson test supports the notion that the **errors of the model are independent** of each other.

```{r}
durbinWatsonTest(fit_h2)
```

The variance is the same throughout the plot. This indicates that the **assumption of heterogeneity of variance can be accepted**.

```{r}
plot(rstandard(fit_h2), fit_h2$fitted.values, xlab = "Standardized Residuals", ylab = "Leader Effectiveness")
```

The **errors of the model** seem to be somewhat **normally distributed**.

```{r}
hist(fit_h2$residuals, main = "", xlab = "Residuals of Fit H1A")
```

Ancillary Analyses

Linguistic Style Matching (LSM)
One special characteristic of the SAM is that it only observes actual processes of communication style adaptation (i.e. accommodation) and ignores baselines. This eliminates the danger of confusing accommodation with homophily (i.e. just having similiar language to each other). However, in literature, these two concepts are not properly separated from each other. Thus, LSM will now be used to test the prior hypotheses to gain a deeper understanding on the differences of these constructs.

```{r}
LSM_df <- inner_join(F_L_LSM, TeamEff_df, by="Team")
LSM_df <- inner_join(LSM_df, LeEff_df, by="Team")
head(LSM_df)
```

Firstly, the summary statistics are computed.
```{r}
summary(LSM_df$LSM)
sd(LSM_df$LSM)
```

```{r}
summary(LSM_df$LSM)
sd(LSM_df$LSM)
```

```{r}
fit_LSM_h1c <- lm(LSM_df$TeamEff_mean ~ LSM_df$LSM)
summary(fit_LSM_h1c)
```

```{r}
fit_LSM_h2 <- lm(LSM_df$LeEff_mean ~ LSM_df$LSM)
summary(fit_LSM_h2)
```

The two regression models formulated with LSM as a measure also do not show any relations between Accommodation and Team and Leader Effectiveness.

Differences Between the Baselines
It could be that Accommodation does not occur on an utterance-to-utterance level, but takes more time to come into effect. One possible case would be that the team members already adapt their communication style towards the style of the group at the very beginning of the meeting. In other words, a leader might change his/her language just by knowing that a certain set of followers will be present. In that case, no accommodation could be observed with the method used in this paper. Instead, the leaders' baselines would have to be compared in different settings. However, tentative support for this idea could be gained if the function word baselines of the teams would be significantly different from each other. This would indicate that each group has a specific 'signature' of function words.

To test this, an anova is deployed.

First, a suitable data frame is generated from doc_df.

```{r}
group_base <- doc_df %>%
```

````
  group_by(Team, ID) %>%
  summarise(article = mean(article),
            certain = mean(certain),
            conj = mean(conj),
            discrep = mean(discrep),
            excl = mean(excl),
            incl = mean(incl),
            ipron = mean(ipron),
            negate = mean(negate),
            preps = mean(preps),
            quant = mean(quant),
            tentat = mean(tentat),
            i = mean(i),
            we = mean(we),
            you = mean(you))

group_base <- bind_cols(group_base[,1], group_base[,3:ncol(group_base)])
group_base$Team <- as.factor(group_base$Team)
```
````

Looking at the boxplots for the category *article* per team, it seems that there are no big differences between the baselines.

````
```{r}
ggboxplot(group_base, x = "Team", y = "article", color = "Team") +
theme(legend.position="none")
```
````

````
```{r}
aov_p_val <- vector(mode="numeric", length(ncol(group_base)-1))
for(c in 2:(ncol(group_base))){
  aov_fit <- aov(group_base[[c]] ~ group_base[[1]])
  aov_p_val[c-1] <- summary(aov_fit)[[1]][["Pr(>F)"]][[1]]
}
options("scipen"=100, "digits"=4)

print(aov_p_val)
```
````

The summary statistics of the anova can be seen here:

````
```{r echo=FALSE}
summary(aov_p_val)
```
````

With a median p value of 0.003, it can be assumed that for most categories, the groups have significantly baselines. However, an anove does already react when one (of 77!) mean is different from the rest. This alone, however, does not support the idea that each team has a different 'finger print'. Moreover, the analysis would benefit if all 14 categories could be included at the same time.
````

Correlations within the data

Finding the correlations between Team and Leader Effectiveness, L-t-F Acc., F-t-L Acc., F-t-F Acc. .
```{r}
cor_df <- inner_join(TeamEff_df, LeEff_df, by = "Team") # Team and Leader Effectiveness
cor_df <- cor_df[,c("Team", "TeamEff_mean", "LeEff_mean")]
L_T_F <- filter(SAM_df, Role_A == "1" & Role_B == "2") # Leader-to-Follower Acc
L_T_F <- L_T_F[,c("Team", "Acc")]
cor_df <- inner_join(cor_df, L_T_F, by = "Team")
F_T_L <- filter(SAM_df, Role_A == "2" & Role_B == "1") # Follower-to-Leader Acc
F_T_L <- F_T_L[,c("Team", "Acc")]
cor_df <- inner_join(cor_df, F_T_L, by = "Team")
F_T_F <- filter(SAM_df, Role_A == "2" & Role_B == "2") # Follower-to-Follower Acc
F_T_F <- F_T_F[,c("Team", "Acc")]
cor_df <- inner_join(cor_df, F_T_F, by = "Team")

cor_df_c <- cor(cor_df[,2:ncol(cor_df)])
write.csv("./data_gen/cor_table.csv", row.names = F)
head(cor_df_c)
```

Conclusion

No relation between Accommodation and Team and Leader Effectiveness could be found in this analysis. However, the evidence indicating that there is no such a relation is also weak, as the internal consistencies of the SAM measures raise the doubt that there was no Accommodation to measure. Due to this, the priority for future research should be the refinement of measuring Accommodation while keeping theoretical considerations in mind:

* Does Accommodation only occur in new teams, but not in already established ones?
* Does Accommodation (commonly) take place at the utterance level or does is prevail mostly on broader levels (e.g. the leader adapting his/her own language for the whole meeting at once)?

/docs/dic_info.Rmd

```
---
title: "DictInfo"
author: "Jakob Buske"
date: "18 Mai 2019"
output: pdf_document
---
```

```{r}
setwd("C:/Users/jakob/Documents/Text Mining/Thesis_Code")
```

~~~
dic <- read.csv("./data/Dutch_LIWC2007_Dictionary.csv")
annotations <- dic[68:nrow(dic),]
annotations[,2] <- as.numeric(as.character(annotations[,2]))

for(i in 1:66){
  temp_id <-  as.numeric(as.character(dic[i,1]))
  temp_name <- dic[i,2]

  temp_sum <- as.character(sum(annotations[,2:ncol(annotations)] == temp_id, na.rm = T))

  true_rows <- rowSums(annotations[,2:ncol(annotations)] == temp_id, na.rm = T)
  true_rows <- annotations[true_rows > 0,]

  #true_rows <- true_rows[true_rows > 0]

  example <-  true_rows[sample(1:nrow(true_rows), 3, replace = T),]
  example <- paste(example$X., collapse = ", ")

  print(paste(temp_name, ": ", temp_sum, " --- ", example))
}
```
~~~

/docs/participants.Rmd

Omitted due to confidentiality agreements.

/docs/simulation.Rmd

~~~
---
title: "SAM Simulation"
author: "Jakob Buske"
date: "24 June 2019"
output: pdf_document
---

```{r results='hide', warning=FALSE, message=FALSE}
setwd("C:/Users/jakob/Documents/Text Mining/Thesis_code")
library(dplyr)


source("./R/helpers.R")
```
~~~

The method is implemented in the SAM() (simplified alignment model - for the lack of a better name) function.

~~~
```{r}

SAM
~~~

```

For the sake of a comparison, the DNM method is also implemented.

```{r}
DNM_coordination
```

Data set will be simulated with the SAM_sim() function.

```{r}
SAM_sim
```

Ideal Accuracy

Mathematical procedures are commonly evaluated by relying on analytical means. In this case, Monte Carlo simulations are choosen due to their approachability.

The first simulation is aimed at estimating how accurate the measures of alignment are under (almost) ideal conditions. As such, this simulation does not directly yield insight into the usefulness of these measures in an empirical situation, but merely demonstrates that, in theory, the developed procedures are suited to measures linguistic alignment. To show this, 25000 data sets with random parameters are simulated.

Preliminary analysis showed that the measures at hand might experience difficutlies when the baselines were high. This observation can be explained intuitivley: When working with propabilities, one requires many observations to benefit from the Law of Large Numbers. When the baselines are high, only few cases where either A or B do not show marker m are available. As a result, estimating underlying propabilities becomes harder. Luckily, prior experience indicates that the baselines are normally low, ranging from 0.01 to 10 %. Due to this, the simulation focuses on this range. However, more extreme baselines are also sampled in order to verify the preliminary analysis.

```{r echo=FALSE}
n <- 25000
```

```{r eval = FALSE, echo = TRUE}
start_time <- Sys.time()
df_1 <- data.frame(A_base_param = vector(mode = "numeric",length = n),
                B_base_param = vector(mode = "numeric",length = n),
                coord_param = vector(mode = "numeric",length = n),
                A_base_estimate = vector(mode = "numeric",length = n),
                B_base_estimate = vector(mode = "numeric",length = n),
                Align = vector(mode = "numeric",length = n),
                C_base = vector(mode = "numeric",length = n),

```
            C_align = vector(mode = "numeric",length = n),
            DNM_coord = vector(mode = "numeric", length = n),
            A_mean = vector(mode = "numeric", length = n),
            N_utt = vector(mode = "numeric", length = n))

for (i in 1:n) {
  print(i)
  temp <- vector(length = 8)

  A_base <- base::sample(seq(0.001, 0.9, by=0.001), size = 1, prob = c(rep(50, 100), rep(1,
800)))
  B_base <- base::sample(seq(0.001,  0.9, by=0.001), size = 1, prob = c(rep(50, 100), rep(1,
800)))

  coord <- base::sample(seq(-(B_base), 1-B_base, by=0.01), size = 1)

  N_utt <- 1000
  test_df <- SAM_sim(A_base = A_base, B_base = B_base, coord = coord, N_utt = N_utt)
  SAM_out <- SAM(test_df)
  DNM <- DNM_coordination(b = test_df$B, a = test_df$A)

  #Testing
  a_mean_sv <- mean(test_df$A > 0)

  temp <- c(A_base, B_base, coord, sum(test_df$A) / sum(test_df$A_WC),
        sum(test_df$B)/sum(test_df$B_WC), SAM_out$Align, SAM_out$C_base,
SAM_out$C_align, DNM$LSM, a_mean_sv)
 # names(temp) <- c("A_base_param", "B_base_param", "coord_param", "A_base_estimate",
   #            "B_base_estimate", "Align", "C_base", "C_align", "DNM_coord")

  df_1[i,] <- temp


}

stop_time <- Sys.time()

# For n = 100: 0.89044 sec.
# For n = 25000: 12.51 mins.
print(stop_time - start_time)
#write.csv(df_1, "df_1.csv")
```

```{r}
df_1 <- read.csv("./data_gen/df_1.csv")
df_1 <- df_1[,2:ncol(df_1)]
```

```

## Estimating the Baselines

```{r}
A_base_cor <- cor(df_1$A_base_param, df_1$A_base_estimate)
B_base_cor <- cor(df_1$B_base_param, df_1$B_base_estimate)

cat(paste("The correlation for A_base is: ", as.character(A_base_cor), "\n",
        "The correlation for B_base is: ", as.character(B_base_cor), "\n", sep=""))

```
The baseline of A can be estimated with high accuracy based on the simulated data. However, the correlation of the true and estimated baseline of B is low. This is expected as the baseline estimate used in the simulation does not exclude the effect of coordination. This can be seen in the following plot. Low coordination values (Black) lead to an underestimation of the baseline. However, high values of coordination (Blue) lead to an overestimation of the baseline.

```{r}
coord_cut <- cut(df_1$coord_param, c(-1, 0.14, 0.4, 0.67, 1))
plot(df_1$B_base_param ~ df_1$B_base_estimate, col=c("black", "red", "green", "blue", "orange")[coord_cut])
```

SAM Alignment

Ideally, the used SAM method perfectly predicts the true coordination used in this simulation. This can be evaluated by plotting the true (coord_param) vs the estimated (Align) coordination levels. Both true coor, indicating that the alignment measured via the SAM function is a suitable predictor of true coordination. Notably, the errors seem to be mostly randomly distributed around the diagonal. Accordingly, one can infer that there is no bias.

```{r}

plot(df_1$coord_param ~ df_1$Align,
    xlab = "Accommodation Estimated via SAM",
    ylab = "True Accommodation")
jpeg("./data_gen/SIM_SAM.jpeg")
plot(df_1$coord_param ~ df_1$Align,
    xlab = "Accommodation Estimated via SAM",
    ylab = "True Accommodation")
dev.off()

```

This notion is supported by a linear model. Here, an adjusted R-squared of .98 indicates that the SAM provides a good estimate of coordination.
```

```{r}
fit <- lm(df_1$coord_param ~ df_1$Align)
summary(fit)
```

DNM Alignment

The SAM method of calculation is a potential replacement for the previously considered DNM method. Thus, DNM should serve as a comparison to SAM. This will indicate which of these methods is more accurate. Firstly, true vs estimated (via DNM) data are plotted.

As can be seen, the DNM method yields a distorted relation between true and estimated values. Further analysis showed that the DNM score depends on the baseline of A, as indicated by the varying colors. This has two implications:

1) The higher the baseline of A, the less sensitive is the DNM method in estimating coordination levels.

2) Even if the baselines of A are in a range that yields sufficient estimates, outcome scores for different markers cannot be compared as they are not on the same scale.

```{r}
a_m <- cut(df_1$A_mean, c(0, 0.4, 0.55, 0.67, 1))

plot(df_1$coord_param ~ df_1$DNM_coord, col=c("black", "red", "green", "blue",
"orange")[a_m],
   xlab = "Accommodation Estimated via SCP", ylab = "True Accommodation")
legend(x = 0.4, y = -0.1, legend = c(" 0 - .4",
                              ".4 - .55",
                              ".55 - .67",
                              ".67 - 1"),
     col = c("black", "red", "green", "blue", "orange"),
     pch = 1)

jpeg("./data_gen/SIM_SCP.jpeg")
plot(df_1$coord_param ~ df_1$DNM_coord, col=c("black", "red", "green", "blue",
"orange")[a_m],
   xlab = "Accommodation Estimated via SCP", ylab = "True Accommodation")
legend(x = 0.4, y = -0.1, legend = c(" 0 - .4",
                              ".4 - .55",
                              ".55 - .67",
                              ".67 - 1"),
     col = c("black", "red", "green", "blue", "orange"),
     pch = 1)
dev.off()

```

This notion is supported by a linear model which yields an explained variance of .63 . Based on the previous plot it can be expected that the DNM method is more accurate when only observing one baseline in isolation, given that the baseline of A is not too high. However, the violation of this assumption poses a considerable risk which should be avoided. Moreover, the ability to compare different scores is a key element as all 14 alignment scores derived by the 14 marker categories need to be aggregated. As long as not all of these scores have the same baseline by any chance, the mean of the single alignment scores would be biased towards zero and thereby lower the variance in observed alignment.

```{r}
fit_dnm <- lm(df_1$coord_param ~ df_1$DNM)
summary(fit_dnm)
```

Influence of Number of Data Points Available

The method proposed is probabilistic and thus requires a sufficient amount of observations in order to accurately estimate probability. Thus, the next simulation will vary the amount of utterance dyads observed to evaluate model accurary relative to available data points. In the actual data set, most leaders seem to have around 200 utterance dyads. Using a range of 1 up to 1000 should cover all extremes that could be found in the sample at hand.

```{r echo=FALSE}
n <- 25000
```

```{r eval = FALSE, echo = TRUE}
start_time <- Sys.time()
df_2 <- data.frame(A_base_param = vector(mode = "numeric",length = n),
                   B_base_param = vector(mode = "numeric",length = n),
                   coord_param = vector(mode = "numeric",length = n),
                   A_base_estimate = vector(mode = "numeric",length = n),
                   B_base_estimate = vector(mode = "numeric",length = n),
                   Align = vector(mode = "numeric",length = n),
                   C_base = vector(mode = "numeric",length = n),
                   C_align = vector(mode = "numeric",length = n),
                   DNM_coord = vector(mode = "numeric", length = n),
                   A_mean = vector(mode = "numeric", length = n),
                   N_utt = vector(mode = "numeric", length = n))

for (i in 1:n) {
  print(i)
  temp <- vector(length = 8)

  A_base <- base::sample(seq(0.001, 0.9, by=0.001), size = 1, prob = c(rep(50, 100), rep(1, 
800)))
  B_base <- base::sample(seq(0.001,  0.9, by=0.001), size = 1, prob = c(rep(50, 100), rep(1, 
800)))
```

```
  coord <- base::sample(seq(-(B_base), 1-B_base, by=0.01), size = 1)

  N_utt <- sample(1:1000, 1)
  test_df <- SAM_sim(A_base = A_base, B_base = B_base, coord = coord, N_utt = N_utt)
  SAM_out <- SAM(test_df)
  DNM <- DNM_coordination(b = test_df$B, a = test_df$A)

  #Testing
  a_mean_sv <- mean(test_df$A > 0)

  temp <- c(A_base, B_base, coord, sum(test_df$A) / sum(test_df$A_WC),
          sum(test_df$B)/sum(test_df$B_WC), SAM_out$Align, SAM_out$C_base,
SAM_out$C_align, DNM$LSM, a_mean_sv, N_utt)
 # names(temp) <- c("A_base_param", "B_base_param", "coord_param", "A_base_estimate",
   #              "B_base_estimate", "Align", "C_base", "C_align", "DNM_coord")

  df_2[i,] <- temp


}

stop_time <- Sys.time()

# For n = 100: 0.89044 sec.
# For n = 25000: 12.51 mins.
print(stop_time - start_time)
#write.csv(df_2, "df_2.csv")
```

```{r}
df_2 <- read.csv("./data_gen/df_2.csv")
df_2 <- df_2[,2:ncol(df_2)]
```

An initial plot does not seem to display a strong relation between the number of utterances observed and model accuracy.

```{r}
a_m <- cut(df_2$N_utt, c(0, 50, 100, 500, 1000))

plot(df_2$coord_param ~ df_2$Align, col=c("black", "red", "green", "blue", "orange")[a_m])
```

Also, there does not seem to be a relation between the number of observed utterances and residuals of a linear model.

```{r}
fit_n_utt <- lm(df_2$coord_param ~ df_2$N_utt)
df_2$res <- residuals(fit_n_utt)
```

```
library(ggplot2)
ggplot(df_2,aes(x=df_2$N_utt,y=df_2$res)) + stat_binhex()
```

This relation only becomes apparent when filtering the data set, keeping only instances with a small number of observed utterances. Here, the R-squared of a linear regression drops to moderate .78 .

```{r}
df_small <- filter(df_2, N_utt < 100)

#df_small <- na.omit(df_small)

fit_small <- lm(df_small$coord_param ~ df_small$Align)
summary(fit_small)

plot(df_small$coord_param ~ df_small$Align)
```

In the above plot, two special situations can be observed: In certain cases, the Align turns either zero or > 0.22, seemingly independent from the true coordination parameter. Manual inspection of the data set shows that this happens when C_align or C_base are zero or one. Obviously, this is not truly due to very high levels of coordination but simply because not enough utterances were collected to observe the expression of a marker.

These special cases can be filtered out. However, this filtering does not affect the overall trend enough to justify implementing an outlier removal rule for the analysis of the empirical data.

```{r}
df_small_cleaned <- filter(df_small, C_align > 0.001 & C_align < 0.99 | C_base > 0.001)

fit_small_cleaned <- lm(df_small_cleaned$coord_param ~ df_small_cleaned$Align)
summary(fit_small_cleaned)


plot(df_small_cleaned$coord_param ~ df_small_cleaned$Align)
```

Based on these observations it can be concluded that one should aim for 100 observations as a bare minimum for a reliable estimation of coordination parameters. In practice, the analysis benefits from the fact that 14 categories of markers are evaluated. If the assumption that all 14 estimates of coordination share one latent variable (i.e. "true" coordination) holds, the minimum requirement of 100 observations can be easily satisfied.

Investigating the Internal Consistency of SAM and DNM

Premimilary analysis showed that the internal consistency of the SAM low (lamda 2 ~ .26), while the internal consistency of the DNM was moderate (lambda 2 ~ 0.67). A suspicion arose that the higher internal consistency of the DNM was not due to a relation between DNM and real coordination, but rather to a shared bias of the items based on utterance length.

To investigate this suspicion, data sets with equal parameters are generated 14 times. Thus, these data sets (and their corresponding SAM/DNM scores) will be different from each other, but the underlying parameters are the same. This process will be repeated n/14 times.

The result is one data set that simulates the empirically collected data. Three of those data sets with varying conditions are generated:

* There is an underlying coordination value != 0.
* The underlying coordination value varies randomly.
* The underlying coordination is euqal to zero.

Underlying Coordination Value != 0
```{r echo=FALSE}
# Has to be divisble by 14!
n <- 4200
```

```{r echo=TRUE, warning=FALSE}
start_time <- Sys.time()
df_3 <- data.frame(A_base_param = vector(mode = "numeric",length = n),
               B_base_param = vector(mode = "numeric",length = n),
               coord_param = vector(mode = "numeric",length = n),
               A_base_estimate = vector(mode = "numeric",length = n),
               B_base_estimate = vector(mode = "numeric",length = n),
               Align = vector(mode = "numeric",length = n),
               C_base = vector(mode = "numeric",length = n),
               C_align = vector(mode = "numeric",length = n),
               DNM_coord = vector(mode = "numeric", length = n),
               A_mean = vector(mode = "numeric", length = n))


# Setting up parameter vectors.

# The baselines are assumed to be different in each run: A person can have two
# different baselines for two different marker categories at the same time.

# Based on prior findings, the baselines are between 0.001 and 0.2, a range
# only a little wider than the actually observed range.
A_base <- base::sample(seq(0.001, 0.2, by=0.001), replace=T, size = n)
B_base <- base::sample(seq(0.001,  0.2, by=0.001), replace=T, size = n)


# Getting Word counts (i.e. utterance lengths). As we simultate the
```

```
# measurement of 14 categories per utterance, the word count repeats
# itself 14 times.
N_utt <- 5
N_len_decay = 0.05
WC_temp <- list()
A_WC_temp <- list()
for(i in 1:(n/14)){
  WC_temp[[i]] <- ceiling(rexp(N_utt, N_len_decay))
  A_WC_temp[[i]] <- ceiling(rexp(N_utt, N_len_decay))
}

WC <- list()
A_WC <- list()
for (i in 1:n) {
  WC[[i]] <- WC_temp[[ceiling(i/14)]]
  A_WC[[i]] <- A_WC_temp[[ceiling((i/14))]]
}

# Getting coordination parameters. As coord is assumed to stay constant
# throughout all 14 categories, it is repeated 14 times. Normally, coord
# is dependent of the B baseline. To decrease code complexity, the baseline
# is entered manually (0.05). As a result, unrealistic values could arise.
# However, this should not affect the overall validity of the simulation.
coord <- c()
for(i in 1:(n/14)){

  coord <- c(coord, rep(base::sample(seq(-0.05, 1-0.05, by=0.01), 1), 14))
}


for (i in 1:n) {
  temp <- vector(length = 8)

  #coord <- rep(base::sample(seq(-(B_base_temp[ceiling(i/14)]), 1-
B_base_temp[ceiling(i/14)], by=0.01)))

  test_df <- SAM_sim(A_base = A_base[i], B_base = B_base[i], coord = coord[i], N_utt =
N_utt,
                  WC = WC[[i]], A_WC = A_WC[[i]])



  SAM_out <- SAM(test_df)
  DNM <- DNM_coordination(b = test_df$B, a = test_df$A)

  #Testing
  a_mean_sv <- mean(test_df$A > 0)

  temp <- c(A_base[i], B_base[i], coord[i], sum(test_df$A) / sum(test_df$A_WC),
```

```
        sum(test_df$B)/sum(test_df$B_WC), SAM_out$Align, SAM_out$C_base,
SAM_out$C_align, DNM$LSM, a_mean_sv)
 # names(temp) <- c("A_base_param", "B_base_param", "coord_param", "A_base_estimate",
   #             "B_base_estimate", "Align", "C_base", "C_align", "DNM_coord")

  df_3[i,] <- temp


}

stop_time <- Sys.time()

# For n = 100: 0.89044 sec.
# For n = 25000: 12.51 mins.
print(stop_time - start_time)
#write.csv(df_3, "./data_gen/df_3.csv")
```

Now, the data frame is transformed to establish the internal consistency.

```{r}
df_3_turned <- as.data.frame(matrix(ncol = 29))


for (i in 1:(n/14)) {
  df_3_turned[i,1] <- df_3$coord_param[i * 14]
  df_3_turned[i,2:ncol(df_3_turned)] <- c(df_3$Align[((i-1)*14+1):((i-1)*14+14)],
                                     df_3$DNM_coord[((i-1)*14+1):((i-1)*14+14)])

}

df_3_turned[,ncol(df_3_turned)+1] <- rowMeans(df_3_turned[,2:15], na.rm = T)
df_3_turned[,ncol(df_3_turned)+1] <- rowMeans(df_3_turned[,16:29], na.rm = T)
```

```{r}
library(psych)

splitHalf(df_3_turned[,2:15], check.keys = F)
```

```{r}
splitHalf(df_3_turned[,16:29], check.keys = F)
```

```{r}
fit_SAM <- lm(df_3_turned[,1] ~ df_3_turned[,30])
summary(fit_SAM)
```

```{r}
fit_DNM <- lm(df_3_turned[,1] ~ df_3_turned[,31])
summary(fit_DNM)
```

Despite the remarkable sparseness of the data (5 utterances per pair), the internal consistency of both methods is well. Here, the SAM seems to be more coherent than the DNM.

When fitting the coordination estimates to the true parameters, the SAM performs moderately well (R^2 ~ .82), while the DNM is less accurate in predicting coordination (R^2 ~ .58). Notably, the SAM performs better than in the previous simulation investigating the requirements of the SAM. There, a minimum of 100 utterances was found for a R^2 of .8. However, the simulation at hand uses 70 (14 * 5) utterances.

It can be concluded that given that there is a shared underlying factor as coordination, the SAM is able to estimate it, which is also shown in its internal consistency.

The underlying coordination value varies randomly.

This simulation randomly varies coordination to investigate how the DNM method responds to random data. It is expected that the internal consistency still remains high due to the bias all "items" share.

```{r}
n <- 4200
```

```{r echo=TRUE, warning=F}
start_time <- Sys.time()
df_4 <- data.frame(A_base_param = vector(mode = "numeric",length = n),
              B_base_param = vector(mode = "numeric",length = n),
              coord_param = vector(mode = "numeric",length = n),
              A_base_estimate = vector(mode = "numeric",length = n),
              B_base_estimate = vector(mode = "numeric",length = n),
              Align = vector(mode = "numeric",length = n),
              C_base = vector(mode = "numeric",length = n),
              C_align = vector(mode = "numeric",length = n),
              DNM_coord = vector(mode = "numeric", length = n),
              A_mean = vector(mode = "numeric", length = n))


# Setting up parameter vectors.

# The baselines are assumed to be different in each run: A person can have two
# different baselines for two different marker categories at the same time.

# Based on prior findings, the baselines are between 0.001 and 0.2, a range
# only a little wider than the actually observed range.
```

```
A_base <- base::sample(seq(0.001, 0.2, by=0.001), replace=T, size = n)
B_base <- base::sample(seq(0.001,  0.2, by=0.001), replace=T, size = n)


# Getting Word counts (i.e. utterance lengths). As we simultate the
# measurement of 14 categories per utterance, the word count repeats
# itself 14 times.
N_utt <- 5
N_len_decay = 0.05
WC_temp <- list()
A_WC_temp <- list()
for(i in 1:(n/14)){
  WC_temp[[i]] <- ceiling(rexp(N_utt, N_len_decay))
  A_WC_temp[[i]] <- ceiling(rexp(N_utt, N_len_decay))
}

WC <- list()
A_WC <- list()
for (i in 1:n) {
  WC[[i]] <- WC_temp[[ceiling(i/14)]]
  A_WC[[i]] <- A_WC_temp[[ceiling((i/14))]]
}

# Getting coordination parameters. Normally, coord
# is dependent of the B baseline. To decrease code complexity, the baseline
# is entered manually (0.05). As a result, unrealistic values could arise.
# However, this should not affect the overall validity of the simulation.
coord <- base::sample(seq(-0.05, 1-0.05, by=0.01), size = n, replace = T)



for (i in 1:n) {
  temp <- vector(length = 8)

  #coord <- rep(base::sample(seq(-(B_base_temp[ceiling(i/14)]), 1-
B_base_temp[ceiling(i/14)], by=0.01)))

  test_df <- SAM_sim(A_base = A_base[i], B_base = B_base[i], coord = coord[i], N_utt =
N_utt,
                    WC = WC[[i]], A_WC = A_WC[[i]])



  SAM_out <- SAM(test_df)
  DNM <- DNM_coordination(b = test_df$B, a = test_df$A)

  #Testing
  a_mean_sv <- mean(test_df$A > 0)

  temp <- c(A_base[i], B_base[i], coord[i], sum(test_df$A) / sum(test_df$A_WC),
```

```
        sum(test_df$B)/sum(test_df$B_WC), SAM_out$Align, SAM_out$C_base,
SAM_out$C_align, DNM$LSM, a_mean_sv)
 # names(temp) <- c("A_base_param", "B_base_param", "coord_param", "A_base_estimate",
   #             "B_base_estimate", "Align", "C_base", "C_align", "DNM_coord")

  df_4[i,] <- temp


}

stop_time <- Sys.time()

# For n = 100: 0.89044 sec.
# For n = 25000: 12.51 mins.
print(stop_time - start_time)
#write.csv(df_3, "./data_gen/df_3.csv")
```

Now, the data frame is transformed to establish the internal consistency.

```{r}
df_4_turned <- as.data.frame(matrix(ncol = 29))


for (i in 1:(n/14)) {
  df_4_turned[i,1] <- df_4$coord_param[i * 14]
  df_4_turned[i,2:ncol(df_4_turned)] <- c(df_4$Align[((i-1)*14+1):((i-1)*14+14)],
                                          df_4$DNM_coord[((i-1)*14+1):((i-1)*14+14)])

}

df_4_turned[,ncol(df_4_turned)+1] <- rowMeans(df_4_turned[,2:15], na.rm = T)
df_4_turned[,ncol(df_4_turned)+1] <- rowMeans(df_4_turned[,16:29], na.rm = T)
```



```{r}
# SAM Reliability
splitHalf(df_4_turned[,2:15], check.keys = F)
```

```{r}
#DNM Reliability
splitHalf(df_4_turned[,16:29], check.keys = F)
```

```{r}
fit_SAM <- lm(df_4_turned[,1] ~ df_4_turned[,30])
summary(fit_SAM)
```

```{r}
fit_DNM <- lm(df_4_turned[,1] ~ df_4_turned[,31])
summary(fit_DNM)
```

When using random coordination parameters, the SAM's internal consistency is, as expected, approaching zero. The measures of the DNM, however, still show a internal consistency of ~ .3 . This cannot be explained by the DNM's ability to estimate coordination as there is no underlying coordination value. Accordingly, this internal consistency can attributed to and only to the DNM method itself. As such, it does not relate to the construct to be measured. A possible explanation is that the scores are all somewhat similiar due to the bias (depending on the A baseline) described before. Moreover, the fact that the DNM does not take into account the word count of the utterances might contribute to this issue.

It can be concluded that the DNM's internal consistency is over-estimated when there is a lack of an underlying construct as coordination. Potentially, this over-estimation also takes place when there is an underlying construct.

The underlying coordination is equal to zero.

```{r echo=FALSE}
# Has to be divisble by 14!
n <- 4200
```

```{r echo=TRUE, warning=FALSE}
start_time <- Sys.time()
df_5 <- data.frame(A_base_param = vector(mode = "numeric",length = n),
                B_base_param = vector(mode = "numeric",length = n),
                coord_param = vector(mode = "numeric",length = n),
                A_base_estimate = vector(mode = "numeric",length = n),
                B_base_estimate = vector(mode = "numeric",length = n),
                Align = vector(mode = "numeric",length = n),
                C_base = vector(mode = "numeric",length = n),
                C_align = vector(mode = "numeric",length = n),
                DNM_coord = vector(mode = "numeric", length = n),
                A_mean = vector(mode = "numeric", length = n))


# Setting up parameter vectors.

# The baselines are assumed to be different in each run: A person can have two
# different baselines for two different marker categories at the same time.

# Based on prior findings, the baselines are between 0.001 and 0.2, a range
# only a little wider than the actually observed range.
A_base <- base::sample(seq(0.001, 0.2, by=0.001), replace=T, size = n)
B_base <- base::sample(seq(0.001,  0.2, by=0.001), replace=T, size = n)
```

```
# Getting Word counts (i.e. utterance lengths). As we simultate the
# measurement of 14 categories per utterance, the word count repeats
# itself 14 times.
N_utt <- 5
N_len_decay = 0.05
WC_temp <- list()
A_WC_temp <- list()
for(i in 1:(n/14)){
  WC_temp[[i]] <- ceiling(rexp(N_utt, N_len_decay))
  A_WC_temp[[i]] <- ceiling(rexp(N_utt, N_len_decay))
}

WC <- list()
A_WC <- list()
for (i in 1:n) {
  WC[[i]] <- WC_temp[[ceiling(i/14)]]
  A_WC[[i]] <- A_WC_temp[[ceiling((i/14))]]
}




for (i in 1:n) {
  temp <- vector(length = 8)

  #coord <- rep(base::sample(seq(-(B_base_temp[ceiling(i/14)]), 1-
B_base_temp[ceiling(i/14)], by=0.01)))

  test_df <- SAM_sim(A_base = A_base[i], B_base = B_base[i], coord = 0, N_utt = N_utt,
                     WC = WC[[i]], A_WC = A_WC[[i]])



  SAM_out <- SAM(test_df)
  DNM <- DNM_coordination(b = test_df$B, a = test_df$A)

  #Testing
  a_mean_sv <- mean(test_df$A > 0)

  temp <- c(A_base[i], B_base[i], coord[i], sum(test_df$A) / sum(test_df$A_WC),
          sum(test_df$B)/sum(test_df$B_WC), SAM_out$Align, SAM_out$C_base,
SAM_out$C_align, DNM$LSM, a_mean_sv)
 # names(temp) <- c("A_base_param", "B_base_param", "coord_param", "A_base_estimate",
   #               "B_base_estimate", "Align", "C_base", "C_align", "DNM_coord")

  df_5[i,] <- temp
```

}

stop_time <- Sys.time()

# For n = 100: 0.89044 sec.
# For n = 25000: 12.51 mins.
print(stop_time - start_time)
#write.csv(df_3, "./data_gen/df_3.csv")
```

Now, the data frame is transformed to establish the internal consistency.

```{r}
df_5_turned <- as.data.frame(matrix(ncol = 29))


for (i in 1:(n/14)) {
  df_5_turned[i,1] <- df_5$coord_param[i * 14]
  df_5_turned[i,2:ncol(df_5_turned)] <- c(df_5$Align[((i-1)*14+1):((i-1)*14+14)],
                                          df_5$DNM_coord[((i-1)*14+1):((i-1)*14+14)])

}

df_5_turned[,ncol(df_5_turned)+1] <- rowMeans(df_5_turned[,2:15], na.rm = T)
df_5_turned[,ncol(df_5_turned)+1] <- rowMeans(df_5_turned[,16:29], na.rm = T)
```



```{r}
splitHalf(df_5_turned[,2:15], check.keys = F)
```

```{r}
splitHalf(df_5_turned[,16:29], check.keys = F)
```

```{r}
fit_SAM <- lm(df_5_turned[,1] ~ df_5_turned[,30])
summary(fit_SAM)
```

```{r}
fit_DNM <- lm(df_5_turned[,1] ~ df_5_turned[,31])
summary(fit_DNM)
```

When the underlying coord parameter is zero, the internal consistency of both the SAM and the DNM are low. The resulting scores are more or less normally distributed.

```{r}

```
hist(df_5_turned$V30, main = "Histogram of the Coordination Estimates \nusing SAM when Coordination is Zero")
```