**Zurich University of Applied Sciences**

Centre for Artificial Intelligence

MASTER THESIS

# Transcript-Based Action Item Detection in Multiparty Conversational Speech

*Author:*
Flurin Gishamer

*Supervisor:*
Prof. Dr. Mark Cieliebak

# Abstract

In this paper, we propose a novel approach for detecting action items in transcripts of multiparty conversational speech, specifically in meetings. The motivation for this work stems from the need to efficiently process and structure the information exchanged during meetings, which can be aided through automatic meeting-minute generation, of which automatic action item detection is an essential part. However, current solutions for action item detection are lacking in practical usability.

Our main contributions include the identification of linguistic features characteristic of action items based on existing annotations, leading to the development of a concept of indicativity for action items in terms of utterances or speaker turns. Using this concept of indicativity, we created a new corpus consisting of annotations for the transcripts of the ICSI and ISL corpus, which we make openly available. We then examine the sequential ordering of speaker turns to determine if a speaker turn can be assigned to an action item based on its position.

Experiments were conducted using transformer-based machine learning models on our newly created annotations, resulting in significant improvements in precision, recall, and F1-score compared to existing annotations. Our analysis also shows that, based on sequential properties of speaker turns, it is possible to assign them to an action item, however, further work is needed to improve performance in this regard.

In summary, this paper presents a thorough study on transcript-based action item detection and makes a valuable contribution towards the development of a system for automatic meeting-minute generation through the identification of linguistic features, the creation of a new corpus, and examination of sequential properties of speaker turns that belong to an action item.

# Acknowledgements

I would like to express my heartfelt gratitude to all the individuals who have supported me during my journey of completing this master thesis.

Firstly, I would like to thank my supervisor, Prof. Dr. Mark Cieliebak, for his invaluable guidance, support, and mentorship. He has taught me most of what I know about NLP and his expertise has been instrumental in my research.

I am also grateful to Dr. Jan Milan Deriu and Pius von Däniken for providing me with valuable advice and insights throughout my research.

I would like to extend my sincere thanks to the Centre of Artificial Intelligence at ZHAW for providing me with a great learning environment and opportunities to grow as a data-scientist.

I am also grateful to Christoph Bräunlich, my supervisor at work and the head of AI, who has supported me and allowed me to learn a lot during this journey. I would also like to thank Kurt Wieland for his technical support.

I would like to express my deep appreciation to my partner, Mina Hamie, who always supported me and was extremely patient throughout the process. Lastly, I would like to thank my mother, Eva Gishamer, for being there for me throughout this journey and for her love and support.

This thesis would not have been possible without the help and support of all these individuals, and I am truly grateful for all their contributions. Thank you!

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Overview

Meetings are a frequently used means of exchanging information in companies. In [1], Keith states that in 2022, between 62 and 80 million meetings per day were held in the U.S. alone. A significant influence on the increase in meetings in recent years might have been the Covid pandemic, with DeFilippis stating that the number of meetings per person increased by 12.9% in that period alone [2]. An essential part of ensuring the effectiveness of meetings are meeting minutes. They are often an essential part of the outcome of meetings for several reasons: Firstly, they create consensus on the exact scope of the discussion. Secondly, they help to document and assign tasks the participants have agreed upon during the meeting, and thirdly they serve as a means of providing information to people who did not attend the meeting.

To prepare for meetings, one or more participants create a written agenda, using it to determine the topics to be discussed and to provide the general structure of the meeting. The counterpart to the agenda is the meeting minute. Typically one of the meeting members is responsible for writing the meeting minute. However, who is responsible for writing the meeting minute can vary. For example, there may be one person in the meeting whose only task is to write the minute or a participant who writes the minute as an additional task. In the second case, the participant can no longer give his full attention to the meeting. Especially if this person is vital to discuss the topics, it can negatively impact the quality of the meeting. A meeting minute can be structured in a variety of ways but often contains the following elements:

- Date and time of the meeting
- Names of attendees
- Topics discussed
- Decisions agreed upon during the meeting
- Action items

The last point in the list, called action items, is a section about the tasks to be carried out and their deadlines, as well as to whom those tasks are assigned, i.e., a responsible individual. Typically the responsible individual belongs to the attendees of the

meeting. Action items can either be informal, such as when a participant says they will send someone an email with additional information, or official, such as when the group assigns a participant responsibility for a task that involves legally binding steps (e.g., placing orders). Therefore action items are an integral part of meeting minutes because, in the case of informal commitments, the participants can use the meeting minutes to remember what they have confirmed to the other participants. In the case of official obligations, the action items in written form can also have a legally binding character.

## 1.2 Motivation

Often the creation of minutes requires the special attention of one person, which leads to that person not being able to devote themselves entirely to the discussion, resulting in the loss of valuable resources. In addition, consistently recording meeting content according to a predefined structure is challenging. These factors lead to a situation where stakeholders often neglect meeting minutes, which in turn can cause participants to discuss specifics again afterward, informing people who could not attend the meeting. There is often no consensus on the tasks and their execution discussed in the meeting. All this takes considerable time, meaning a significant loss of resources.

With the increase in virtual meetings, but more generally with the availability of computers in most meetings, it has become easy to record meetings as audio files. In addition, since ASR systems (automatic speech recognition) are available in many systems at no additional cost, one can easily create transcripts of meetings. The growing popularity of LLM (large language models), which are mainly based on the Transformer architecture of Vaswani et al. [3], has shown impressive results in the areas of natural language processing (NLP) and natural language understanding (NLU). Therefore, the use of such models to automatically generate meeting minutes seems obvious. However, one obstacle to automatic processing is that spontaneous speech differs fundamentally from written language. It contains idiosyncrasies such as repetitions, interruptions, and more frequent grammatical errors than written language. These idiosyncrasies require the development of models specialized in the processing of spontaneous speech. In this thesis, we focus on one aspect of meeting minutes, namely, action items. Our goal is to detect action items in transcripts of meetings automatically. We formulate this task as a binary classification problem. The current approach to solving such tasks defines them as supervised-learning problems. This procedure, in turn, requires many data points, each of which assigns a label to a concrete example (in the case of meetings, they are either called utterances or speaker turns), whether it is part of an action item or not. Unfortunately, there are few corpora of natural, non-scripted meetings and action item annotations in English.

## 1.3 Contribution

Our goal is to improve the automatic recognition of action items, as we believe this is an important step towards systems that can automatically generate meeting minutes based on transcripts. To this end, we make the following contributions:

1. Identification of the main linguistic features characteristic of action items based on available annotations.

2. Development of the concept of indicativity for action items in terms of utterances or speaker turns.

3. Creation of a new corpus consisting of annotations based on indicativity, which we make openly available at `https://github.com/gishamer/indicat`.

4. Examination of the sequential ordering of speaker turns to determine if a speaker turn can be assigned to an action item based on its position.

## 1.4 Related Work

Previous work in the area of action item detection has focused chiefly on emails or dialogue: For emails, the goal is to detect action items in written text as described in [4], [5], [6] and [7]. The goal in the case of dialogues is to detect action items using transcripts from meeting recordings, which is what this thesis investigates. In [8], Gruenstein et al. introduce a set of annotations for the ICSI corpus [9] covering topic segmentation as well as action items. Subsequent work used those annotations to perform action item classification. However, we do not report results based on the Gruenstein annotations but use them to identify the most significant features for action items, among other annotations.

Morgan et al. use in [10] a subset of the annotations presented by Gruenstein, consisting of 15 transcripts. They use a combination of lexical, conceptual, syntactic, prosodic, temporal, and semantic (TIMEX) features in conjunction with a maximum entropy model. They give an F1-score of 25.62 when no prosodic features or dialogue acts are used. We, too, found features that indicate the presence of dates and time to be useful, but instead of TIMEX used NER.

Purver et al. report in [11] that they use a subset of the annotations created by Gruenstein (6 ICSI and one ISL meeting) but use transcripts created as part of the CALO project [12] to evaluate their results, which are not publicly available. The hierarchical annotation scheme for action items introduced in their work is of great relevance, far more than the actual classification results achieved. Consequently, this hierarchical annotation scheme is used in much of the subsequent work cited. We refer to this in the following as the AIDA annotation scheme, i.e., AIDA classes.

In [13], Purver et al. apply the AIDA annotation scheme they introduced in [11] for action items to 18 ICSI meetings. We will refer to this set of annotations as the AIDA annotations. They used 4 SVMs with linear kernels to consider the 4 different AIDA classes as a binary classification problem each (We do refrain from this approach due to the relative sparsity of the annotations). They then use a super-classifier that identifies individual time windows containing action items based on the 4 binary

classifiers. In addition to lexical features, they use prosodic features, TIMEX, contextual features (up to 5 preceding utterances), and MRDA dialogue acts from [14]. Finally, they report F1-scores for the AIDA classes of 0.15 for Description, 0.14 for Timeframe, 0.24 for Owner, and 0.17 for Agreement.

Yang et al. [15] examine the relation between the MRDA dialogue acts and the AIDA action item annotations from Purver. They conclude that there is a strong connection between the group of dialogue acts referred to as action motivators and action items, which we also found during the evaluation and re-annotation of the ICSI corpus. They underline this by showing that they can achieve F1-scores of 0.21 for the Description class of the AIDA annotations when only using dialogue acts. Moreover, apart from dialogue acts, they report using only lexical features, i.e., word unigrams and bigrams.

In [16], Frampton et al. simulate a scenario in which participants in a meeting press a button to indicate that an action item is being discussed. Their approach significantly narrows the utterances to be considered, which also explains the relatively high F1-scores of 0.48-0.60. Frampton et al. also use the AIDA annotations introduced by Purver et al. in [13] and an SVM with a linear kernel.

Murray et al. [17] perform action item detection on the AMI corpus [18], using prosodic, structural, speaker, length (duration of utterances), and lexical features. They employ a logistic regression classifier. Due to the use of the AMI corpus, their results are not directly comparable to ours.

Sachdeva et al. [19] apply transformer-based models to the action item detection task using the AIDA annotations from Purver and the action item annotations available for the AMI corpus. They obtained state-of-the-art results for the binary classification task (whether an utterance belongs to an action item or not), with an F1-score of 0.39, i.e., 0.42. F1-scores for the AIDA classes are 0.27 for Description, 0.23 for Timeframe, 0.37 for Owner, and 0.31 for Agreement, and are higher than any other published results to date. Unlike Sachdeva et al., we did not use the action item annotations included in the AMI corpus because the class imbalance in the AMI corpus is even more pronounced than is the case with the ICSI corpus.

Hsueh et al. [20] deal with detecting decisions in meetings without specifically addressing action items. Instead, they use lexical, prosodic, contextual, and topical features from the AMI Corpus [18]. Training a maximum entropy model, they classify segments of the dialog. Fernandez et al. [21] also deal with detecting decisions and use the AMI corpus for this task. Still, in contrast to Hsueh, they propose a scheme similar to the AIDA annotations scheme proposed by Purver in [13], where an utterance can belong to one of five classes.

*Actionable items* are a concept related to action items. Chen et al. [22] created a set of annotations for the ICSI corpus, focusing on actionable items that an automated meeting assistant could process. They also assigned annotations to different classes, some matching our definition for action items, allowing us to use a subset of them. Finally, they trained convolutional deep-structured semantic models to obtain vector embeddings and find the utterances closest to a given actionable item.

# Chapter 2

# Theory

This chapter presents foundational concepts critical for comprehending this thesis. Initially, we delve into the notion of dialogue acts, encompassing the various methodologies for segmenting utterances. Subsequently, we formalize the definition of action items. Building upon this, we introduce our novel concept of indicativity. Additionally, we provide a succinct overview of the weak supervision and SHAP techniques, which play a pivotal role in our approach.

## 2.1 Dialogue Acts

Dialogue acts are a concept used to describe the intent or purpose of a statement in a dialogue and are similar to speech acts [23]. They are an essential means to model discourse structure and, as explained in [23], an important first step to understanding spontaneous speech.

### 2.1.1 Segmentation

A prerequisite to dialogue act labeling is the identification of utterance boundaries. Here we want to emphasize that there are multiple ways to define utterances. One way is to define an utterance as equivalent to a speaker turn. Another way is to use a more fine-grained definition, which Stolcke et al. [23] refer to as sentence-level based. In the following, we describe how to segment spontaneous speech according to the sentence-level based definition:

#### Utterances

Dhillon et al. define an utterance as "a segment of speech occupying one line in the transcript by a single speaker which is prosodically and/or syntactically significant within the conversational context" [24]. Assuming this definition, the main factors to segment an utterance are, according to Dhillon et al., syntax, pragmatic function, and prosody (which we use again to perform the actual classification of utterances to dialogue acts). On a syntactical level, conjunctions and parentheticals (expressions not essential to the meaning of a sentence and separated by punctuation marks or brackets) often yield cues as to whether we should segment a given text into further utterances. On the level of pragmatic function, we can consider an utterance as complete as long as it has a unique function within the discourse, where function in this context refers to dialogue acts. Dhillon et al. point out that grammatically incomplete phrases can still be considered complete utterances under this condition.

In summary, as soon as a sequence of words forms a complete dialogue act, we can consider it an utterance.

#### Speaker Turns

Unlike utterances, speaker turns denote the sum of a speaker's statements independent of their syntax or pragmatic function. A speaker change determines the boundaries of a speaker turn. It follows that a speaker turn consists of one or more utterances, meaning that segmenting by speaker turns leaves less room for interpretation and can be automated by so-called speaker diarization models when applying automatic speech-to-text systems.

#### Implications

When we say an utterance is equivalent to a speaker turn, in many cases, the resulting segments are longer than when using the sentence-level-based definition. Since speaker turns can span multiple sentence-level-based utterances, they also often have more than one function within the dialogue, requiring multiple dialogue acts to be assigned to them in such cases. The authors of the MRDA corpus, which consists of dialogue act annotations, also used this finer-grained segmentation unit. When doing so, speaker turns can differ from utterances in 2 ways: As already mentioned, a speaker turn can comprise multiple utterances. An example is when a speaker begins by answering a question and then goes over to suggest a new task. Here, we could say that this speaker turn comprises two utterances, each corresponding to 1 dialogue act. As noted by Stolcke et al. [23], speaker turns can also consist of an incomplete utterance, for instance, when speaker B utters a backchannel (a dialogue act that can signal agreement such as "mhmm") and thereby segments speaker A's utterance.

### 2.1.2 Dialogue Act Classification

Existing work in this area [23] [25] [24] uses categories to group the different dialogue acts. Dialogue acts are described in terms of tags that can be assigned to an utterance. Stolcke [23] states that the decision to which dialogue act an utterance is assigned is made based on three criteria:

1. **Syntax:** As explained by Yule, semantics is concerned with the rules by which we structure language [26, p. 86]. In our case, this refers to the specific structure of utterances, which often differ significantly from written language.

2. **Semantics:** Semantic function refers to the meaning of the words, regardless of the context of the conversation. According to Yule, it is a technical approach "... concerned with objective or general meaning and avoids trying to account for subjective or local meaning" [26, p. 100]

3. **Pragmatics:** Refers to what a speaker meant by her utterance, or as explained by Yule [26, p. 112], how a listener infers the meaning of an utterance based on shared assumptions and expectations.

The DAMSL annotation scheme of Core and Allen [25] is a widely used dialog tag set, which, in slightly modified form, is also used in the MRDA corpus of Shriberg et al. [14], which comprises annotations for the ICSI corpus.

In the following, we will focus on the MRDA tag set since Shriberg et al. [14] developed it to annotate the ICSI corpus. We also use the MRDA annotations in this thesis for analysis as well as in the experiments. After we have described this tag set, we will also relate the most relevant tags of the MRDA tag set to those of the DAMSL tag set. The MRDA tag set consists of 13 groups, and we will limit ourselves here to describing the three most important groups (the remaining groups can be consulted in [24]):

- **Backchannels and Acknowledgments:** As Dhillon et al. explain, The speaker who has the floor does not utter backchannels, but other speakers in the background to signal that they follow along with what the speaker who currently has the floor is saying. Speakers uttering Backchannels do not directly address a person. Backchannels usually take the form of "okay", "right", "yeah", and "sure", among others. We mention backchannels because it is in their nature to segment a speaker turn. For example, when speaker A says something, and speaker B and speaker C utter backchannels, they segment the semantically coherent utterance of speaker A into several speaker turns. The easiest way to counteract this is, of course, to remove utterances annotated as backchannels. However, when applying an action item detection system in real life, i.e., when using it on meeting recordings, we have to consider that utterances signaling agreement often use the same vocabulary as backchannels, which we cannot distinguish via dialogue act annotations in this case.

- **Responses:** The group Response is divided into 3 subcategories, namely: *postive*, *negative*, and *uncertain*. In the *positive* subcategory, there is a dialogue act called *accept*. An important feature of action items is some form of acceptance by other meeting members. To identify this form of agreement in a dialogue, we can use the dialogue act type *accept*.

- **Action Motivators:** The action items belonging to this category are particularly interesting in connection with action items because they refer to future actions. The dialogue acts themselves do not specify an exact time frame. On the other hand, action items refer to a concrete point in the future, which is why all action items contain statements of the dialogue act category action motivator. However, not all dialogue acts of type action motivator indicate an action item. The group of these dialogue acts includes:

  - **Command:** Utterances of dialogue act type *command* can appear in two different variants. In the first variant, we formulate commands as statements, e.g., "Go get me coffee"; in the second variant, we formulate commands as questions, e.g., "Would you like to go get me a coffee?". Dhillon et al. [24] highlight that the distinction between *command* and *suggestion* can easily be confused and go on to suggest that assignment to one of the two categories can be inferred based on the interpretation of the response. They say that if the person responding to the question denies the utterance, and the person asking would most likely perceive it as impolite, it is a command. It is a suggestion whenever the person asking the question would not consider a denial impolite. Staying with the previous sentence, an example of a suggestion would be, "If you want, I can get you a coffee." In a real-world setting, it is also important to note that a person's role in a meeting can be critical in determining whether the other participants

interpret an utterance as a *suggestion* or a *command*.

- **Suggestion:** A dialogue act of type *suggestion* is used when, besides a *suggestion*, one wants to mark a proposal or advice. Dhillon et al. [24] explain that suggestions frequently can be identified from constructions such as "maybe we should". Especially in the context of this dialog act, it becomes clear that in many cases, an action item cannot be determined based on a single utterance because if utterances of other participants follow a suggestion that signal agreement and another participant signals commitment, an action item can emerge from the context, which we cannot assume based on the suggestion alone.

- **Commit:** Utterances with a dialogue act of type *commitment* indicate that the speaker of the utterance states that she will perform the future action that the group is currently discussing. Again, Dhillon et al. note that a *commitment* should not be confused with a *suggestion* since: "With commitments, a speaker mentions what he will do in the future, not what he might do." [24],

The action motivator category from the MRDA tag set we just described also has its counterpart in the DAMSL tag set. The DAMSL tag set has a category called "Forward-Looking Function", which describes utterances related to a target person's future actions. This category contains two aspects that can be related to action motivators, namely "Influencing-addressee-future-action (Influence-on-listener)" which refers to the act of influencing the actions of the target person, and "Committing-speaker-future-action (Influence-on-speaker)" which refers to a target persons commitment to perform an action. The category "Backward Looking Function" describes utterances related to the previous discourse. Within this category, there is the aspect of "agreement", which "codes how the current utterance unit affects what the participants believe they have agreed to, typically at the task level" [27]

### 2.1.3 Action Items

As pointed out by Purver et al. [11], we can define action items as group decisions made within a meeting. An action item needs to be assigned to a responsible individual who ensures the task discussed will be carried out. Purver et al. call this person the owner of the task. For a task to be a valid action item, the other group members must approve it, or in other words, it requires their agreement. Another requirement for an action item to be valid is that a meeting participant has assigned it to a specific point in time. Additionally, the task in question must be detailed enough to be carried out. From the above-stated requirements, it follows that we can characterize action items as entities that comprise four aspects, as defined by Purver et al.:

- **D**escription
- **O**wner
- **T**imeframe
- **A**greement

To summarize: action items can be defined as tasks to be carried out after a meeting ends, where a specific time in the future must be specified. At least one person must be responsible for the respective action, and the meeting participants must have agreed upon execution. If there is some form of team lead, that confirmation might come from a single person.

### 2.1.4 A Dialogue Act-Based Definition For Action Items

Action items are challenging to identify because they represent an overarching concept that usually emerges over multiple utterances or speaker turns. Therefore, dialogue acts are not sufficient for identifying action items since they constitute annotations on the utterance level and cannot capture an entire action item. However, on a structural level, an action item can be understood as a set of utterances comprising at least one dialogue act of the category *action motivators* i.e., either *command*, *suggestion* or *commit*. An additional condition is that at least one utterance must refer to a specific point in the future after the meeting ends, and the referred future action must be relevant to the group's goals.

This definition differs from Purver's proposed definition used in the AIDA annotations in that we have defined it using the MRDA dialogue act tag set. We think this facilitates the detection of action items, especially in the case of the ICSI corpus, which provides dialogue act annotations. This approach can also be applied to other transcripts using a dialogue act tagger. We can describe each utterance's function in a given action item by its associated dialogue act. The advantage of the DOTA scheme used in the AIDA annotations is that they are specifically adapted to action items. However, the number of available annotations is small, which is problematic in the context of LLMs (Large Language Models).

### 2.1.5 Action Item Recognition vs. Detection

Action item recognition identifies a given action item within a dialog. This activity identifies several utterances, which may be distributed over a long sequence, spanning potentially multiple speaker turns by an arbitrary number of speakers. Accordingly, they do not have to be adjacent, but they all have to belong to the same entity, namely the recognized action item. When we consider an action item as a well-defined entity, we can approach the assignment of utterances to a particular action item as a form of co-reference resolution.

In contrast, the task of action item detection represents a simplification compared to action item recognition since the goal is a binary decision, namely whether an action item is present. In this case, we must neither recognize the entity itself nor the function of an utterance within an action item.

### 2.1.6 Indicativity of Action Items

As the introduction explains, an action item usually spans several utterances made by different speakers. The literature usually discusses one of two approaches:

1. **Binary:** An utterance belongs to an action item or not.

2. **Multi class:** If an utterance belongs to an action item, it is assigned a category based on its function. Purver's AIDA scheme is an example of this.

Previous research took the same approach in both cases. Namely, every utterance that could somehow be considered to belong to an action item was annotated. If we use the binary approach, where one either classifies an utterance as belonging to an action item or not, the set of annotated utterances is highly heterogeneous, as it contains utterances with very different functions that we nevertheless assign all to the same category. To illustrate, a suggestion like "Someone should send John the reports next week", as well as a statement like "You mean the red forms?" or "Yes, exactly", as well as "Well, I will", are considered to belong to the same category. We can use the multi class approach to counteract heterogeneity, but this creates another problem: By using different classes, we make the problem of label sparsity worse since there are fewer examples per class. Finally, both approaches have to deal with the same problem: To return to the example from above: although the statement that John should be sent the reports can be considered as belonging to an action item, the following three statements can only be assigned through the context. Thus, there are many scenarios where someone can say "Yes, exactly" or "Okay, I'll do it" without referring to an action item. We suspect that this ambiguity is one of the reasons why previous approaches cannot handle the task as well as a human can. Below are two dialogue segments from the ICSI corpus, which illustrate that an utterance can belong to an action item without necessarily being indicative of itself:

| Speaker | Indicative | Utterance |
|---|---|---|
| Speaker A | no | Is Srini gonna be at the meeting tomorrow, do you know? |
| Speaker B | no | Quite possibly. Oh, oh, sorry. Sorry, Wednesday, yeah. |
| Speaker A | yes | Maybe we can ask him about it. |

TABLE 2.1: **Example 1:** Multiple utterances all related to an action item, yet only the last utterance is indicative of an action item, here the corresponding dialogue act is of type suggestion

| Speaker | Indicative | Utterance |
|---|---|---|
| Speaker A | no | you got a favorite belief-net that you've, you know, played with? JavaBayes or something? |
| Speaker B | no | No, not really. |
| Speaker A | yes | O_K. Well, anyway. f- Get one. |

TABLE 2.2: **Example 2:** Same as example 1, but here the corresponding dialogue act is of type command

Because of this ambiguity, we propose to divide the problem as follows: We determine features that are strong indicators of the presence of an action item and restrict ourselves to detecting them. When we only consider features that we consider to be indicative of an action item, we do not mark all utterances that somehow belong to an action item. Afterwards, we can search for utterances related to the same action item as the indicative utterances we have identified to obtain an overall picture of the action item. In this thesis, we want to concentrate on detecting indicative utterances.

We intuitively know that an utterance refers to an action item when the respective speaker suggests or even commands the other participants to perform an action in

the near future. The same is true when someone says that he will perform an action in the near future. Using the MRDA dialogue act tag set, we can formalize our intuition by requiring that an utterance that we consider indicative of the presence of an action item must hold one of the three dialogue act tags *command*, *suggestion*, or *commit*. All these tags, in turn, belong to the category of action motivators. For example, below is a sentence for each of the three dialogue act tags:

- **Commitment:** So I'll - I'll take a closer look at it.
- **Command:** Tell them about the free lunch.
- **Suggestion:** And why don't you also copy Jane on it?

To formalize the statement "in the near future", we can require that the speaker in question refer to a specific time or date after the meeting, indicated by the presence of a named entity of type *date*. In addition, since an action item must be assigned to a responsible person, we may require that one of the personal pronouns "I" or "you" occur in the context of the task to be performed (except the verb used is in imperative form) or that the speaker designate a specific person, which is we can recognize by the presence of the named entity "Person".

## 2.2 Weak Supervision

Weak supervision is an approach to help train supervised machine learning models in domains with little annotated training data. In this thesis, we use a framework called Snorkel [28]. We use the approach to reduce the number of segments to review manually but also to train a discriminative model. Below we describe the procedure to generate training data using Weak supervision, as presented in [28]:

1. **Labeling Functions**: The first step in a weak supervision setting is to write a set of so-called labeling functions $\lambda : \chi \rightarrow \mathcal{Y} \cup \{\emptyset\}$ (from [29]). Labeling functions encode domain knowledge with heuristics, such as regex expressions, NER, and keywords. The output of a labeling function can be one of three values:
   - *Positive*: We indicate that the presence of the feature increases the probability of belonging to the class.
   - *Negative*: Likewise, the sample does not belong to the class.
   - *Abstain*: With this, we say that the labeling function cannot make any statement about the label. For example, a segment can indicate an action item even if it does not contain a reference to a person. Nevertheless, the presence of a named entity of type person increases the probability.
2. **Generative Model**: Based on these labeling functions, we can train a generative model, i.e., as described in [29] a weak supervision estimator $P_\mu(Y|\lambda)$, with parameters $\mu$ consisting of source correlations and accuracies. Given the noisy labels $\lambda$ (computed by the label functions), the model will output a probabilistic label vector $\bar{Y}$. In other words: the model will predict confidence-weighted, i.e., weak labels based on the labeling function mentioned before.

3. **Discriminative Model**: Based on the outputs of the generative model, Ratner et al. [28] recommend training a discriminative model, which then should be able to generalize beyond the labeling functions outputs. They point out that such a model should ideally accept probabilistic labels but recommend using the class with the highest probability output.

## 2.3 SHAP

For simple models, such as linear regression or decision trees, we can analyze a model to understand its predictions. However, this becomes increasingly difficult for more complex models, such as the transformer-based ones used in this thesis. For this reason, Lundberg et al. [30] developed a framework named SHAP, which stands for SHapley Additive Explanations.

### 2.3.1 Shapley Values

Shapley values are a concept from cooperative game theory that deals with players and games. For example, suppose we assume a game with $n$ players cooperating to obtain some final payoff. We further assume that the payoff, when all players cooperate, is bigger than the sum of payoffs if each player played the game by herself, the surplus. However, this also means that when we try to split the total payoff fairly, we need a way that incorporates the generated surplus.

Shapley values are a way to distribute the total payoff fairly, taking into account the surplus. It does so by computing the expected marginal distribution, which we can explain as follows: "suppose the players enter a room in some order and that all $n!$ orderings of the players in $N$ are equally likely. Then $\phi_i(v)$ is the expected marginal contribution of player $i$ as she enters the room." [31]. The expected marginal distribution is shown in equation 2.1 as introduced by Shapley [32].

$$\phi_i(v) = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|!(|N| - |S| - 1)! \left[ v(S \cup \{i\}) - v(S) \right] \tag{2.1}$$

Where $N$ is the set of all players participating in the game $\phi_i(v)$ is the expected marginal contribution, and $v(S)$ can be seen as the worth of the coalition (a coalition being a subset of the players). So here the term $[v(S \cup \{i\}) - v(S)]$ is player $i$'s marginal contribution to coalitions S, meaning the value he adds to this coalition with Shapley values being a weighted sum over all possible configurations of a game for this marginal contribution.

### 2.3.2 From Shapley Values to SHAP

To apply Shapley values to machine learning, we can interpret the input features of a model as the players $N$, and the game as the prediction of that model given the inputs. The marginal contribution then gives us a measure of the importance of each feature.

A player joining the game is equivalent to a feature being present. We can model this by encoding each input feature as a binary variable, meaning the feature is either present or not. Lundberg et al. [30] represent this as simplified input $x'$. These simplified inputs map to the original input, in our case words, by a mapping function $h_x(x') = x$. To explain the predictions of our original model $f$, we use a simpler model, a so-called explanation model $g$, which Lundberg [30] define as "any approximation of the original model". For the explanation model, Lundeberg et al. use local methods which one can characterise by the property $g(z') \approx f(h_x(z'))$, whenever $z' \approx x'$ [30]. The explanation model is then defined in terms of an additive feature attribution method that attributes an effect $\phi_i$ to each feature, as seen in equation 2.2 from [30]

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2.2}$$

where $z' \in 0, 1^M$, M is the number of simplified input features.

SHAP works by assigning each feature, in our case, each word of a given segment or speaker turn, an importance value for a given prediction. According to Lundberg et al.: "SHAP (SHapley Additive exPlanation) values attribute to each feature the change in the expected model prediction when conditioning on that feature. They explain how to get from the base value $E[f(z)]$, which would be predicted if we did not know any features, to the current output $f(x)$." [30], as shown in Figure 2.1

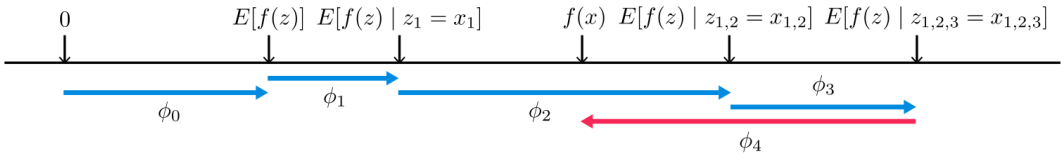


FIGURE 2.1: Change in the expected model prediction when conditioning on different features

# Chapter 3

# Methodology

In this section, we examine the datasets employed and the available annotations. Subsequently, the preprocessing procedures applied to the transcripts are detailed, with particular emphasis on the metrics that drove the preprocessing at the utterance level. Finally, a thorough re-evaluation and revision of the initial annotations are presented, resulting in the creation of a novel corpus grounded in our definition of indicativity.

## 3.1 A Remark About Plots and Color Maps

All the plots in this thesis use Viridis as provided by matplotlib [33]. Viridis is a sequential color map, meaning the lightness value increases monotonically. According to Moreland, "Sequential color maps are clearly appropriate for scientific visualization" [34]. Additionally, Viridis is perceptually uniform, meaning that the distance between two colors will be perceived as proportional to the distance between their associated scalar values on a given plot [34]. In the case of continuous color maps, perceptual uniformity as defined by Moreland [34] means that the quantity shown in Equation 3.1 rom [34] is constant for all $x$, where $c(x)$ denotes a color map that takes scalar value $x$ and returns color $a$:

$$\frac{\Delta E\{c(x), c(x + \Delta x)\}}{\Delta x} \tag{3.1}$$

Throughout this thesis, light colors mean better. Conversely, dark colors mean worse. We will ensure that this property holds for all plots, regardless of whether bigger is better or smaller is better.

## 3.2 Units of Segmentation

In this thesis, we use annotations from different sources. The authors of these annotations use different methods for segmenting the transcripts in the ICSI corpus. The ISL corpus again has a different way of segmenting the transcripts, which is why we describe the terminology we use below. Throughout this document, we refer to the following definitions whenever we use one of the words utterance, speaker turn, or segment.

### 3.2.1 Utterances

As we explained in section 2.1.1, Janin et al. segmented the transcripts of the ICSI corpus based on pragmatic, semantic, and syntactic function, a process explained in detail by Dhillon et al. in the official labeling guide [24]. This segmentation resulted in the utterances on top of which Janin et al. produced the annotations included in the MRDA corpus. Segmenting a transcript in this way requires expertise and is far more involved than simply detecting speaker changes. Therefore, when we talk about utterances, we refer to this concept. However, in our experiments, we do not use this segmentation unit.

### 3.2.2 Speaker Turns

Speaker turns are our main focus in this thesis, especially in the experiments, as they allow us to recreate a setup that most closely resembles real-world conditions. We mean a setup with automatic speech processing with speaker diarization. Whenever we mention speaker turns, we refer to the definition in Section 2.1.1

### 3.2.3 Segments

Segments are a peculiarity of the ICSI corpus. As described in more detail in section 3.4.1, they were created based on time intervals (so-called time bins) detected by a speech activity algorithm. We mention them here because, unfortunately, they are the only unit that allowed us to unify all annotations. The reason why both the Gruenstein annotations and the AIMU annotations use segmentations based on these time bins may be related to the fact that the initially published corpus contained XML files that used these time bins. However, it was not until the release of the MRDA corpus that utterance-based segmentations were available. As a result, only the annotations of Purver [11] are available for the utterance-based segmentation proposed in the MRDA corpus.

## 3.3 Special Considerations for Dialogue Transcripts

Corpora consisting of dialogue transcripts have some special properties that require attention. This section will explain those properties in detail and how we handled them. Dialogue or meeting transcripts consist of the text that consists of the actual message the respective speaker uttered, with additional metadata such as *speaker*, *start time*, *end time*, and *labels*. These transcripts are sequential, meaning the elements (in our case, either segments or speaker turns) have an ordering determined by their start and end times. To retain this ordering, we must pay special attention to operations that alter the number of elements within a transcript. If we, for instance, merge segments to obtain speaker turns, we must assign the resulting speaker turn the start time of the first segment and the end time of the last segment. Furthermore, each segment property, i.e., feature, must be handled differently during merge operations. For example, while for labels, we can say that the resulting speaker turn is positive as soon as one contained segment is positive, for other features such as NER, we need to retain the set of all NERs contained in the segments we want to merge. Whenever we wanted to compute statistics, we had to differentiate between two different cases:

1. **Computing statistics over multiple transcripts:** For instance, this was the case when counting the total number of segments in a split, the average length of a segment/speaker turn, etc. Here the transcript boundaries would not matter.

2. **Computing transcript aware statistics**: Meaning statistics where the transcript boundaries would matter, for instance, the mean number of annotations per transcript or the mean number of speakers.

For the first case, we could compute the statistics over all elements, whereas for the second case, the first step was to always group the elements by transcript and only then compute the respective numbers.

### 3.3.1 Impacts on Splits and Batching

We created the splits at the transcript level by considering the transcripts as atomic units, thus always assigning the entirety of a transcript to a split. This approach is critical when calculating past and future contexts for a segment, i.e., speaker turn, because here, we rely on the sequential order of utterances, which only has meaning in the context of a single transcript.

## 3.4 Datasets Used

This section gives an overview of all the data sets we used as source material. We then describe how, through substantial editing and re-annotation, we generated a new corpus which we used as the basis for all following experiments. While the ICSI and the ISL corpus contain recordings of transcripts on their own, the MRDA corpus and the Gruenstein-corpus do not contain any additional recordings but instead use the transcripts from ICSI and ISL (MRDA only uses ICSI).

### 3.4.1 ICSI Meeting Corpus

The ICSI meeting corpus was recorded at the International Computer Science Institute (ICSI) in Berkeley between 2000 and 2002 [9]. It consists of recordings and transcripts of 75 natural meetings. Natural means that these meetings were not scripted and would have taken place regardless of whether the meeting had been recorded. The duration of the corpus totals approximately 72 hours, with an average of 6 participants per meeting and a total of 53 unique speakers, of whom 28 were native English speakers. Most of the recorded meetings are weekly group meetings. In [9], Janin et al. note that the "Meeting Recorder" and the "Robustness" meetings "have a significant number of speakers in common. [While] Others are mostly speaker disjoint". One peculiarity of the recordings is the read-out digit strings at the beginning and end of each meeting.
There are a total of 5 meeting categories, each with a specific code. Table 3.2 shows the code of each meeting along with a name and the number of meetings belonging to that category. While the first three categories are related to work in the field of NLP and speech recognition "Network Services & Applications" are not, and "Other one-time only meetings" consist of different topics.

#### Metadata

All transcripts are in XML format, with metadata directly embedded. A transcript consists of word-level transcriptions. The transcript also contains annotations for the specific features of spontaneous speech, including backchannels, interruptions, repetitions at the sentence and word level, and filled pauses. In addition, contextual information is included directly in the transcription (e.g., whether a person is whispering while saying something). Coughing, laughing, and other non-lexical acoustic events are all represented as XML tags.

#### Segments

Each transcript is divided into segments. Each segment represents a so-called time bin. Time bins resulted from a pre-segmentation using a speech-activity detection algorithm and subsequent manual adjustments, as explained by Dhillon et al. [24]. One condition of these adjustments was placing them between word boundaries to avoid truncation. In the describing the methods used for transcription and annotation, Edwards points out that this segmentation has less semantic than practical use in that it should simplify the process of transcription [35]. Accordingly, utterances or speaker turns can extend over several segments. If speaker A utters a backchannel during an utterance of speaker B, it will result in the semantically coherent utterance of speaker B being split into two segments.

| Start-Time | End-Time | Participant | Text |
|---|---|---|---|
| 90.460 | 92.148 | me011 | stuff before everyones here. |

TABLE 3.1: A segment in the ICSI-corpus has a start-time, end-time, a participant, and text

Table 3.1 shows the structure of a single segment in a transcript of the ICSI Corpus. The segment tag contains the speaker's name and the start and end time of the segment.

| Name | Code | Count |
|---|---|---|
| Even Deeper Understanding | Bed | 15 |
| Meeting Recorder | Bmr | 29 |
| Robustness | Bro | 23 |
| Network Services & Applications | Bns | 3 |
| Other one-time only meetings | varies | 5 |

TABLE 3.2: Meeting types recorded for the ICSI corpus

### 3.4.2 ICSI Meeting Recorder Dialog Act Corpus (MRDA)

The MRDA corpus from Shriberg et al. [14] uses 72 of the total 75 recorded meetings from the ICSI corpus. It contains over 180'000 hand-annotated dialogue act (DA) tags. These follow a DA tag set called the MRDA tag set, which was created specifically for this corpus based on the DAMSL schema [25]. A significant distinction from the original ICSI corpus is how the transcripts were segmented. While the original corpus used time bins, the MRDA corpus used segmentation based on discourse function, meaning a sophisticated definition of utterances, as described in 3.2, was used to perform the segmentation. The organization of the MRDA corpus differs

from the ICSI corpus in that the actual transcripts were divided into the following 3 categories:

- **Dialogue Acts:** Contain the DA annotations. Each annotation specifies the start and end word and the corresponding DAs, where one annotation or segment can contain several DAs.
- **Segments:** Contains information about the segmentation of the transcripts. The segment annotations are structured similarly to the DA annotations but contain segments with no associated DA tag.
- **Words:** Contains the actual text of a transcript. Each word is listed, with associated start and end times and an ID through which the DA and segment annotations reference them.

In addition, each of the 3 folders of the individual annotations contains a separate file for each speaker and meeting, i.e., if 4 people participated in a meeting, then there are 3 x 4 = 12 related files. Table 3.3 shows the structure of the DA annotation. The MRDA corpus is in XML format, and each dialogue act has one child, as shown in Table 3.3. Each child contains an href attribute, which in turn points to the corresponding words file and the start word and the end word of the dialogue act within this word file. Listing 3.1 shows such a child, here "Bdb001.A.words.xml" references the words file, "id(Bdb001.w.2,122)" is the id of the start word within the file "Bdb001.A.words.xml" and "id(Bdb001.w.2,134)" is the end word.

LISTING 3.1: Href attribute from a child element of a dialogue act in the MRDA-corpus

```
1    <nite:child href="Bdb001.A.words.xml#id(Bdb001.w.2,122)..id(Bdb001.w
     .2,134)"/>
```

| ID | Start-Time | End-Time | Type | Participant | ... |
|---|---|---|---|---|---|
| Bdb001.A.dialogueact243 | 501.277 | 501.757 | s^bk | mn017 | ... |

TABLE 3.3: A dialogue act in the MRDA corpus has an id, start-time, end-time, type, and participant, among other attributes

### 3.4.3 ISL Meeting Corpus

The ISL Meeting Corpus was recorded at the Interactive System Labs (ISL) of CMU, Pittsburgh, between 2000 and 2001. We found while examining the ISL corpus, that the 104 meetings reported in [36] is not the number of meetings present in the publicly available corpus, which is a subset labeled the "ISL Meeting Transcripts Part 1" [37]. The difference from the reported number implies that the authors recorded 104 meetings, of which they only released 18 to the public. Furthermore, in [37], they state that "The ISL Meeting Corpus Part 1 is a first subset", yet there has been no release of further subsets since May 21, 2004. The first subset consists of recordings and transcripts of 18 meetings in 5 different categories. Of those 18 meetings, only 2 were project meetings (only because "meetings" is the type of transcript most relevant to our thesis, meaning the majority of the corpus does not quite fit our domain of interest), 9 were moderated discussions, 1 open chatting, and 6 playing games. In [36], Burger et al. say that the meetings are natural in that they would have taken place regardless of the recording. The duration of the corpus totals approximately

10 hours, with an average of 5 participants per meeting and a total of 31 unique speakers, of whom 20 were native English speakers.
**Metadata:** In [38], Burger et al. explain that "The meetings were transcribed at the orthographic word level. In addition to words the transcriptions label spontaneous phenomena and dysfluencies". They used the VERBMOBIL-II format to annotate those phenomena, a system for the transliteration of spontaneous speech. VERBMOBIL-II uses symbols to describe the characteristic properties of spontaneous speech, such as interruptions and repetitions, and defines a notation for indicating human noises, filled pauses, and interjections, among other things.
**Segments:** In contrast to the ICSI corpus, the segments in the ISL corpus represent speaker turns. As explained in [38] speaker turns are ordered by their start time. Contrary to the ICSI corpus, an interfering turn, i.e., backchannel, does not lead to segmenting the preceding turn. The order of speaker turns is therefore determined by their start time.

| Speaker | Text | Time Stamp |
|---|---|---|
| CHR | yeah, &=breath I had to do that, too, actually . | 9550_11990 |

TABLE 3.4: A segment as used in the ISL Meeting Corpus

In Table 3.4, one can see how speaker turns are encoded in the ISL corpus. Each speaker turn occupies one line and starts with the speaker's name prefixed with an asterisk followed by the actual speaker turn. In addition, the text includes the transliterations encoded in VERBMOBIL-II. For example, the string "&=breath" shows how to indicate a breathing sound in the VERBMOBIL-II annotation schema. After the sentence, follows the start-time and end-time enclosed by two negative-acknowledge symbols, which are omitted in the Tables since they are control characters and therefore do not have a visual representation.

### 3.4.4 Gruenstein Action Item Annotations

In [8], Gruenstein et al. introduce a set of annotations of hierarchical topic segmentation and action item dialogues. They built their annotations on top of the ICSI and ISL corpus transcripts. Two undergrad students performed the annotation task. The resulting corpus comprises 65 meetings. Out of the 65 annotated transcripts, 49 are from the ICSI corpus and 16 from the ISL corpus. Of those 16 of the ISL corpus, only 6 contain action item annotations, totaling 55 transcripts with action item annotations. Gruenstein et al. state that the two annotators labeled 765 and 1076 segments belonging to the category action item [8]. Their objective, however, was to identify utterances that would belong to a discussion about action items, which does not imply that they are indicative of an action item. Our goal in this thesis is action item detection, focusing on identifying utterances indicative of an action item according to the definition in 2.1.6. Therefore, we did not use Gruenstein's annotations in our experiments.
**Metadata:** Listing 3.2 depicts the structure of annotations for a given action item, as defined in [8]. Each action item is assigned to a parent topic, specified in the "actionItem" tag in the "name" attribute. Also specified directly as an attribute is the name of the corresponding transcript. The "segment" tag specifies the individual segments which belong to an action item.

LISTING 3.2: Structure of the action item annotation from Gruenstein et al.

```
<actionItem name="Goals for the end of the year" color="-26164" annotator=
    "cgilbert" discourse="Bdb001">
    <segment channel="B" start="1864.9300537109375" />
    ...
    <segment channel="B" start="1879.77001953125" />
</actionItem>
```

The annotations only contain a segment's start time, which is problematic due to the different meanings of a segment in the ICSI and the ISL Corpus. While in the ICSI corpus, segments are merely time bins without any semantic meaning, in the ISL corpus, segments represent complete speaker turns. Therefore, to infer a specific text span from the annotations, One has to assume the end time of said annotation. If one chooses the end time of the respective segment, the results will differ semantically between segments of the ICSI corpus and segments of the ISL corpus.

### Inter Annotator Agreement

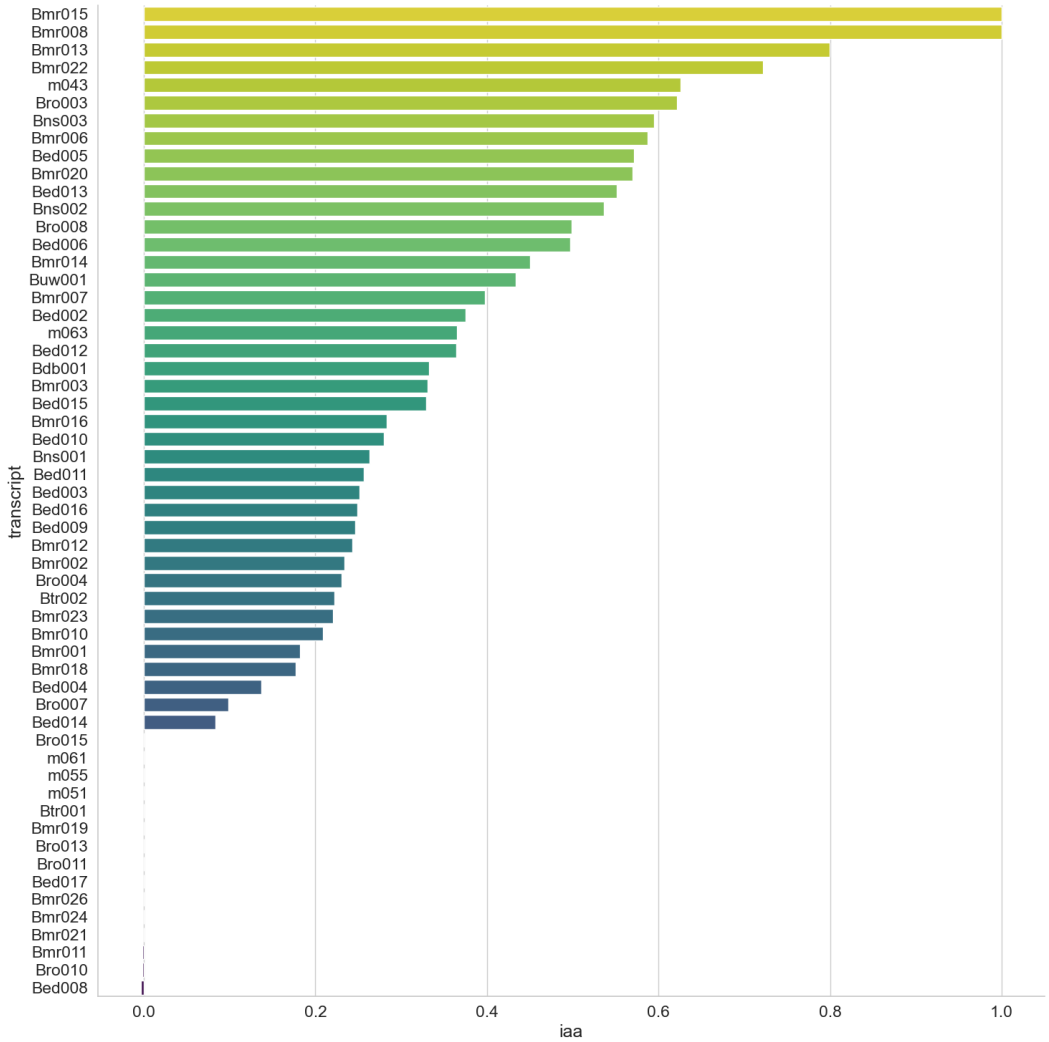


FIGURE 3.1: Inter Annotator Agreement on the transcripts containing action item annotations before preprocessing

In [8], Gruenstein et al. say that one annotator marked 1076 utterances as belonging to a discussion about action items, and 765 utterances were marked by the other. After parsing the Gruenstein corpus, attributing each segment annotation to an annotator, and calculating the number of annotations based on the data obtained by proceeding in this manner, we obtained the following numbers: There were a total of 1'791 segments annotated as belonging to an action item, of which an annotator with id "cgilbert" marked 1'267 and 929 by an annotator with id "michaeld". The Inter Annotator Agreement between the two annotators would vary quite substantially across transcripts, as can be seen in Figure 3.1, resulting in a mean of $\kappa = 0.293$, where $\kappa$ refers to the Kappa Statistic as shown in Equation 3.2 from Carletta [39]:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{3.2}$$

Where according to Carletta [39], $P(A)$ denotes the proportion of times the annotators agreed, i.e., chose the same label for an item, and $P(E)$ denotes the proportion of times the annotators would agree by chance. Artstein et al. provide an interesting perspective when they explain that $1 - P(E)$ is a measure of how much agreement "over and above chance is attainable" [40] and $P(A) - P(E)$, in turn, is a measure of "how much agreement beyond chance was actually found" [40].

### 3.4.5 AIDA Annotations

Purver et al. [11] created hierarchical action item annotations for the ICSI meeting corpus. They annotated 18 ICSI meeting transcripts, with 12 meetings belonging to the class "Even Deeper Understanding" 2 of type "Meeting Recorder", 2 of type "Network Services & Applications", 1 of type "Robustness" and 1 from "Other one-time only meetings". They say 3 authors annotated between 9 and 13 transcripts, with 3 meetings annotated by all authors and another 4 by 2 authors. There are 4 different AIDA classes, and one can assign each utterance to one or more AIDA classes. The 18 annotated transcripts contain 25'862 utterances, of which they labeled 1.4 %, i.e., 792 as belonging to an action item. The average number of annotations per transcript is 44. Table 3.5 shows the number of annotations per class. They report pairwise $\kappa$ values from 0.64 to 0.78 for the individual AIDA classes, which they point out is better than the inter-annotator agreement obtained in [8].

| AIDA Class | # Annotations |
|---|---|
| Description | 375 |
| Owner | 228 |
| Timeframe | 108 |
| Agreement | 328 |
| Total | 792 |

TABLE 3.5: Number of annotations per AIDA class

### 3.4.6 AIMU Actionable Items

The annotation structure proposed by Chen et al. assigns a domain to each actionable item: Calendar, Reminders, OnDevice, and Search. Each domain has actions, where actions consist of intent and argument. For example, the Calendar domain has the intents "find_calendar_entry", "create_calendar_entry", and so on. Arguments include: "contact_name", "start_date", and so on. From all the annotations, we decided to consider actionable items with action types "create_calendar_entry", "create_reminder", or "send_email" as action items, which resulted in 130 annotations of the 328 annotations the AIMU annotations comprise, Table 3.6 shows the number for the individual actions categories. The authors annotated Text passages they regarded as belonging to a domain using XML tags, where these tags can be the intent or the argument of an actionable item, which Listing 3.3 shows:

LISTING 3.3: Annotation scheme for the AIMU annotations

```
<create_single_reminder>
    I'll - I'll come back up
    <start_time>in about an hour</start_time>
    and
    <reminder_text>
        check and see if you're still meeting
    </reminder_text> .
</create_single_reminder>
```

| AIMU Action | # Annotations |
|---|---|
| create_single_reminder | 87 |
| send_email | 24 |
| create_calendar_entry | 16 |
| make_call | 3 |
| Rest | 198 |
| Total | 328 |

TABLE 3.6: Number of annotations per AIMU action

## 3.5 Creation of the Corpus

This section provides an overview of our methodology for constructing a corpus of indicative annotations. Our approach involves preprocessing the transcripts, aligning existing action item annotations, generating new annotations utilizing weak supervision and applying the concept of indicativity, creating speaker turns, and assigning transcripts to train, dev, and test datasets. This section presents the steps involved in building our corpus.

### 3.5.1 General Preprocessing

The transcripts of the ICSI corpus contain sections where the participants would read out digits. They recorded these sections to facilitate research on far-field acoustics without the added complexity of large vocabularies [41]. Since, for our task, we are only interested in discussions typical for meetings, those digit strings were of no use to us. Therefore, we removed all segments containing only read-out digits (an XML tag called "DigitTask" indicates when a segment only contains digits). The ICSI corpus contains annotations for non-vocal sounds such as "door slam" or "coughing". We believed these segments were not adding value to our use case and removed

them. A critical step we performed during preprocessing was to match the structure of the ICSI and ISL corpora. This step was necessary for the subsequent preprocessing stages and ultimately enabled the experiments to be performed. An important aspect when aligning the two corpora was that both use different conventions to represent transliterations. Therefore, the most efficient way to align the corpora was to remove transliterations altogether, which we did, using regular expressions in the case of the ISL corpus and flattening and removing nested XML structures in the case of the ICSI corpus. Another reason we removed transliterations was that we wanted to make our results as representative as possible of a practical setup, and we assumed that such handcrafted features would not be available in a real-life system for automated meeting minutes generation either. In contrast to removing transliterations, we have kept punctuation and capitalization unchanged as much as possible in both corpora.

### 3.5.2 Alignment with Annotations

The second step consisted of merging the various annotations with the segments of the transcripts. For the binary label of the Gruenstein, AIDA, and AIMU annotations, we proceeded in the same way: If a segment had an action item annotation, we set the label to 1, if the segment belonged to a transcript that had annotations, but the segment itself did not contain an action item annotation, we set it to 0, for utterances belonging to a transcript for which there are no annotations, we set the label to -1. Since the segmentation of ICSI, MRDA, AIDA, and AIMU, as well as the Gruenstein annotations, differ, we had to proceed in each of these cases slightly differently. Which we describe in the following:

#### MRDA

Firstly we want to point out that only a subset of our dataset contains dialogue act annotations since there are no such annotations for the ISL corpus. Therefore, for the remaining ICSI transcripts, we had to develop a procedure to assign DAs to the ICSI segments, i.e., speaker turns. We proceeded to assign a DA annotation to a speaker turn if either its start-time or its end-time was between the start-time and end-time of the respective speaker turn. This assignment leads to multiple dialogue acts being assigned to a single speaker turn under some circumstances. Here we would like to point out that this is a consequence of our approach to segmentation not being as fine-grained as the approach used in creating the MRDA corpus.

#### Gruenstein Annotations

In order to assign action item annotations of Gruenstein et al. [8] to the individual segments, we had to make some assumptions to determine a segment's end time. Therefore, we chose the most straightforward procedure: The end time of an action item was assumed to be the end time of the segment to which the action item annotation's start time referred.

#### AIDA Annotations

When merging the Purver annotations with the ICSI corpus, we chose a similar procedure to Gruenstein. We took a segment's end time as the corresponding annotation's end time.

#### AIMU Annotations

As described in section 3.4.6, the authors annotated text passages that they considered to belong to a domain using XML tags. To merge the annotations with our corpus, we determined whether the text of the segments contained an XML tag with one of the intents' create_calendar_entry', 'create_single_reminder', 'make_call', or 'send_email'. If so, we annotated the segment as a positive sample without using the annotated text, i.e., in our corpus, these XML tags no longer appear in the text. Instead, we added a column with binary annotations.

### 3.5.3 Annotating the Corpus

One of our goals in this thesis was to generate annotations that meet our definition of indicativity. To verify our assumptions on the one hand and to reduce the number of segments to be considered on the other hand, we determined the features that serve as good indicators of whether a segment is an action item or not as part of the data analysis (see section 4.3). Based on these features, we performed weak supervision as described below:

#### Weak Supervision

Using the snorkel framework [28], we wrote a set of labeling functions for the most important features. The following is a list of the used labeling functions:

- Keywords:
  1. send
  2. email
  3. tomorrow
  4. week
  5. pointer pages
  6. constructions
  7. web
  8. write
  9. meet
  10. group
  11. list
- Dialogue Acts (positive):
  1. commitment
  2. suggestion
  3. command

- Dialogue Acts (negative):
    1. continuer
    2. backchannel
    3. about-task
- Named Entities
    1. date
    2. person
    3. time
- Future Tense

Based on these label functions, we would then train a label model to generate weak labels as described in Section 2.2. From all segments Snorkel's label model would consider a positive sample, we selected the 8'000 segments with the highest confidence and marked them as weak labels.

### Selection Process

To create what we will refer to as indicative labels in the rest of this thesis, we first took the union over the Gruenstein, AIDA, and AIMU annotations and the 8'000 segments obtained by weak supervision, which resulted in a total of 9289 segments. We then went over those segments to apply our definition of indicativity to them. After manually considering all segments, we obtained 1908 segments that we consider indicative of the presence of an action item. In Figure

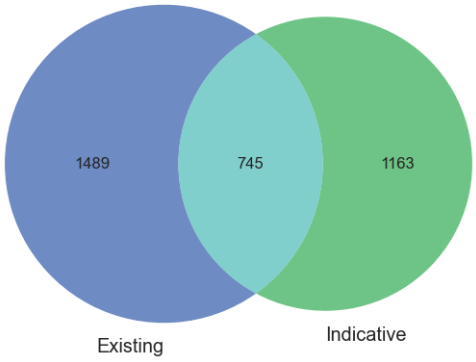


FIGURE 3.2: Overlap between the existing annotations and the re-annotated labels

When annotating the utterances, there were three different cases: first, there were utterances that had not been annotated as action items before but which we considered due to the weak label and that matched our definition of indicativity. Second, some utterances were annotated as action items in one of the annotation sets we used but did not fit our definition of indicativity. Therefore we did not select them, and third, there were utterances annotated as action items in one of the annotation sets that fit our definition of indicativity. In the following, we will contrast one negative and

one positive example from the AIDA and Gruenstein annotations to illustrate our approach.

**AIDA Description**

1. **Not Indicative:** "you know like two examples I mean, y-"
2. **Indicative:** "and, um, get a first draft of that."

Both utterances shown above are *descriptions* in Purver's AIDA annotation scheme. However, only the second utterance meets our definition of indicativity. We only consider the second utterance indicative because, although both utterances contain a partial description of a task to be executed, the first utterance is explanatory. In contrast, the second utterance is an order we can unambiguously assign to the dialogue act *command*. Furthermore, it refers to a future action that, with high certainty, refers to a specific time after the meeting.

**AIDA Owner**

1. **Not Indicative:** "O_K. So, first of all is, uh, do e- either of you guys,"
2. **Indicative:** "What I'm - what my job is, I will, um,"

Both utterances shown above are of type *owner* in Purver's AIDA annotation scheme. The first utterance is a question addressed to two persons but does not refer to any action. Therefore, we consider it to be non-indicative. On the other hand, the second utterance refers to the speaker herself and conveys that she will perform an action in the future. We can therefore assign this utterance to the dialog act *commitment*. Because of the dialog act and the content of the utterance, we consider it to be indicative of an action item.

**AIDA Time frame**

1. **Not Indicative:** And between now and then yeah."
2. **Indicative:** "Oh, I i- Yeah, I actually - Two is the earliest I can meet on Monday."

Both utterances above refer to a point in time, which is why they are both assigned to the AIDA class *time frame*. In the first utterance, it is clear from the context that a time in the future has already been discussed to which the speaker refers (then), but we cannot assume an action based on what the speaker said. The second utterance is an excellent example of how an utterance's pragmatic function implies more than can be inferred from the semantic function alone. To elaborate, while the interpretation on a purely semantic level only allows us to conclude that the speaker has time to meet at two o'clock, the pragmatic function indicates that the speaker suggests meeting on Saturday and no earlier than two o'clock. Suppose we interpret the utterance in this way. In that case, we can assign it to the dialogue act *suggestion* and say that it refers to an action at a specific time after the meeting, thus considering it indicative.

**AIDA Agreement**

1. **Not Indicative:** Yeah, something like that."

2. **Indicative:** "Perfect. Can you also write it up?"

Both utterances signal that the speaker agrees with the previous one, which is why both are assigned to the AIDA class *agreement*. However, the first utterance only indicates agreement. Beyond that, we cannot conclude an action, so this utterance does not meet our definition of indicativity. The second utterance is an excellent example of a command formulated as a question. Here, in contrast to the negative example for *owner*, the question refers to an action in the future. The utterance starts with agreeing to whatever has been said before and then continues to order someone to "write it up", which we can assign to the dialogue act *command*. The order to perform an action in the future, as posed in this utterance, matches our definition of indicativity.

**Gruenstein**

1. **Not Indicative:** "Just observable nodes, evidence nodes?"

2. **Indicative:** "So I would say you guy- the first task for you two guys is to um, pick a package."

Both of these statements come from the Gruenstein annotations. The first utterance clearly shows how these annotations differ from our definition of indicativity. Namely, something is being talked about (observable nodes), which belongs to a discussion about action items. In this case, there is a contextual connection between this utterance and an action item. However, the utterance itself does not suggest a concrete action and thus does not meet our definition of indicativity. In the second utterance, the speaker addresses two participants of the meeting, instructing them to perform an action (pick a package). In addition, we can recognize from the phrasing of the utterance that it has a dialogue act of type *command*. Because of these properties, this utterance satisfies our definition of indicative.

### 3.5.4 Obtaining Speaker Turns

The segments in the ICSI corpus are neither speaker turns nor semantically self-contained, as explained in section 3.4.1, whereas, in the ISL corpus, they represent speaker turns. One of our goals was to perform evaluations that closely correspond to a real-life setup. Since there are neither models that segment discourse into utterances nor the segments as offered by the ICSI corpus, we believe the best way to perform this evaluation is to use the ICSI transcripts to generate speaker turns. To obtain speaker turns from the ICSI transcript, we merged the segments as described below:

1. We removed backchannels, meaning all segments annotated with a dialogue act tag of either *b* (for backchannel) or *bk* (for acknowledgment), which are mostly of the form "Oh, O_K." or "Mm-hmm.". We did this because our subjective experience with current speech-to-text systems shows that such utterances are not transcribed in the first place or filtered out by successive cleaning steps.

2. We then took the remaining segments and joined all consecutive segments with the same speaker. This approach was another reason we removed backchannels in step 1, as they often fragmented otherwise coherent statements.

3. For speaker turn, we used the start time of the first segment and the end time of the last segment. In addition, we labeled a speaker turn as an action item if at least one of the merged segments already contained a positive label.

The initial number of segments from the combined transcripts of the ICSI and ISL corpus contained 122'606 segments. The removal of backchannels reduced that number to 106'543. Merging the segments into speaker turns reduced the number again to 51'349. The number of speaker turns marked as action items are 1562.

### 3.5.5 Data Splits

We use the splits proposed by Lee et al. [42], mainly because Sachdeva et al. [19] also use these splits, and by adapting them, we will allow comparability between the results. The splits refer to whole transcripts, i.e., a split is defined as a list of transcripts and includes all utterances of the transcripts in question. The splits apply only to the transcripts of the ICSI corpus. We assigned all transcripts of the ISL corpus to the training set. They were used for the experiments with Purver's annotations as well as for the experiments using our annotations.

## 3.6 Architecture

This section describes the loss function, including our approach to class imbalance. Additionally we describe our approach to creating the context we use for segments and utterances.

### 3.6.1 Loss Functions

When performing undersampling, we use the binary cross entropy function for some of the experiments. For all other experiments, we use the Focal loss function from Lin et al. [43] as shown in Equation 3.3a.

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \, log(p_t) \tag{3.3a}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{3.3b}$$

$$\alpha_t = \frac{w_t}{\|w\|_1} \tag{3.3c}$$

$$w_t = \frac{N}{T * n_t} \tag{3.3d}$$

Where $\alpha_t$ is a weighting factor obtained by normalizing the class weight $w_t$ for class $t$. $T$ is the total number of classes, $N$ is the total number of data points and $n_t$ is the number of data points belonging to class $t$.

The Focal loss function adds a modulating factor $(1-p_t)^\gamma$ to the weighted binary cross entropy loss function, where the parameter $\gamma$ can be tuned to control the intensity of the effect. In Figure 3.3, the authors show the resulting behavior for $\gamma \in [0, 5]$. When $p_t$ goes to 1, meaning the error gets smaller, the modulating factor goes to 0. This results in the loss being down-weighted for well-classified examples, thereby focusing on hard-to-classify samples.

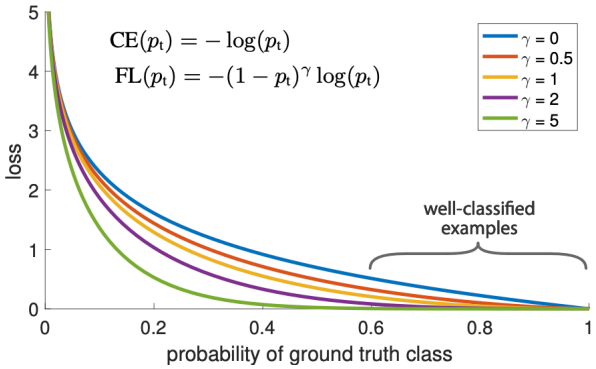


FIGURE 3.3: Effect of the $\gamma$ parameter on the loss function, from Lin et al. [43]

### 3.6.2 Context

We computed the context to evaluate to what extent the integration of past and future segments or speaker turns affects the classifier's performance. We based our approach on Sachdeva et al. Figure 3.4 shows the split between segments or speaker turns (here referred to as content) and the context for past and future segments or speaker turns. We use the sentence types employed by BERT for question-answering to distinguish between content and context, i.e., content is of sentence type A, and context is of sentence type B. For the classifier to distinguish between past and future context, we use the special token SEP employed by BERT. The only parameter used in our implementation is the context length. Here we used either 128 or 256. These numbers refer to the total length, i.e., the sum of the tokens of context-past and context-future. Sequence A from Figure 3.4 shows the configuration at the beginning of a transcript. Since we cannot use past segments in this stage, we insert the string "EMPTY" as a placeholder. Sachdeva et al. do not mention how they treat the beginning and end of a transcript nor if they add a surrogate token for the case of the beginning/ending segment or speaker turn. In sequences A and C, the lengths for context past and context future are calculated dynamically in each case. For example, if we assume a context length of 128, the context length of context-past = 1, and context-future = 126 = 128 - "EMPTY" - SEP, analogous to context-past = 126 in sequence C. Sequence D shows what happens for content longer than 512 - context-length, then the context is truncated from the right.

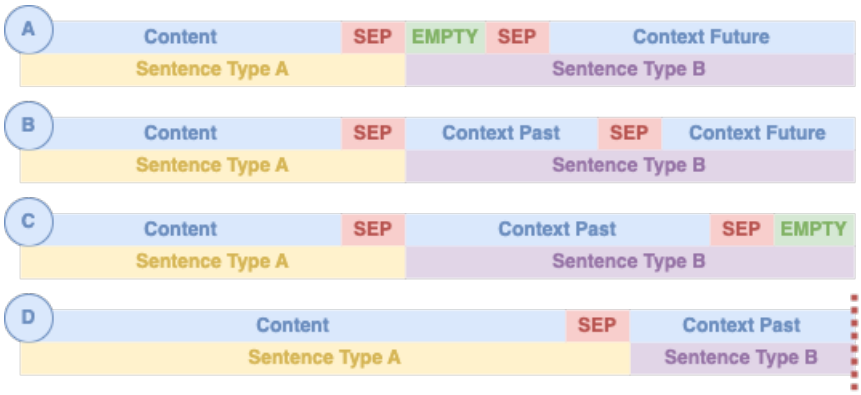


FIGURE 3.4: Computation of the context

# Chapter 4

# Data Analysis

In the data analysis, we aim to learn more about the structure and characteristics of action items using existing annotations. In the first step, we formed the union of the Gruenstein, AIDA, and AIMU annotations to determine features that frequently occur in action items based on the ICSI and ISL corpus, meaning the transcripts used can be of either corpus. Forming the union resulted in 2'234 annotated segments, for a total of 122'603 segments. The following analysis is collected based on this data.

## 4.1 Statistics

The corpus has 93 transcripts (union of ICSI and ISL), with a maximum of 10 speakers per transcript, a minimum of 3, and a mean of 6.32. Table 4.2 shows that the mean number of speakers uttering an action item is between 2.75 and 4.5, compared to the total mean of 6.32. The shortest transcript in the corpus has a duration of 7.77 minutes, and the longest is 101.96 minutes, with a mean of 51.42 minutes.

### 4.1.1 Splits

Table 4.1 shows the number of speakers per transcript that are uttering statements labeled as belonging to an action item for each set of annotations. Additionally, the Table shows the number of transcripts assigned to each split for the different annotations. The AIDA annotations were created for 18 transcripts. Our indicative annotations were created for 85 transcripts. We assign the transcripts to train, dev, and test set using the splits suggested by Lee et al. [42]. For Gruenstein and our indicative annotations, the split between dev and test sets is balanced; for the AIDA annotations, there are 3 transcripts in the dev set and 2 in the test set. Only for the AIMU annotations, there are no transcripts in the Train set. However, this has no impact on the training of our models since we use AIMU only in the context of the data analysis performed here.

| | **n. Speakers** | | | **n. Transcripts** | | | |
|---|---|---|---|---|---|---|---|
| | Min. | Max. | Mean | All | Train | Dev | Test |
| AIDA | 2 | 7 | 4.50 | 18 | 13 | 3 | 2 |
| Gruenstein | 1 | 8 | 3.25 | 56 | 42 | 7 | 7 |
| AIMU | 1 | 5 | 2.75 | 20 | 0 | 9 | 11 |
| Indicative | 1 | 8 | 4.19 | 85 | 63 | 11 | 11 |

TABLE 4.1: Number of speakers and number of transcripts for the different annotations

### 4.1.2 Segments

Table 4.2 shows the number of segments per split for each annotation set and the summary statistics for the duration of the segments across all annotated transcripts. The *all* column refers to the total number of segments contained in the annotated transcripts per annotation set (e.g., for AIDA, the total number of segments contained in the 18 annotated transcripts), the *pos* columns refers to the total number of annotated segments contained in each respective annotation set. As can be seen in the table section labeled *n. Segments*, the AIDA and the AIMU annotations cover the least annotations, and, as shown in Table 4.1, both cover a similar number of transcripts (18 for AIDA and 20 for AIMU), yet the AIMU annotations only contain 126 annotated segments, whereas the AIDA annotations contain 792 annotated segments. Our indicative annotations contain the most labeled segments (1'908), followed by Grunestein with 1'777. However, they are also spread across most transcripts. Namely, 85 compared to 56 for Gruenstein, as shown in Table 4.1.

| | **n. Segments** | | | | | **Duration** | | |
|---|---|---|---|---|---|---|---|---|
| | All | Pos | Train | Dev | Test | Min. | Max | Mean |
| AIDA | 25'862 | 792 | 541 | 133 | 118 | 0.22 | 17.90 | 2.47 |
| Gruenstein | 78'984 | 1'777 | 1'335 | 223 | 219 | 0.22 | 20.21 | 3.29 |
| AIMU | 25'393 | 126 | 0 | 58 | 68 | 0.51 | 16.99 | 4.22 |
| Indicative | 111'743 | 1'908 | 1'410 | 268 | 230 | 0.59 | 26.86 | 4.51 |

TABLE 4.2: Number of segments and durations of segments for the different annotations

The shortest segment has a duration of 40 milliseconds (an interrupted word), and the longest is 28.86 seconds, with a mean of 2.97 seconds. The minimum number of segments per transcript is 198; the maximum is 2'674, with a mean of 1'318.34. As can be seen in Table 4.2, the mean duration of the annotated segments is slightly higher than the mean duration for all segments, except for AIDA, which we assume is because AIDA's agreement class contains very short segments, i.e., utterances. Our indicative annotations have the longest mean duration, with 4.51 seconds.

### 4.1.3 Speaker Turns

We only analyzed speaker turns in connection to our indicative annotations. As shown in table 4.3, the average number of speaker turns per transcript is 527.08. The minimum number of speaker turns in a transcript is 67, and the maximum number of speaker turns per transcript is 1209. In contrast, the average number of indicative speaker turns per transcript is 18.38., the minimum number is 1, and the maximum number of indicative speaker turns per transcript is 61. Assuming the average values, we have 3.49% indicative speaker turns per transcript.

| | **Min.** | **Max.** | **Mean** |
|---|---|---|---|
| **All** | 67 | 1209 | 527.08 |
| **Indicative** | 1 | 61 | 18.38 |

TABLE 4.3: Number of speaker turns per transcript

In table 4.4 and chart 4.1, we can see the duration of the general speaker turns, and the indicative speaker turns in seconds. We note that the indicative speaker turns generally have a longer duration than the regular speaker turns, i.e., the average duration of speaker turns is almost 4 times longer than that of the other speaker turns. In Figure 4.1, we notice that the interquartile range of the indicative speaker turns is broader than that of the other speaker turns. For clarity, we have not shown the outliers in the box plot.

| | Min. | Max. | Mean | Median |
|---|---|---|---|---|
| **All** | 0.12 | 452.79 | 5.92 | 2.47 |
| **Indicative** | 0.68 | 452.79 | 19.93 | 9.09 |

TABLE 4.4: Duration in seconds of all speaker turns compared to indicative speaker turns

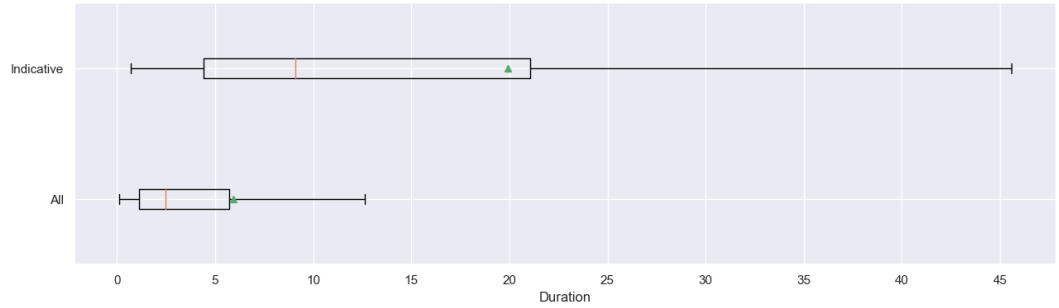


FIGURE 4.1: Duration in seconds of all speaker turns compared to indicative speaker turns

In table 4.5, we see the number of speaker turns in the different splits. Here we see that the number of annotations has been reduced by merging the segments into speaker turns in the dev set from 268 to 216 and in the test set from 230 to 199. As a result, the total number of positive annotations was reduced from 1'908 to 1'147.

| | Train | Dev | Test |
|---|---|---|---|
| **All** | 33'395 | 5'788 | 5'619 |
| **Indicative** | 1'147 | 216 | 199 |

TABLE 4.5: Number of speaker turns in the different splits

## 4.2 Overlap

Figure 4.2 shows the overlap between the different annotations. One can see that the Gruenstein annotations contain the highest number of segments labeled as belonging to an action item. Next come the AIDA annotations, with an Overlap of 52.8% to the Gruenstein annotations. The overlap between AIMU and AIDA annotations (with 17.5%), i.e., Gruenstein annotations (with 31%), is significantly smaller.

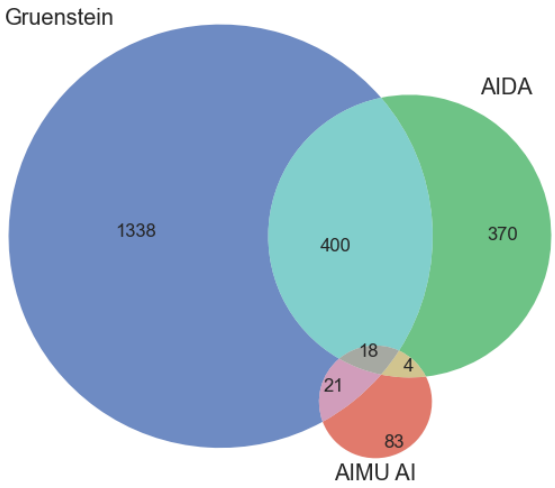


FIGURE 4.2: Overlap between the different annotations

### 4.2.1 Starting Point

First, we determined what categories of features we wanted to look at to analyze the data. Our goal was to choose those features that provide good explainability, i.e., features that describe sentence or word function or directly indicate characteristic content of action items. We decided to use the following features:

1. **NER**: Named entities most commonly used in segments containing action items
2. **Dialogue Acts**: Relating the MRDA ICSI annotations from utterances back to segments
3. **Tenses**: Tenses most commonly used in segments containing action items With a focus on future tenses
4. **Characteristic Words**: Identifying words most characteristic for segments containing action items

### 4.2.2 Characteristic Words

We employed the vector space model to understand which words are characteristic of an action item. Peters et al. [44] explain that each dimension in the m-dimensional vector space represents a unique feature, i.e.word. We represented each transcript as an m-dimensional document vector as shown in Equation 4.1a from [44] with each entry containing the respective feature weight $w$ that we obtain by computing Equation 4.1c from [44], where $ff$ denotes the feature frequency, $\varphi$ a single feature and $d$ a single document.

$$\vec{d_j} = \big(w(\varphi_0, d_j), ..., w(\varphi_k, d_j), w(\varphi_k, d_j)\big)^T \tag{4.1a}$$

$$idf(\varphi_k) = \log\left(\frac{1+N}{1+df(\varphi_k)}\right) \tag{4.1b}$$

$$w(\varphi_k, d_j) = ff(\varphi_k, d_j) * idf(\varphi_k) \tag{4.1c}$$

We then concatenated all utterances marked as action item, thereby obtaining a synthetic document (synthetic in the sense that we only created it for this specific setting) containing the sum of all features, i.e., words comprising action items in our dataset. As a last step, we could compute the features weights of this synthetic document, and by using the $idf$, where $N$ represents the total number of documents, as shown in Equation 4.1b from [44], we could rank the words according to their importance, where importance in this setting means the words that most often occur in utterances marked as action items, but whose relative frequency is low in the other documents, i.e., transcripts comprising the corpus. The plot in Figure 4.3 shows the 20 words from the synthetic document with the highest Tf-Idf weighting. From this, we can see that words related to time, such as weekdays, or words like "tomorrow" get a high weighting, as well as verbs and nouns related to typical tasks of knowledge workers, such as "send", "email", "meet" and "list". What is also consistent with the intuition regarding action items is the fact that several names of people appear in this list, such as "Fey", "Nancy", or "Brian".

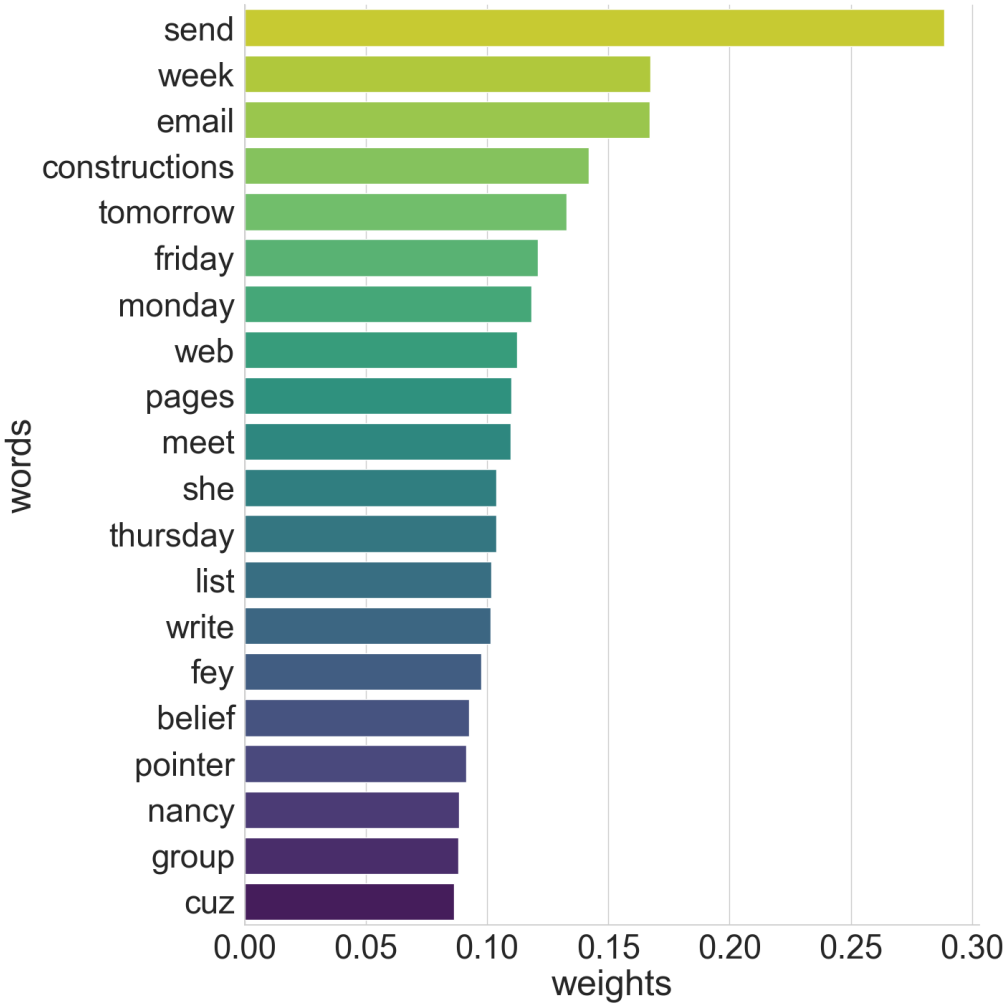


FIGURE 4.3: The 20 words with the highest Tf-Idf weights from segments marked as action items

## 4.3 Significance Test

To test which features have the strongest relation to the label, i.e., whether a segment belongs to an action item, we used the $\chi^2$-test, since both predictor and outcome variables are categorical. In Figure 4.4, the 20 features with the highest value for the test statistic are presented, where the p-value for the bottom feature (the word list) is $6.299e-11$. Note that the x-axis (displaying the value for the test statistic) is scaled logarithmically.

1. **Null hypothesis ($H_0$):** The occurrence of the computed feature in an utterance and whether the utterance is an action item are not related in the population; The proportions of times when the feature occurs is the same regardless of the utterance is an action item or not.

2. **Alternative hypothesis ($H_a$):** The occurrence of the computed feature in an utterance and whether the utterance is an action item are related in the population; The proportions of times when the feature occurs are different when the utterance is an action.

For the significance test, we removed the segments that had an AIDA annotation of type agreement, had a dialogue act of type backchannel, and were at most 4 tokens long since they typically would look like the following examples:

- *Sure.*
- *Hmm.*
- *Right. Yeah.*
- *O_K.*

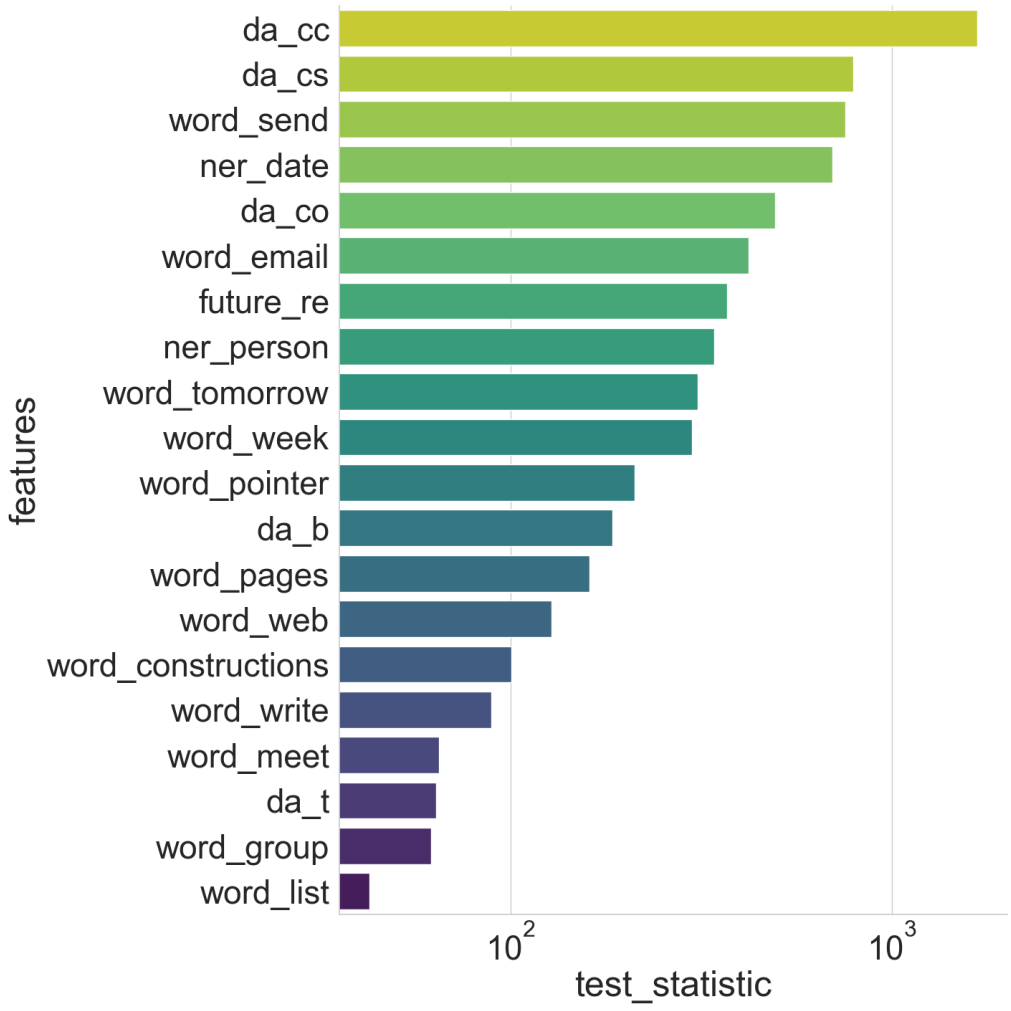


FIGURE 4.4: Most significant features w.r.t. to being an action item

Figure 4.4 shows a strong connection between segments marked as action items and dialogue acts of type action motivators. The dialogue acts of type action motivators include da_cc (commitment), da_cs (suggestion), and da_co (command). The connection between action motivators and action items is in line with Frampton et al., who point out that: "we observed that using the MRDA dialogue act tags commitment and suggestion improved precision significantly" [16]. Furthermore, Yang et al. mention in [15] that "because action motivators lead to future actions, they are probably also action item descriptions.". Among the characteristic words, it is noticeable that both the word 'send' and the word 'email' rank highest. This ranking can be explained by the fact that many action items are related to emails, be it the request to send an email or the commitment to follow up with an email. It is noticeable that features with a temporal reference are strongly represented. For example, the named entity Date is in fourth place, the regex expression for future tense is in seventh place, and the word Tomorrow is in ninth place. In eighth place is the Named Entity Person. We conclude that segments that refer directly to a person appear more frequently in segments labeled as action items.

## 4.4 Action Item Clustering

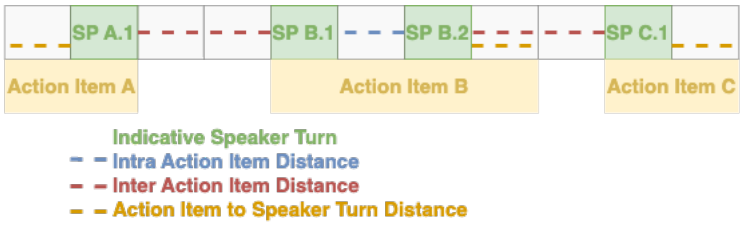


FIGURE 4.5: relation and measures of distance for indicative speaker turns and topics

Since action items can comprise more than one speaker turn, we need a way to assign multiple speaker turns to an action item. Gruenstein et al. use topics, while Purver et al. uses an action item type. In Figure 4.5, we visualize this concept as a yellow block labeled action item. The green blocks represent speaker turns, which have been annotated as indicative of the presence of an action item. If a green block is located within a yellow block, it belongs to the yellow action item. The gray blocks represent speaker turns without annotations. The blue dashed lines indicate the distance between two annotated blocks within an action item (intra-action item distance). The red dashed lines represent the distance of the first indicative speaker turn to the last indicative speaker turn of the preceding action item or the last speaker turn to the first speaker turn of the following action item (inter action item distance). As part of the analysis, we would like to evaluate the following:

1. Determine if the intra action item distance is significantly smaller than the inter action item distance.

2. Get a sense for the distance between the first indicative speaker turn and the first speaker turn that is thematically related to the action item under consideration.

To thematically assign the speaker turns to an action item, we use the topics in Gruenstein's annotations and the action item types in Purver's annotations. We also used topics and action item types to identify the first and last speaker turn, which is thematically related to the action item under consideration.

### 4.4.1 Analysis of Annotations

We separately considered the topics of authors Cgilbert and Michaeld (contained in Gruensteins annotations). In the same way, we considered the annotations of Purver. In the first step, we identified action items that contained at least one speaker turn annotated as indicative, which yielded the following numbers:

- Cgilbert: 193
- Michaeld: 207
- Purver: 150

### Speaker Turns and Action Items

Figure 4.6 shows how many indicative speaker turns are contained in an action item. Here we can see that more than half of all action items contain exactly 1 indicative speaker turn, for Cgilbert 52%, for Michaeld 72%, and for Purver 75%. Figure 4.7 shows how many speaker turns are contained in an action item. The diagrams show that about half of all action items contain only one indicative speaker turn, which is evident when considering that also about half of the action items contain exactly one speaker turn in total. However, this characteristic is surprising since, as stated by Purver in [11], an action item usually spans several utterances, i.e., speaker turns from different speakers.

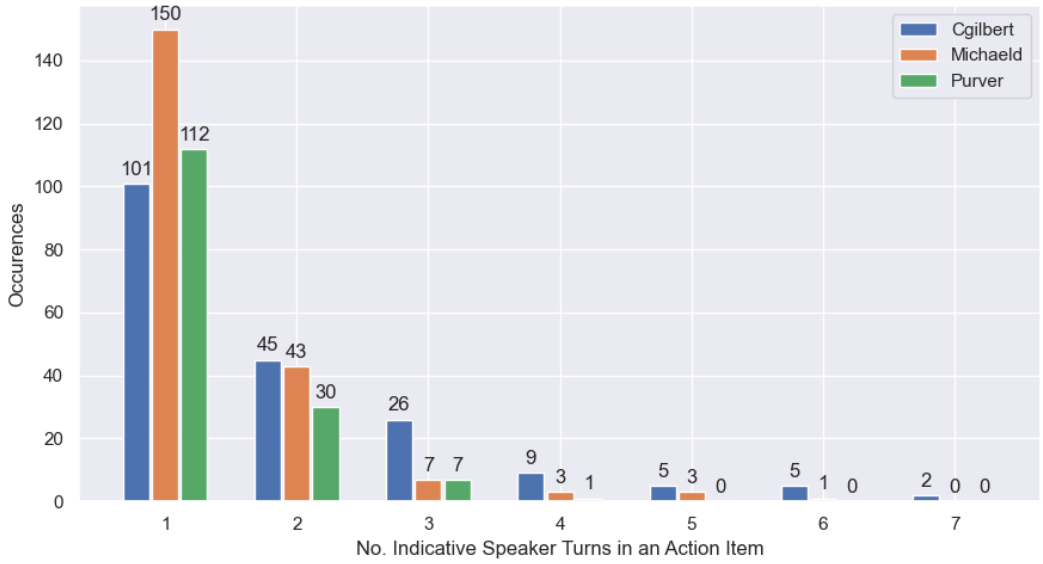


FIGURE 4.6: Number of indicative speaker turns per action item

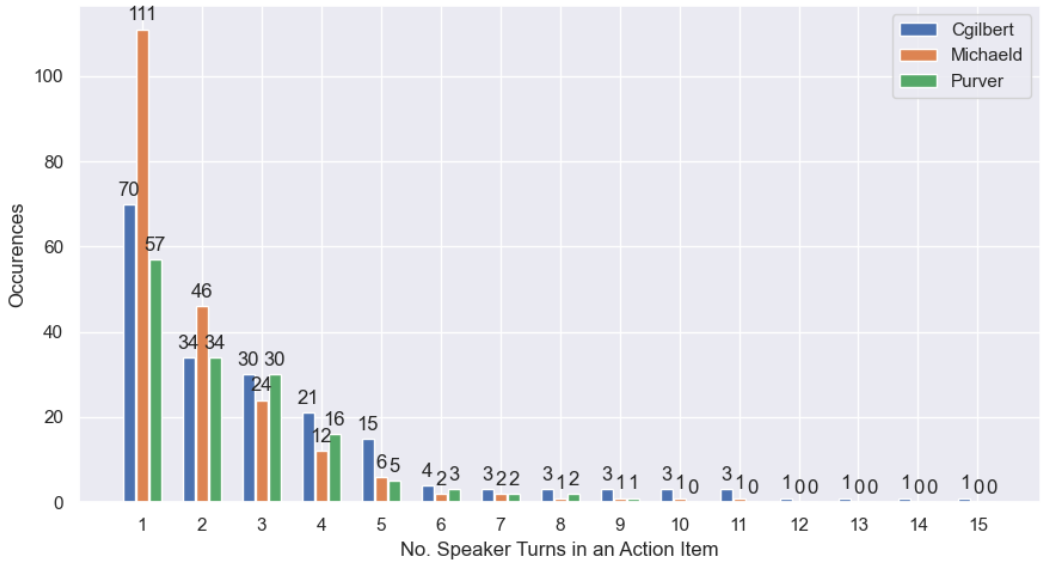


FIGURE 4.7: Number of speaker turns per action item

To determine the length of the context before and after an indicative speaker turn, within which there are other speaker turns that belong to the same action item, we determined the distance between the first speaker turn of an action item and the first indicative speaker turn of the same action item. Analogously we proceeded to the last speaker turns of an action item. As shown in Figure 4.8, in more than 80% of the cases, the distance is 0, meaning that the first speaker turn of an action item is also the indicative speaker turn, for Cgilbert namely 83% for Michaeld 90% and for Purver 85%. In a similar vein, the last indicative speaker turn and the last speaker turn of an action item are shown in Figure 4.9, where the number of indicative speaker turns that are also the last speaker turns of the respective action item is 74% for Cgilbert, 79% for Michaeld, and 58% for Purver. These numbers indicate that action items more often start with an indicative speaker turn than end with one.

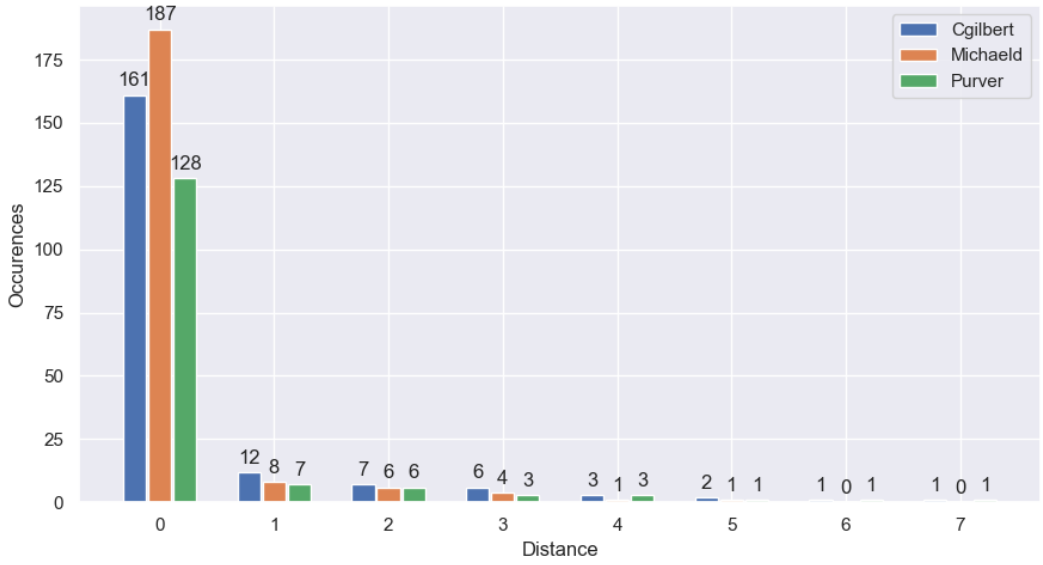


FIGURE 4.8: Distance between the first speaker turn of an action item and the last occurring indicative action item

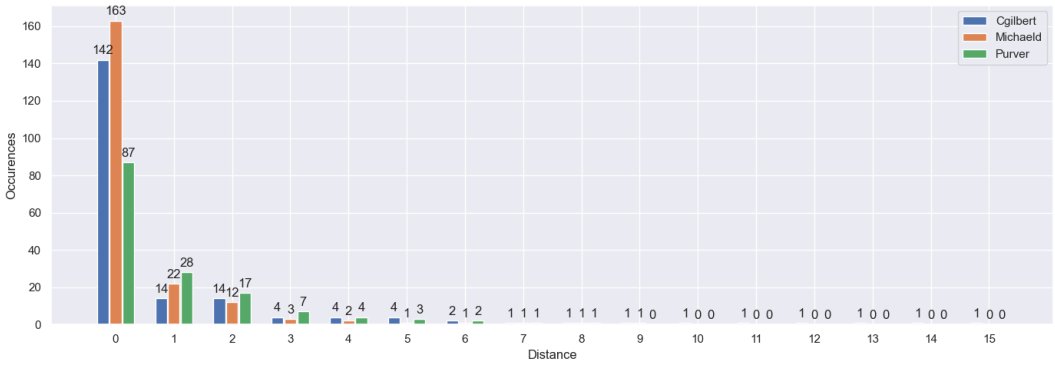


FIGURE 4.9: Distance between the last speaker turn of an action item and the last occurring indicative action item

### Distances

We studied the distance between indicative speaker turns. We divide the analysis into 2 categories, namely, as visualized in Figure 4.5, intra action item distance, where we calculate the distance between speaker turns of the same action item, and inter action item distance, where we calculate the distance between indicative speaker turns of adjacent action items. In Figure 4.10, we see that the median distance for all authors is 2. The third quartile is 6 for Cgilbert, 5 for Michaeld, and 4 for Purver. Comparing these numbers with those of the inter action item distance is shown in Figure 4.11, we can see that the inter action item distance is significantly higher. Here the median distance is 19 for Cgilbert, 24 for Michaeld, and 16.5 for Purver. Here the third quartile is 98 for Cgilbert, 65.75 for Michaeld, and 45.25 for Purver.

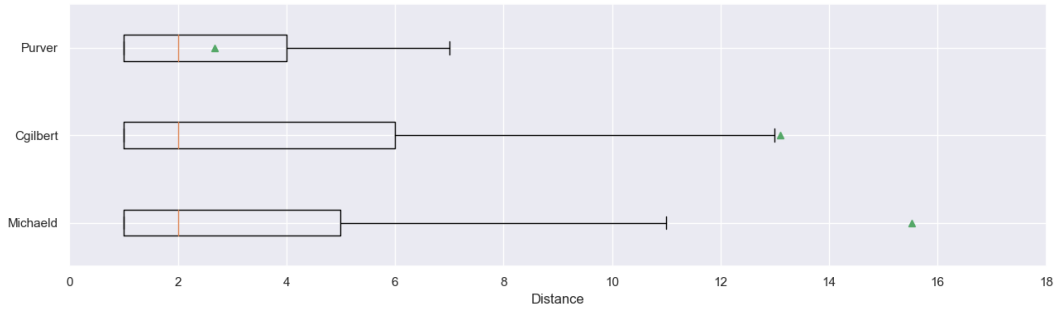


FIGURE 4.10: Distance between indicative speaker turns belonging to the same action item

In our analysis, we did not consider overlapping action items, which is the case when a speaker turn is assigned to action item A, followed by a speaker turn that is assigned to action item B, and then another speaker turn follows, which is assigned to action item A. After that, further speaker turns assigned to action item B can follow. Here, the discussion about action item B is already started, while action item A is still being discussed. In such a case, action items cannot be distinguished unambiguously only using distance-based methods. In our corpus, however, only 12 indicative speaker turns are affected by this circumstance, and only for Cgilbert's action items.

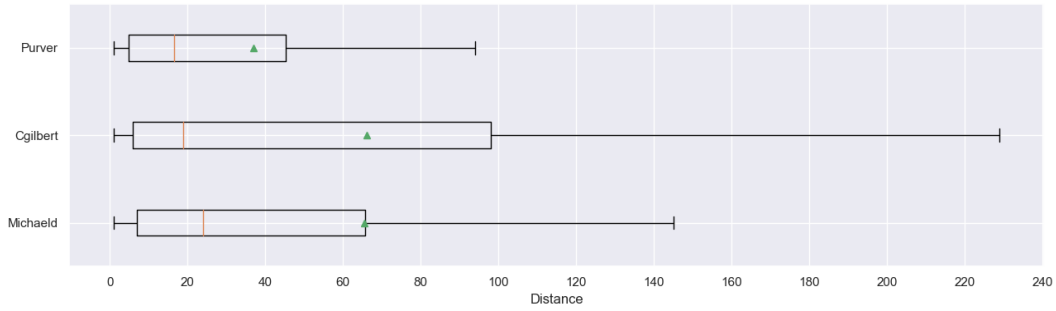


FIGURE 4.11: Distance between 2 action items, measured from the indicative speaker turns located at the respective edges.

# Chapter 5

# Experiments

This section describes the configuration of the experiments carried out in this thesis and the results obtained. We group the experiments into the following categories:

- Segment-based experiments
  - AIDA annotations
  - Indicative annotations
- Speaker turn based experiments
  - Indicative annotations

In the segment-based experiments, we classify segments as found in the original transcripts of the ICSI corpus. In the speaker turn based experiments, we classify speaker turns obtained by merging segments by speakers as described in section 3.5.4.

## 5.1 Common Settings

The common settings were used throughout all experiments, regardless of annotations and segmentation unit, i.e., whether we used segments or speaker turns.

### 5.1.1 Model and Hyper Parameters

For all the experiments, we used the bert-base-cased model from Devlin et al. [45], and we employed the implementation from Huggingface [46]. We also tried bert-large and xlm-roberta-base, but that did not improve the results. We used a learning rate of $2e-5$ for finetuning and a batch size of 32. We used early stopping for all the experiments, with a patience of 10.

### 5.1.2 Experiment Configuration

We tried to train the models used on the data as-is and with a class-weighted binary cross-entropy loss function, but in both cases, that led to an F1 score of 0. Only when we started to use undersampling or Focal-loss did we start to obtain results with an F1-score $> 0$. For this reason, we will only report the experiments using undersampling or Focal-loss. For Focal-loss, we always used a $\gamma$-value of 2, and $\alpha$-value dynamically computed as explained in Section 3.5.5.

For the AIDA annotations on segments and the indicative annotations on segments, as well as for the indicative annotations on speaker turns, we always followed the same sequence of experiments:

1. Focal-loss
2. Focal-loss with a context length of 128
3. Focal-loss with a context length of 256
4. Undersampling
5. Undersampling with a context length of 128
6. Undersampling with a context length of 256
7. Focal-loss and $\frac{1}{10}$-Undersampling
8. Focal-loss and $\frac{1}{10}$-Undersampling with a context-length of 128
9. Focal-loss and $\frac{1}{10}$-Undersampling with a context-length of 256

$\frac{1}{10}$-Undersampling means that we calculated the number of positive samples and picked 10 times as many negative samples. The $\alpha$-value was computed on this $\frac{1}{10}$-ratio rather than the original class distribution. We found this showed better results, even when evaluating on the test set.

## 5.2 Segment Based Experiments

In the segment-based experiments, the classifier had to predict whether or not a single segment (as defined in Section 3.2) was annotated as an action item. We performed the segment classification based on Purver's AIDA annotations to obtain a baseline that would allow us to put our results in perspective with previous results. We then performed the same experiments instead of using the AIDA annotations with our indicative annotations.

## 5.3 Speaker Turn Based Experiments

In the speaker turn based experiments, the classifier had to predict whether a speaker turn was annotated as an action item. Since we are unaware of other work that uses the available action item annotations in conjunction with speaker turns as a segmentation unit, we performed the speaker turn based experiments only with our indicative annotations. We computed the context the same way as we did for the segment-based experiments. While a significant part of the speaker turns have less than 512 tokens, some speaker turns are longer than 512 tokens. The mean token length of speaker turns is 26, the speaker turn with the most tokens is 1'953 tokens long, and there is a total of 44 speaker turns with more than 512 tokens, of which 19 are labeled as being indicative. For those 19 speaker turns, there will be no context due to the truncation-from-right we apply, meaning additional tokens will be truncated from the end for speaker turns longer than 512 tokens. Since we append the context after the speaker turns, it will be removed.

# Chapter 6

# Results

In this chapter, we present the results of our experiments. First, we will discuss the qualitative analysis. Here we mainly use SHAP values and look at text examples. For practical reasons, we limit ourselves to examples of segments since speaker turns would often extend over half a page. We will then discuss the quantitative analysis, which we divide into two parts, segment-based experiments, and speaker turn-based experiments.

## 6.1 Qualitative Analysis

In this section, we first use SHAP bar charts, as they allow us to evaluate the influence of words across multiple segments. Then, we will look at examples of individual segments for *true positives* and *true negatives* to show concrete examples of what criteria the classifier uses to make its decision.

The SHAP values bar chart shown in Figure 6.1 shows the words for the indicative annotations that have the most substantial impact when classifying a segment as an action item. We generated them using the 10 speaker turns classified as true positives with the highest confidence. The red bars mean that a word contributes positively to the classifier's decision. When we compare the words shown here with those from Figure 4.3 in Section 4.2.2 concerned with identifying the most characteristic words, we can see that the following words overlap: send, week, email, tomorrow, meet and write. Interestingly the bar chart lists day as a word, whereas Friday, Monday, and Thursday are listed separately in Figure 4.3. This overlap shows that the features we identified while examining existing annotations are, to a large extent, the same features the classifier uses to discriminate between positive and negative examples.

In the previously described case, one could argue that this overlap was caused by the fact that the characteristic words themselves were part of the features we used to generate the training data (when we applied weak supervision). Interestingly, we can also see this overlap in the model trained exclusively on the AIDA annotations, which we can see in diagram 6.2 depicting the True Positives. Here the following words overlap: *send*, *email*, *tomorrow* and *meet*.

The SHAP values bar chart shown in Figure 6.3 shows the words that have the most significant impact when classifying a segment as not being indicative of an action item. Again it was generated using 10 speaker turns, but this time those classified as true negatives with the highest confidence. The red bars mean that a word contributes positively to the classifier's decision. However, here we see that the words

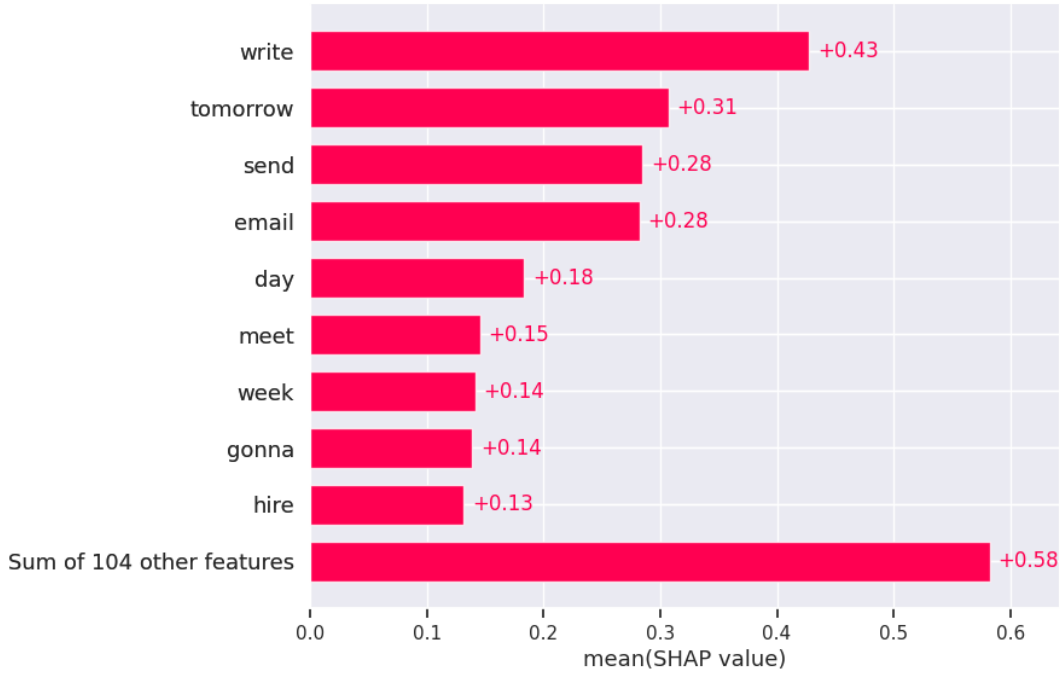


FIGURE 6.1: Words with highest Shapley Value on True Positives, evaluated on the indicative annotations

with the largest SHAP values are blue, which means they contribute negatively to the classifier's decision. We hypothesize that the classifier decides that a segment does not belong to the class of speaker turns indicative of the presence of an action item rather by the absence of indicative features than by the presence of features typical of speaker turns that are not related to action items, as the SHAP values that contribute negatively to the classifier's decision are more prominently represented in the respective bar chart then the features that contribute positively.

In the SHAP bar plot shown in Figure 6.4 for the evaluation of the True Negatives of the model trained and evaluated on the AIDA annotations, we can see that the features that have the most substantial influence on whether the model classified an utterance as not belonging to the class of speaker turns indicative of the presence of an action item are all verbs and that 3 of the 4 verbs are in the past tense. Similar to the SHAP Bar plot of True Negatives for our indicative annotations in Figure 6.3, the negative features dominate the model's decision not to classify an utterance as belonging to an action item. We also want to highlight the verbs here, namely: *visiting*, *doing*, and *plan*.

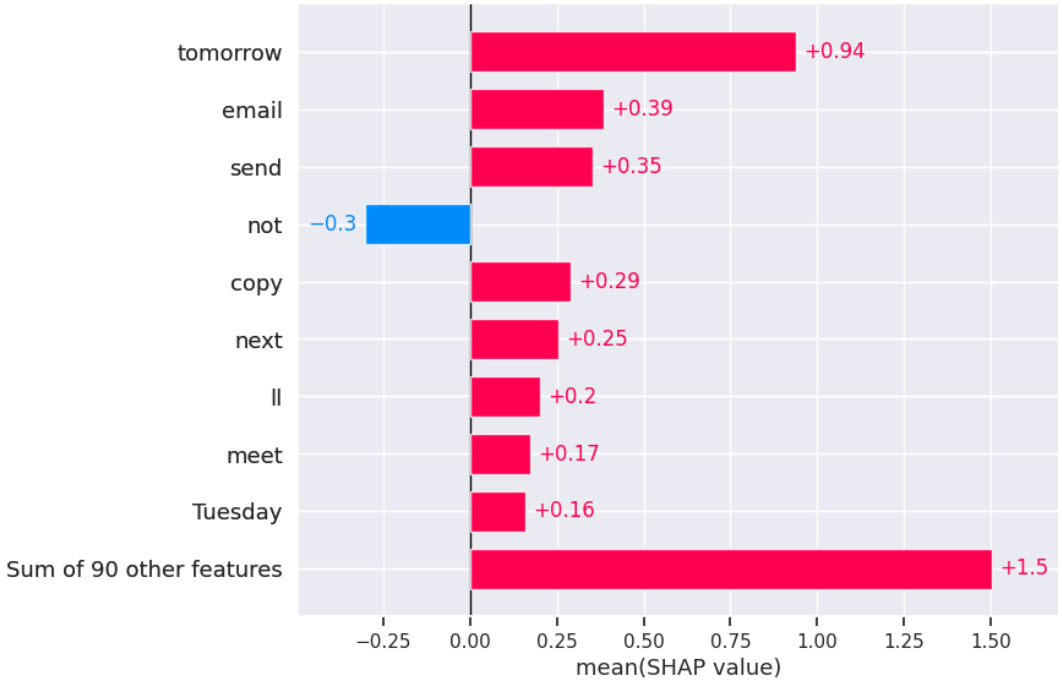


FIGURE 6.2: Words with highest Shapley Value on True Positives, evaluated on the AIDA annotations

### 6.1.1 True Positives

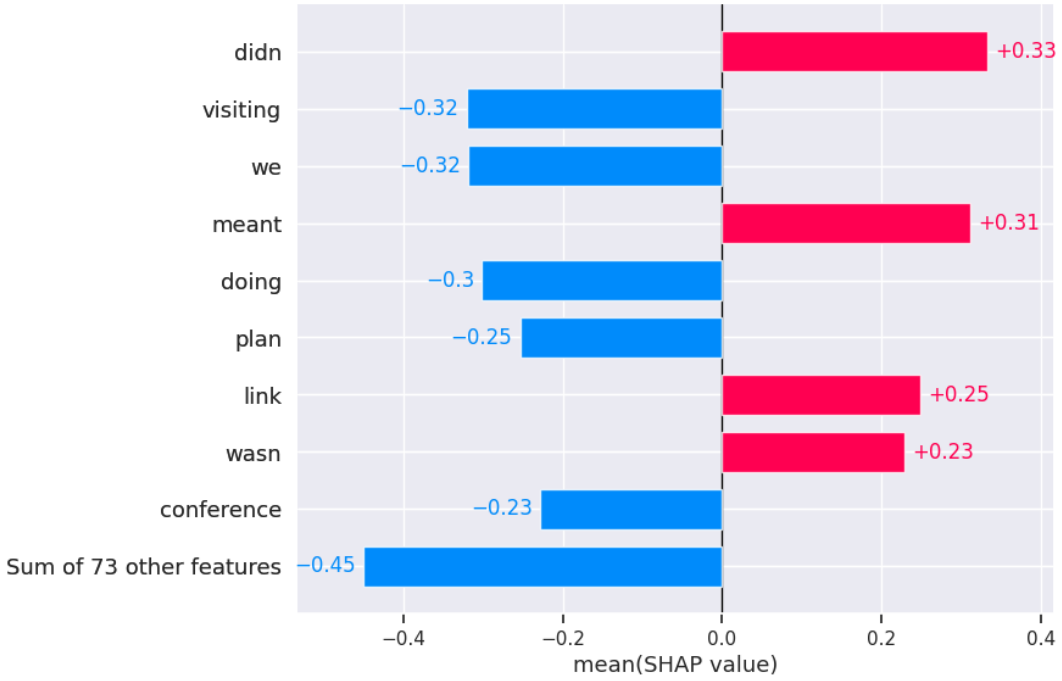


FIGURE 6.4: Words with highest Shapley Value on True Negatives, evaluated on the AIDA annotations

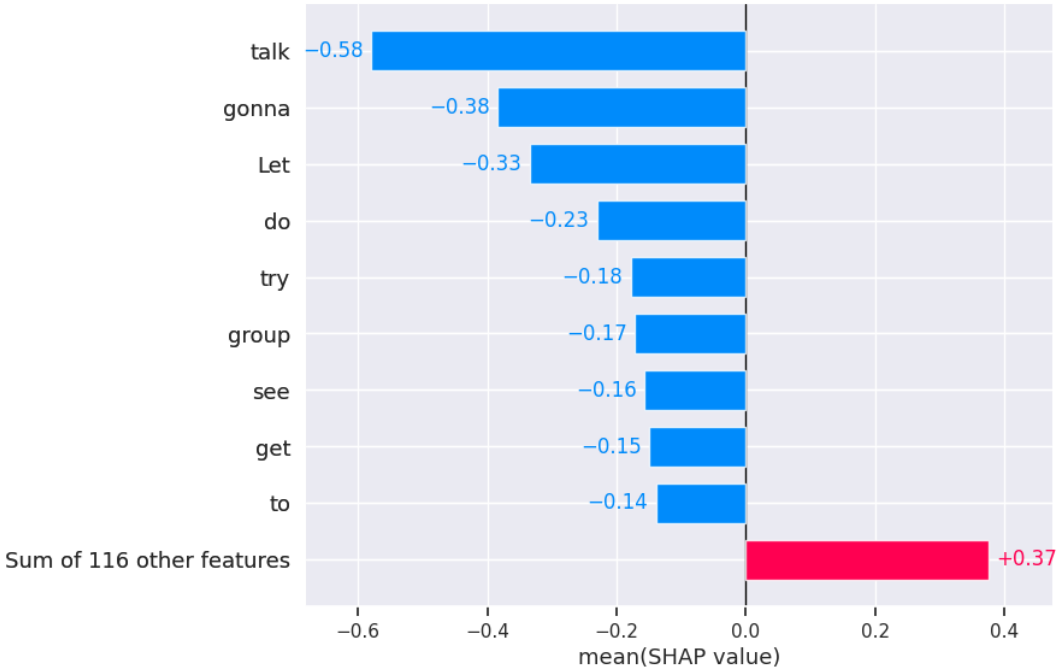


FIGURE 6.3: Words with highest Shapley Value on True Negatives, evaluated on the indicative annotations

Figure 6.5, 6.6 and 6.7 show the SHAP values text plot for sentences where the classifier predicted that a segment is indicative for an action item. For each of the following plots, the base value shows the model output when all words of the respective utterance are masked, and the value $f_{LABEL\_1}(inputs)$ shows the model's output given the full unmasked input text. Words highlighted in red contribute positively to the classifier's decision, whereas words highlighted in blue contribute negatively. The arrows above the text show the magnitude by which each highlighted word affects the classifier's decision.

What all segments displayed in Figures 6.5, 6.6, and 6.7 have in common is that they make references to the future. We can see that words like *gonna* or *will*, in the form of *'ll* have the most substantial impact (meaning they are highlighted in red), but also words such as *check*, *ask*, *week* and *tomorrow* contribute positively. As with the SHAP values bar chart, we see that most of the words that have the most decisive influence on the classifier also overlap with the most influential words we identified during the data analysis.

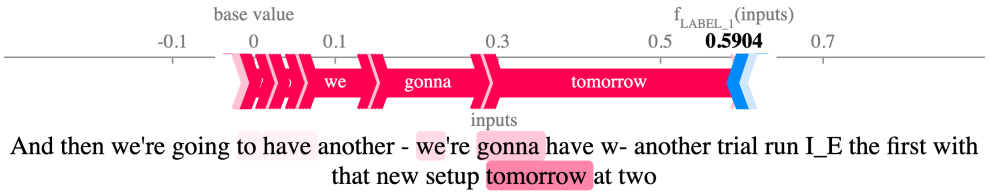


FIGURE 6.5: Shapley Value on True Positives ex. 1

The utterance in Figure 6.5 is of dialogue act type *command*. It refers to an exact time ("tomorrow at two") and a concrete task ("another trial run"). We can see that

the reference to a future date (*tomorrow*) impacts the classifier's decision the most, followed by the verb in the future tense ("gonna have"). Interestingly, the word *going* used in "going to have" does not strongly impact the classifier's decision, nor does the actual task description.

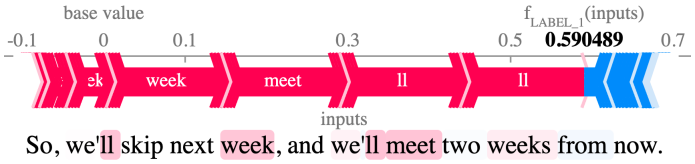


FIGURE 6.6: Shapley Value on True Positives ex. 2

The utterance in Figure 6.6 is also of dialogue act type *command*. Here we see that the verb indicating the future tense, in this case, *will*, impacts the classifier's decision the most. In contrast to the utterance in Figure 6.5, the verb implying the action itself (*meet*) has a strong influence on the classifier's decision, followed by the word *week*.

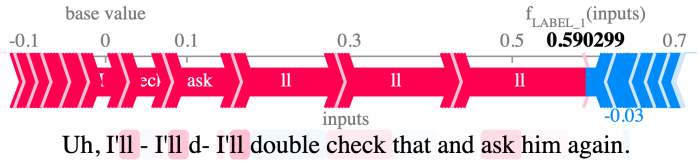


FIGURE 6.7: Shapley Value on True Positives ex. 3

The utterance in Figure 6.7 is of dialogue act type *commitment*, and, similar to the utterance in Figure 6.6, again the verb *will* impacts the classifier's decision the most, as well as the verbs *check* and *ask* which imply the activity. In summary, we can say that the dialogue act types of all three utterances belong to the category of action motivators, and all utterances are in the future tense.

### 6.1.2 True Negatives

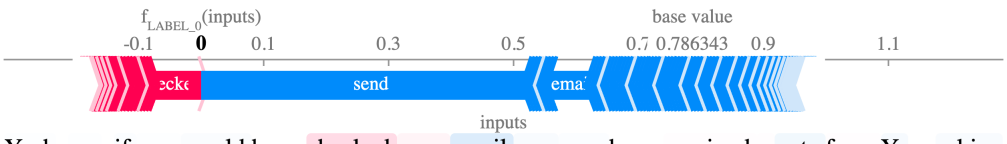


FIGURE 6.8: Shapley Value on True Negatives ex. 1

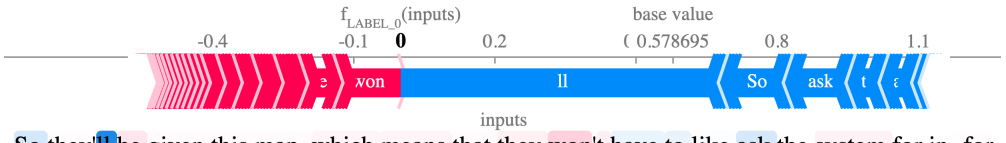


FIGURE 6.9: Shapley Value on True Negatives ex. 2

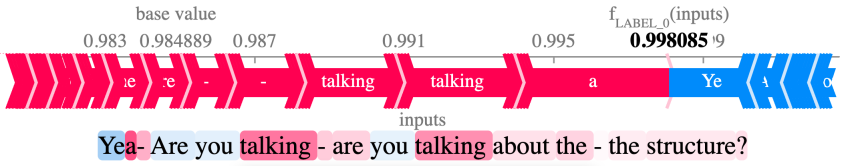


FIGURE 6.10: Shapley Value on True Negatives ex. 3

In the visualizations for the True Negatives, we see that the features with the most substantial influence are the negative ones in blue. In Figure 6.8, the words *email* and *send* are among them. The only word that contributes positively to the decision is the word *checked*. We see that the classifier judges a verb in past continuous as not indicative of the presence of an action item in the corresponding speaker turn. Similarly, the classifier evaluates a verb in present continuous as not indicative, as shown in the sentence in Figure 6.10 for the verb *talking*. In Figure 6.9, we can see that the only word contributing positively to the decision, colored in red, is *won't*. The strong impact of negation on the classifier's decision can be explained intuitively by the fact that an action item cannot be described by not doing something. In summary, we can say that none of the 3 speaker turns dialogue acts belonged to the category of action motivators. Furthermore, we can say that, in contrast to the true positives, the true negatives contained verbs in the past tense and the continuous Present.

## 6.2 Quantitative Analysis

In this section, we first show the results of the segment-based experiments. These comprise two experiment series:

1. In the first series, we used Purver's AIDA annotations. We did this primarily to compare the results obtained using our indicative annotations with those when using Purver's and to relate our results to those of previous work.

2. In the second series, we used our indicative annotations. Otherwise, we used precisely the same setup as in the first series.

We focus our analysis on *precision*, *recall*, and *F1-scores*. This approach allows us to go into more detail about the differences between the various approaches.

### 6.2.1 Segment Based Experiments

Table 6.1 summarizes the results of the experiments for the AIDA annotations. We have run these only for the segments, but not for the speaker turns. The upper 2 rows show the results that Sachdeva et al. obtained. The first line shows their result when using Bert Large as a base model. The sequence length in this model is 512 tokens, as in our experiments. Sachdeva told us that in training, "it was a combination of using class weights and oversampling" (Kishan Sachdeva, personal communication, January 5, 2023), yet we were not able to achieve the 0.39 F1 score. In line 2, we can see a 0.43 F1-score, which they achieved using the ETC transformer model of Ainslie et al. [47], which allows input sequence lengths of 4096 tokens. However, as also mentioned by Sachdeva et al. [19], using the ETC model is computationally very demanding and therefore was no option in this thesis.

| | Precision | Recall | F1 |
|---|---|---|---|
| **Sachdeva et al. Bert Large** | - | - | 0.39 |
| **Sachdeva et al. ETC (4096 Token Input)** | - | - | **0.43** |
| **Focal** | 0.07 | **0.76** | 0.13 |
| **Focal+Context 128** | **0.49** | 0.28 | **0.35** |
| **Focal+Context256** | 0 | 0 | 0 |
| **Undersampling** | 0.15 | 0.69 | 0.25 |
| **Undersampling+Context 128** | 0.17 | 0.75 | 0.28 |
| **Undersampling+Context 256** | 0.14 | 0.61 | 0.22 |
| **10th-Undersampling+Focal** | 0.20 | 0.40 | 0.27 |
| **10th-Undersampling+Focal+Context 128** | 0.24 | 0.22 | 0.23 |
| **10th-Undersampling+Focal+Context 256** | 0.26 | 0.23 | 0.24 |

TABLE 6.1: Experiments on segments using AIDA annotations

We can see that our best result achieves a *F1-score* of 0.35. However, we would like to highlight that we achieved this result using the Huggingface model "bert-base-cased", whereas Sachdeva et al. used "bert-large-uncased" in their experiments. With a value of 0.49, it is also the result with the highest *precision*. The setting used consists of Focal loss in conjunction with context 128. It is interesting to note that the smaller context yields better results than the larger one. Assuming an average segment length of 26 tokens, we get the best results with 154 (26 + 128) tokens. So in this series, more tokens do not automatically mean better results. On the contrary, in the experiment where we used Focal loss in conjunction with a context of 256, we got an *F1-score* of 0. The training showed that the classifier had reached an accuracy of over 96% within the first epoch and then got stuck in a local optimum. Due to the strong class imbalance, the classifier could reach this accuracy without having to detect a single positive sample.

| | Precision | Recall | F1 |
|---|---|---|---|
| **Focal** | 0.33 | 0.62 | **0.43** |
| **Focal+Context 128** | 0.28 | 0.60 | 0.38 |
| **Focal+Context256** | **0.34** | 0.57 | **0.43** |
| **Undersampling** | 0.19 | 0.80 | 0.31 |
| **Undersampling+Context 128** | 0.15 | **0.90** | 0.25 |
| **Undersampling+Context 256** | 0.16 | 0.84 | 0.27 |
| **10th-Undersampling+Focal** | 0.31 | 0.70 | **0.43** |
| **10th-Undersampling+Focal+Context 128** | 0.31 | 0.67 | **0.43** |
| **10th-Undersampling+Focal+Context 256** | 0.26 | 0.23 | 0.24 |

TABLE 6.2: Experiments on segments using indicative annotations

Table 6.3 shows the results for our indicative annotations. A clear difference to the

results on the AIDA annotations from 6.1 is that here the larger context with 256 tokens (except when using $\frac{1}{10}$-undersampling) yields the higher *F1-scores* than, the smaller context with 128 tokens. Another difference is that several configurations provided the same *F1-score*. However, we recognize that 0.43 is the upper limit for the *F1-score*. The model that gives the highest *recall* is the one that combines undersampling and context 128, the model that gives the highest recall among the models with the highest *F1-score* is the one that combines $\frac{1}{10}$-undersampling and context 128. In our experiments, we observed that undersampling often results in high *recall* but at the expense of *precision*. What we find particularly striking about the results is that the model that uses *Focal loss* exclusively, without any context, is also among the models with the best *F1-score*.

### 6.2.2 Speaker Turn Based Experiments

| | **Precision** | **Recall** | **F1** |
|---|---|---|---|
| **Focal** | 0.27 | 0.78 | 0.40 |
| **Focal+Context 128** | 0.47 | 0.53 | 0.50 |
| **Focal+Context256** | 0.39 | 0.61 | 0.47 |
| **Undersampling** | 0.23 | 0.87 | 0.36 |
| **Undersampling+Context 128** | 0.19 | **0.91** | 0.31 |
| **Undersampling+Context 256** | 0.20 | 0.89 | 0.32 |
| **10th-Undersampling+Focal** | 0.07 | **0.91** | 0.14 |
| **10th-Undersampling+Focal+Context 128** | 0.41 | 0.71 | 0.52 |
| **10th-Undersampling+Focal+Context 256** | **0.58** | 0.57 | **0.58** |

TABLE 6.3: Experiments on speaker turns using indicative annotations

In this section, we describe the experiments we performed using speaker turns. In these experiments, we used only our indicative annotations. Table 6.3 shows the experiments' results. As with the segment-based experiments, we can see that the configurations where we used undersampling have the highest *recall*. Specifically, it is undersampling in combination with context 128 and $\frac{1}{10}$-undersampling. We can also observe across all series that the experiment with the highest *F1-score* is always the one with the highest *precision*. In this case, it is the experiment with $\frac{1}{10}$-undersampling in combination with context 256, which has both a *F1-score* and a *precision* of 0.58. What is striking is that in the speaker turn-based experiments, we can see that the best results all use context and that here, in contrast to the segment-based experiments, those using $\frac{1}{10}$-undersampling perform best, followed by those combining context with *focal loss*.

# Chapter 7

# Conclusions

This work demonstrates that transformer-based models yield competitive results starting from Purver's AIDA annotations. Although the F1 scores of the results on the AIDA annotations were slightly lower than those of Sachdeva et al., they were achieved using the small "bert-base-cased" model. Additionally, we showed that Focal loss could effectively address the class imbalance in the dataset without resorting to re-sampling.

It is important to note that BERT fine-tuning can produce unstable results with small training sets. Zhang et al. in [48] highlighted that seemingly minor factors can fundamentally change the results when using small data sets, which are defined as having fewer than 10,000 samples. In the case of AIDA annotations with only 792 positive samples and 541 in the train set, the results must be interpreted with caution when training and evaluating transformer-based models. Furthermore, the small test set of 118 samples limits evaluation.

In this thesis, we identified the characteristic features of utterances that correspond to an action item. Supported by findings from this analysis, we developed the concept of indicativity. Then, using the concept of Indicativity in combination with weak supervision, we created a new corpus consisting of action item annotations for the transcripts of the ICSI and ISL corpus, which we make publicly available at `https://github.com/gishamer/indicat`. We were able to show, starting from the same setup, that we can achieve significantly better results with the indicative annotations than with the AIDA annotations.

The evaluation was performed using speaker turns to align with real-world systems. Our results showed that context, as suggested by Sachdeva, combined with resampling strategies, significantly improved performance, unlike in segment-based experiments with our indicative annotations. We find the fact that context improves performance in the case of speaker turn-based experiments particularly interesting since a speaker turn contains more tokens than a segment in most cases. However, using a larger context in the segment-based experiments did not improve the results. Our Qualitative analysis confirmed that the features we identified during data analysis were indeed the features with the most significant impact on the classification results.

The data analysis revealed that action items could be clustered based on temporal features. In concrete terms, in the meetings we studied, only one action item was discussed in most cases in a given period. Therefore, the distance between speaker

turns belonging to the same action item is smaller on average than for speaker turns belonging to different action items.

# Chapter 8

# Discussion

The scarcity of training data is a significant issue in the domain of action item detection. To mitigate this, we propose the utilization of synthetic data generated via advanced language models such as GPT-3 and Chat-GPT to generate meeting transcripts with action item annotations. Our evaluation of these generated transcripts showed promising results. Obtaining real meeting recordings remains a desirable goal, however, obtaining access to such data is challenging due to privacy concerns in both corporate and academic settings.

Clustering utterances or speaker turns into action items is a promising area for investigation. This task can be approached from various angles, such as co-reference resolution or topic discovery using techniques such as Bert-Topic. Although we focus on transcribed data in this thesis, incorporating prosodic features, as highlighted by prior studies such as [10, 20], holds great potential for improving action item detection performance. However, exploring the use of prosodic features with transformer-based models remains an uncharted area. An interesting direction for future work would be to adopt an approach based on automatically transcribed recordings, which would more closely simulate a real-world scenario. De-noising methods such as BART may also be evaluated to gauge their impact on the performance of such systems in the presence of dysfluencies.

In our analysis, we found that dialogue acts play a crucial role in assigning utterances to action items and detecting indicative utterances. Although high accuracy in dialogue act classification is attainable, the segmentation of speaker turns into utterances with clear dialogue act assignments remains challenging. A possible avenue for future work is to formulate the problem of dialogue act tagging as a sequence tagging task, similar to Named Entity Recognition.

# Bibliography


[1] Elise Keith. *How many meetings are there per day in 2022? (and should you care?)* Sept. 2022. URL: https://blog.lucidmeetings.com/blog/how-many-meetings-are-there-per-day-in-2022.

[2] Evan DeFilippis et al. "The impact of COVID-19 on digital communication patterns". In: *Humanities and Social Sciences Communications* 9.1 (2022), pp. 1–11.

[3] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[4] Sudipto Mukherjee et al. "Smart to-do: Automatic generation of to-do items from emails". In: *arXiv preprint arXiv:2005.06282* (2020).

[5] Simon Scerri et al. "Classifying action items for semantic email". In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. 2010.

[6] Paul N Bennett and Jaime Carbonell. "Feature representation for effective action-item detection". In: *ACM SIGIR Special Interest Group on Information Retrival* (2005).

[7] Simon H Corston-Oliver et al. "Integration of email and task lists". In: (2004).

[8] Alexander Gruenstein, John Niekrasz, and Matthew Purver. "Meeting structure annotation: Data and tools". In: *6th SIGdial Workshop on Discourse and Dialogue*. 2005.

[9] A. Janin et al. "The ICSI Meeting Corpus". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).* Vol. 1. 2003, pp. I–I. DOI: 10.1109/ICASSP.2003.1198793.

[10] William Morgan et al. "Automatically detecting action items in audio meeting recordings". In: *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. 2006, pp. 96–103.

[11] Matthew Purver, Patrick Ehlen, and John Niekrasz. "Detecting action items in multi-party meetings: Annotation and initial experiments". In: *International Workshop on Machine Learning for Multimodal Interaction*. Springer. 2006, pp. 200–211.

[12] Gokhan Tur et al. "The CALO meeting assistant system". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6 (2010), pp. 1601–1611.

[13] Matthew Purver et al. "Detecting and summarizing action items in multi-party dialogue". In: *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*. 2007, pp. 18–25.

[14] Elizabeth Shriberg et al. *The ICSI meeting recorder dialog act (MRDA) corpus*. Tech. rep. INTERNATIONAL COMPUTER SCIENCE INST BERKELEY CA, 2004.

[15] Fan Yang, Gokhan Tur, and Elizabeth Shriberg. "Exploiting dialogue act tagging and prosodic information for action item identification". In: *2008 IEEE*

*International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2008, pp. 4941–4944.

[16] Matthew Frampton et al. "Leveraging minimal user input to improve targeted extraction of action items". In: *LONDIAL 2008 the 12th Workshop on the Semantics and Pragmatics of Dialogue*. 2008, p. 108.

[17] Gabriel Murray and Steve Renals. "Detecting action items in meetings". In: *International Workshop on Machine Learning for Multimodal Interaction*. Springer. 2008, pp. 208–213.

[18] Iain Mccowan et al. "The AMI meeting corpus". In: *Int'l. Conf. on Methods and Techniques in Behavioral Research* (2005).

[19] Kishan Sachdeva, Joshua Maynez, and Olivier Siohan. "Action Item Detection in Meetings Using Pretrained Transformers". In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2021, pp. 861–868.

[20] Pei-Yun Hsueh and Johanna D Moore. "What decisions have you made?: Automatic decision detection in meeting conversations". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. 2007, pp. 25–32.

[21] Raquel Fernández et al. "Modelling and detecting decisions in multi-party dialogue". In: *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. 2008, pp. 156–163.

[22] Yun-Nung Chen and Dilek Hakkani-Tur. "AIMU: Actionable Items for Meeting Understanding". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, pp. 739–743.

[23] Andreas Stolcke et al. "Dialogue act modeling for automatic tagging and recognition of conversational speech". In: *Computational linguistics* 26.3 (2000), pp. 339–373.

[24] Rajdip Dhillon et al. *Meeting recorder project: Dialog act labeling guide*. Tech. rep. INTERNATIONAL COMPUTER SCIENCE INST BERKELEY CA, 2004.

[25] Mark G Core and James Allen. "Coding dialogs with the DAMSL annotation scheme". In: *AAAI fall symposium on communicative action in humans and machines*. Vol. 56. Boston, MA. 1997, pp. 28–35.

[26] George Yule. *The study of language*. Cambridge university press, 2022.

[27] James Allen and Mark Core. *Draft of DAMSL: Dialog act markup in several layers*. 1997.

[28] Alexander Ratner et al. "Snorkel: Rapid training data creation with weak supervision". In: *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*. Vol. 11. 3. NIH Public Access. 2017, p. 269.

[29] Alexander Ratner et al. "Training complex models with multi-task weak supervision". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 4763–4771.

[30] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[31] Alvin E Roth. "Introduction to the Shapley value". In: *The Shapley value* (1988), pp. 1–27.

[32] L Shapley. "7. A Value for n-Person Games. Contributions to the Theory of Games II (1953) 307-317." In: *Classics in Game Theory*. Princeton University Press, 2020, pp. 69–79.

[33] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.
[34] Kenneth Moreland. "Diverging color maps for scientific visualization (expanded)". In: *Proceedings in ISVC*. Vol. 9. Citeseer, pp. 1–20.
[35] Jane Edwards. *nd The ICSI Meetings Corpus: Transcription Methods*.
[36] Susanne Burger, Victoria MacLaren, and Hua Yu. "The ISL meeting corpus: the impact of meeting type on speech style". In: *INTERSPEECH*. 2002.
[37] Susanne Burger. *ISL meeting Transcripts Part 1*. URL: https://catalog.ldc.upenn.edu/LDC2004T10.
[38] Susanne Burger and Z Sloane. "The isl meeting corpus: Categorical features of communicative group interactions". In: *NIST Meeting Recognition Workshop*. 2004.
[39] Jean Carletta. "Assessing agreement on classification tasks: the kappa statistic". In: *arXiv preprint cmp-lg/9602004* (1996).
[40] Ron Artstein and Massimo Poesio. "Inter-coder agreement for computational linguistics". In: *Computational linguistics* 34.4 (2008), pp. 555–596.
[41] Adam Janin et al. "The ICSI meeting project: Resources and research". In: *Proceedings of the 2004 ICASSP NIST Meeting Recognition Workshop*. 2004.
[42] Ji Young Lee and Franck Dernoncourt. "Sequential short-text classification with recurrent and convolutional neural networks". In: *arXiv preprint arXiv:1603.03827* (2016).
[43] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.
[44] Carol Peters, Martin Braschler, and Paul Clough. *Multilingual information retrieval: From research to practice*. Springer.
[45] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.
[46] Thomas Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (2019).
[47] Joshua Ainslie et al. "ETC: Encoding long and structured inputs in transformers". In: *arXiv preprint arXiv:2004.08483* (2020).
[48] Tianyi Zhang et al. "Revisiting Few-sample BERT Fine-tuning". In: *International Conference on Learning Representations*. 2020.