A Pipeline for Extracting Multi-Modal Markers for Meaning in Lectures

Johannes Ude[1], **Bianca Schüller**[2], **Rebekah Wegener**[2], **Jörg Cassens**[1]
[1] University of Hildesheim, [2] RWTH Aachen University
udejoh@uni-hildesheim.de, bianca.schueller@rwth-aachen.de,
rebekah.wegener@ifaar.rwth-aachen.de, cassens@cs.uni-hildesheim.de

Abstract

This article introduces initial concepts for a context sensitive computing pipeline to *detect* multimodal markers for meaning from video and audio data, to *notify* the audience of markers of importance and then to *classify* sequences of a recorded video into segments by content and importance in order to *summarise* the content as video and audio and in other modalities. In this paper, we first consider the linguistic background, then show the input data for the pipeline. Finally, we outline the concepts which are to be implemented in each step of this pipeline and discuss how the evaluation for this pipeline can be achieved.

1 Introduction

The summarisation of multimodal utterances is an important area of research in linguistics, natural language processing, and multimodal interaction. Summarisation of text is a difficult task in itself and the methods used vary widely depending on the purpose or function of the summarisation. Most recent work in natural language processing now integrates lexical, acoustic/prosodic, textual and discourse features for effective summarisation [Maskey and Hirschberg, 2005]. Only recently, however, are behavioural features being taken into consideration [Hussein *et al.*, 2016] and here only to summarise the movement in a video. Behaviour is frequently under-utilised as a modality because it is treated as a contextual footnote to speech. However, it can be equally meaning bearing and can often signal meaning prior to verbalisation [Butt *et al.*, 2013]. For additional supporting arguments see Lukin *et al.* [2011] and Cartmill *et al.* [2007].

But meaning making is most often multi-modal, including aspects of behaviour, and it is this multi-modality that we make use of in outlining a model for an automatic and context dependent note-taking system for academic lectures (see Wegener and Cassens [2016] for an overview of the research program). Drawing on semiotic models of gesture and behaviour, linguistic models of text structure and sound, and a rich model of context, we argue that the combination of information from all of these modalities through data triangulation provides a better basis for information extraction and summarisation than each alone.

Further, we suggest that by using a rich model of context that maps the unfolding of the text in real time with features of the context, we can produce query driven summarisation. This paper focuses on the proposed computational pipeline for processing the different input data streams. The context of the research presented here is to discuss the different components needed for implementing a functional prototype of the system.

The concept of using different modalities for summarisation of videos has been used in a number of other works before, see for example Maskey and Hirschberg [2005], where an acoustic signal was used in addition to text in order to summarise. The authors used broadcast news as their proof of concept. As a classifier, Maskey and Hirschberg [2006] used a hidden markov model.

The first application domain we are looking at is academic lectures as these are largely monologic in nature, making them easier to work with computationally. They are also readily available as large corpora and represent a clear case where extracting important information is useful. Besides providing a working prototype, we also like to use the system to showcase the concept of "smart data", meaning that we make use of existing knowledge; on the one hand in order to model the necessary contextual parameters and, on the other hand, to be able to use a limited amount of data to learn and tune the computational model. We have successfully used this method before to model intention as expressed through behaviour [Kofod-Petersen *et al.*, 2009], see Butt *et al.* [2013] for some methodological background. In future research, a further application domain next to monologic discourse will be dialogic discourse, as for example telemedicine consultations. Both monologic and dialogic domains provide foundations for multi-participant dialogic and multilingual domains, such as business meetings or team work.

2 Method and Motivation

In order to explore how humans detect and classify important information, note-taking data from two studies was used. Both of them were based on a stimulus lecture from a recorded first-year computer science course at MIT. The lecture is the first lecture in that course and has a length of approximately 55 minutes. It can be divided into four main parts: 1. administrivia, 2. lecture, 3. examples, and 4. conclusion [Wegener, 2017]. The administrivia and lecture parts

consist of three sections each.

In the first study by Wegener *et al.* [2017], experts in computer science were asked to annotate the lecture transcript according to what they consider to be important information that students should take away from the lecture. As the lecturer was not asked personally, experienced computer scientists were asked for their opinion instead so that their annotations could act as a ground truth for measures of importance. It was found that the four experts who took part in the study largely agree on their notions of importance, which is why their annotations were combined into one. The extractive summary that they created was transferred to notes at a later point.

The second study from Schüller [2018] involved undergraduate and graduate students of computer science or mechanical engineering, both native and non-native speakers of English. They were asked to watch the recorded lecture and to take notes either by hand or by typing, depending on their preferences. Afterwards, they filled in a short survey that asked them for some demographic data and for information on their note-taking practices. Notes were collected from nine students in total, one of whom was a native speaker of English. Two students were native speakers of Albanian and the remaining ones were German. There was a balance of male and female participants, writing and typing, and there were slightly more graduate participants than undergraduates. All participants were competent users of English. Three participants spoke three, five participants spoke one, and the native speaker spoke no languages other than English. This study was done in order to compare what experts expect students to take notes on and what the students actually do take notes on, i.e. to compare the different notions of importance.

Initial results show that the notion of importance differs a lot between the experts and the students as well as among the students themselves, which can be seen in the number of words they consider relevant in the different sections. This difference provides a strong motivation for the development of a system that can guide students in identifying importance during during academic lectures. In total, the expert notes included 1775 words, while student notes included fairly even steps between 93 and 691 words. This is partly due to the fact that the experts did not have to take notes while watching the recording, but is also attributed to the proficiency they have in their field as well as their note-taking competence. Among the students, the native speaker and the participants who spoke four languages took more notes than the others. When looking at the different lecture sections, the lowest discrepancy between expert and student notes is in the welcome (1.1), administrivia (1.3), and summary (4.) sections, while the highest discrepancy is in the examples part, followed by the course goals (1.2) and the lecture part (2.). The variation seen in importance across different phases of the lecture suggests that a model of the generic structure of a lecture could be useful for the information extraction process.

While it is natural that experts consider more information to be important than what students take notes on, what needs to be regarded is the size and location of the gap between the experts' and students' number of words that is noted in the different phases of the lecture. Where the distance is the same, the students largely understand what the experts expect; but where the gap increases, there might be problems in capturing the students' attention. This gives the motivation for a system to alert students when important information is triggered so that their note-taking skills can improve. These notifications can also act as cues for an automated summarisation system to denote important parts of the lecture.

3 Linguistic background

The initial goal of this study is to get a summary that is similar in nature to that of the experts' notes automatically without looking at the transcript of the text. Previous research has shown that the important parts of the lecture can be identified consistently when analysing the video and audio [Schüller, 2018]. The important parts (targets) are indicated by the lecturer's behaviour (e.g. gestures, movement, gaze and visual target), their voice (e.g. prosodic markers, pitch, tone, loudness), and, of course, important markers can be found in the textual transcript of the audio if that is available (e.g. words such as "right", "so", "ok").

A marker is a specific pattern which is found in one of the three categories of data. We focus on markers that signal important parts of the lecture. This means whenever a marker of a specific type appears in a lecture, this small part of the lecture should be considered as important and therefore included in a summary.

By applying linguistic analyses from Systemic Functional Linguistics (SFL) [Halliday and Matthiessen, 2014] as well as research on note-taking triggers in Cognitive Linguistics and Psychology [Boch and Piolat, 2005] to the computer science lecture that was focused on in the experiments, markers that signal importance have been found. SFL was chosen as an approach because it looks at language as a system of choices and therefore offers useful tools for investigating how meaning-making works in texts. In the analysis, SFL, which looks at language in its social context, and cognitive linguistics, which looks at language from the perspective of language users, were combined in order to consider different perspectives for the detection of markers.

We differentiate between markers that act as flags and markers that are targets. Flags are multimodal markers that appear before the target text and therefore are a signal to pay attention to the following while targets are the multimodal markers that we try to extract. The types of multimodal markers for meaning that can be found below have already been observed in previous work carried out as part of this ongoing research and will be incorporated into the model:

Board-writing: Building up on work by Boch and Piolat [2005], board-writing was found to act as a significant target [Schüller, 2018]. Being a behaviour, board-writing can be found within the image data.

Pointing Gestures: Schüller [2018] found pointing gestures to be targets as well. Not triggering note-taking to an extent as high as board-writing does, they showed discrepancies between experts' and students' notes and could therefore be a marker that students tend to miss. Being a behaviour as well, it is also part of the image data.

Notes as the Lecturer's Visual Target: Appearing together with the textual marker of continuatives, the image data of the lecturer looking at his notes was found to be reliable in acting as a flag [Schüller, 2018].

Continuatives As it was mentioned above, when appearing together with the lecturer looking at his notes, the textual marker of continuatives acts as a flag.

'Needing', 'Wanting', 'Going' Wegener *et al.* [2017] discovered that certain kinds of process types or the use of the going-to future appear in the computer science lecture from the experiment quite frequently. They are textual markers that are targets.

Most commonly repeated words in lexical bundles: Taking word lists from Martinez *et al.* [2013], specific words from lexical bundles were found to match with the continuatives and process types that were mentioned above, making them further textual markers that are targets [Schüller, 2018].

It needs to be added at this point that the markers above differ largely in their ubiquity, or, in other words, that they are more or less dependent on the context in which they appear, like the lecturer and the topic. For example, while processes like 'needing' and 'wanting' are tightly connected to the lecturer of the computer science lecture that was focused, commonly repeated words in lexical bundles appear to be more universal within the domain of academic lectures.

The next step is to look at the technical parts and determine how these markers could be detected automatically. When possible, we will make use of existing tools that help us to get more information from audio and image data. After that, we want to segment the video into parts, mark important aspects, and generate a summary of important parts.

4 Input data

In the final system [Wegener and Cassens, 2016], which will be applied to live lectures, there will be two sensor streams that collect audio and image data. The audio data will go through an acoustic signal analyser on the one hand in order to get phonetic, prosodic, and text-level data; and on the other hand, it will go through a speech-to-text processor to get grammatical, lexical, and cohesion data. The image data will go through gesture recognition to get behavioural, gesture, and micro-gesture data. Together, these form the multimodal ensembles that are used in combination with models of context and generic structure to detect important information.

The training data consists of the video of the lecture (consisting of audio and image data) and the notes from experts. The expert notes will not be available during deployment of the system. The use of the annotation data is twofold: firstly, the expert notes are used for benchmarking the classification step of the pipeline. Because we want a summarising pipeline, the automatic summarisation output should ideally be comparable to the expert notes.

Secondly, the expert notes are a ground truth for importance that is used to help us with the training data. The manual annotation by experts will denote those parts of the video where markers of importance should show up, if they exist. So instead of learning from data alone, we already know where and when in the data to look for these markers. In essence, without the expert data, we do not know whether a marker is really signifying importance or the time and duration of importance. Neither do we know which parts should be included in the summary, etc., so all of these aspects would have to be learned and validated if the manual annotation was not available.

In our proposed system the computing pipeline to *detect* multi-modal markers for meaning to *classify* video parts by whether they belong to a summary or not can be learned from both manual annotation and data. For deployment and evaluation of the finished system, just the video and audio of the lecture is given.

Another input type, not in the sense of data, but in the sense of a model, is knowledge of linguistic research as a configuration. We already know which markers exist and what meaning they hold. In this pipeline, we use multi-modal markers acting as flags or targets. So the markers signal the parts of the lecture which are considered important.

Furthermore, the configuration can hold a generic structure potential (GSP) model for lectures [Hasan, 1994]. A generic structure potential is a statement of the likely structure of a context. A generic structure however does not mean that there will not be variation. The GSP-Model can be used to determine a weight for each marker in a specific time interval of the lecture. For example at the beginning of the lecture there is just greetings and organizational information. Even when a marker is detected in these parts, the weights could modify whether this marker should be used or not.

5 Pipeline

The overall system will work in two phases. The first one is the learning phase with the given expert notes as a ground truth. In this step, the system learns a configuration based on the expert notes. The second phase is the production phase without the expert notes, but with a generalized configuration. We separate the system into these two phases because the expert notes are a costly factor. Therefore a generalized configuration to classify all lectures is what we want to get.

The main part of the pipeline that deals with identifying the markers themselves and evaluating their meaning potential for the text consists of at least two steps. These steps are *detect markers* and *classify video segments*. Before this pipeline starts, preprocessing steps are necessary. For the user of the overall system, these tasks represent background tasks. Only the input and the output are seen by the user as the user is handing in the video information and is retrieving a summary or a notification. The architecture of a pipeline seems to be a natural fit, because the output of each step acts as the input for the subsequent step. Therefore, each step is a computing unit which is responsible of a certain task. In the different steps of the pipeline, processing even of the same type of input can be handled by different units. For example, the same type of classifier could be implemented both as a hidden markov model, a bayesian classifier or a neural networks, in which case it might make sense to combine all types of clas-

sifiers in an ensemble process. In this sense, the pipeline can be thought of as a directed acyclic graph, with the different input-output-transformation forming the overall pipeline.

Figure 1 shows how the specific computing units in the overall pipeline are intended to fit together.

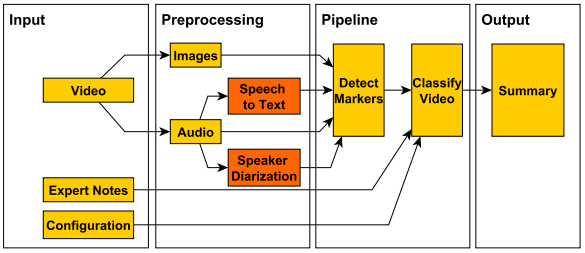

Figure 1: Computing Pipeline

At this stage of the development, the pipeline is restricted to processing recorded lectures only. Because lectures are a restricted and well described context, the meaning of behavioural patterns is more clearly visibly. An example would concern the field, tenor and mode of the situations, where **field** is *"the nature of the social activity..."*, **tenor** is *"the nature of social relations..."*, and **mode** is *"the nature of contact..."* [Hasan, 1999]. As outlined by Wegener and Cassens [2016], we have a specific mode (lecture), a specific tenor (student and lecturer), and a specific field (introduction to computational thinking), as well as a definable material situational setting (the sloping auditorium, with multiple chalk boards). This makes it easier to identify markers.

5.1 Preprocessing Video

The video data is preprocessed to split the video data into audio and image, to generate a transcript of the audio data and to identify when the lecturer is speaking. The video should be split into images and audio because we only have marker detectors for one modality at a time. For the other preprocessing, we use additional software. To generate a transcript we will use a speech-to-text generator like Dragon NaturallySpeaking. For the speaker diarization, to identify whether the lecturer or someone else is speaking, we will use the LIUM software [Meignier and Merlin, 2010]. Under many circumstances, only the lecturer would have the microphone, which would negate the necessity for speaker diarization. However, like in this particular instance, the microphone can be shared by two or more lecturers. If we know that the main speaker is not speaking, we could modify the weight so that such parts will not be included in the summary. Additionally, speaker diarization represents an important preprocessing step in dialogic speaking situations, meaning that considering speaker diarization already now will make it easier to apply the system to different domains in the future.

In any case, speaker diarization is not a single processing step but includes useful subtasks such as the detection of changes in prosody and loudness. In our experiments, speaker diarization sometimes misclassified the main speaker. Further analysis showed that features leading to this misclassification could be used elsewhere within the acoustic marker detector. The speech-to-text function is optional as well. Of course the textual marker detectors are dependent on the speech-to-text generator, but during deployment, the systems should work without the textual markers.

5.2 Detecting Markers

The detection of markers comprises the first step in the pipeline and is done with the video (image and audio) and transcript data. The transcript data would usually come from the audio data though the speech-to-text processor, but in this particular case, the transcript was created and patterned into clauses manually. To detect markers, we want to develop several kinds of detectors which analyse the video and transcribe the text. We can categorize the detectors into audio, including text, and image. For each of these groups, we will develop marker detectors. Thus this step is only responsible for detecting a marker and returning a confidence for a certain marker. This step is not responsible for computing whether a marker carries *importance* or another meaning. The meaning is already interpreted by the linguistic modelling work conducted in other stages of the project.

The image marker detector will be developed with OpenPose [Cao *et al.*, 2016]. This tool can detect a human skeleton in an image. With OpenPose, we want to identify three kinds of markers. We want to identify a) whether the lecturer is looking at their notes, b) pointing at the board and/or c) writing onto the board. We know that these marker types signal importance in the lecture [Schüller, 2018].

The audio marker detectors will focus on prosody and loudness. In this case, we have to apply a machine learning tool to identify the markers. Preliminary research shows that there are prosodic and loudness patterns which can be found when the lecturer switches to a new topic and when they emphasize different bits of information.

As for the textual data, the work of Wegener *et al.* [2017] shows that certain keywords (e.g. the use of 'needing', 'wanting', and 'going' by the lecturer of the computer science lecture) identify important parts in the lecture, too. Therefore, a textual marker of this type will be generated for sets of these words. Another marker to identify is whether the lecturer is using continuatives like 'so', 'ok', 'all right'. These represent flags that signal topic shifts [Schüller, 2018].

The detectors described so far are just examples. The pipeline should provide a plug-in architecture, where several marker detectors can be added. To this end, a programming interface which is to be used by the detectors has to be defined. This also makes it possible to have multiple extractors for the same marker, as described above. The configuration of the classification step defines how the markers should be combined to produce a contextually relevant summary.

5.3 Classifying the Video with the Detected Markers

The different kinds of data input for this step are used to classify the video. We call this *data triangulation* because of the combination of image, audio and textual data. This step is only responsible for classifying the segments of the video which are considered to be important with the help of the previously detected markers. Therefore this step has as an input

the markers of the detecting step. Because we only have the markers for importance, we can only classify segments of the video as important and not important.

This part of the pipeline should be highly configurable because the markers have to be aggregated to map segments of the video in the two classes: *used-in-summary* and *not-used-in-summary*. So far, the following approaches have been considered: Marker Overlapping Analysis, Neural Networks, Linear Classification.

The Overlapping Analysis will visualise the existing video and annotate it with the markers. Then the markers will be compared with the expert input. We analyse where and which types of markers concur with the expert notes in the summary. If a marker is in line with the expert notes, then this type of marker could be used for summarisation. The interval length is configured with the same method. The configuration is made by a linguist who configures for each marker whether the part of the video belongs to an extractive summary and, if so, how long the segment should be. The visualisation for the markers will look similar in principle to figure 2. The configuration is mostly exploratory for linguists.

The configuration of the classification can vary in complexity. A simple way would be to look at just a single marker type and use it directly for the summariser. A more complex configuration could be using multiple marker types and their confidence to model a function which estimates the class. Because of the plug-in architecture, it is possible that there are multiple markers of the same type, too.

An example of a classification could be to use the marker "looking at notes" together with "prosody and loudness". Then we could build a configuration that works such that if the lecturer is looking at the notes right after raising his voice, this scenario could be used as a signal of importance.

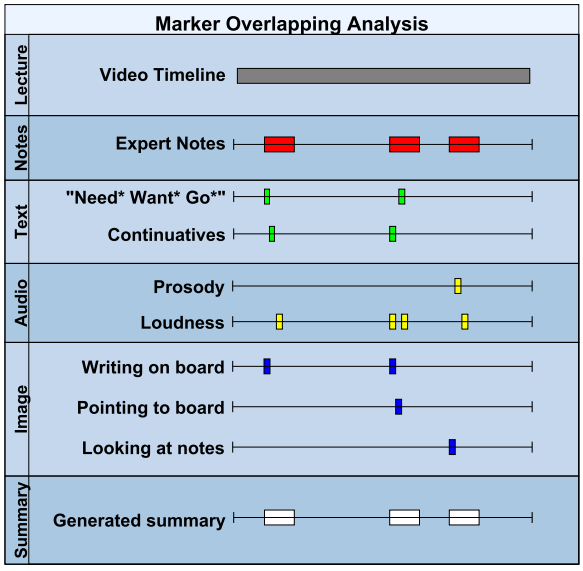

Figure 2: Classify Video with Marker Overlapping Analysis

Given the Marker Overlapping Analysis, we can add some more types of markers which are not already found to be flags or targets. These markers will then be explored iteratively with the same method of analysis. If they seem to co-occur with the expert notes as well, the new marker types could be used to improve the quality of the summary.

5.4 Using Classified Video Segments for Summarisation or Notification

The detected markers can be used to automatically summarise the lecture. In that case we use the time information from the important video parts and cut them into a summary. Figure 2 shows a preview of the generated video. Ideally, the markers could be used for more than just a summary. For example, the video could become searchable. Then a query could be used to search every part where the lecturer is writing on the board. The aim is to have a real time processing pipeline. In that case the students could be notified directly while the lecture is being recorded.

6 Development methodology and evaluation

The methodology for the programming part is a feature driven development [Coad *et al.*, 1999]. First, the image, audio and textual marker detectors will be implemented. Only after this step is done, we know what input data exists for the classification step. Then, a server will be implemented. We want to use a dedicated server because some marker detectors have high computational complexity. Therefore, it could be beneficial if the computation load can be separated to multiple physical client machines. After that, a graphical user interface (GUI) will be implemented to primarily provide the Marker Overlapping Analysis. While the choice of technology can still be changed, the GUI will likely be implemented with the eclipse RAP framework. This implies that the server and the GUI will be implemented in the programming language Java. Java is chosen because programming experience for this language exists. Tools and frameworks (Python comes to mind in particular for NLP functions) will be used through Java language bindings or wrappers.

For the first evaluation, a second video of the same lecturer and a lecture in the same semester should be summarised with this tool and the previous configuration. It is plausible that both the configuration and the models learned are depending on the individual lecturer, so the first evaluation will need to take that into account. Ideally, the same experts that provided the training notes would analyse the output and judge the quality of the summary.

A second evaluation scenario will be comparing the marker overlapping analysis with neural networks or linear classification, but using the same lecturer again. In this second evaluation, a metric could be used to identify the quality. For example, how do the experts judge the quality of the different configurations, how well does the automatic summariser agree with expert annotations.

Third and last, the tool needs to be tested with a) different lecturers and b) different topics. This is delegated to future work to improve the usefulness of the deployed tool.

A use case scenario could be that a student is learning for an exam. Then he or she wants to watch the most significant parts of a lecture as preparation for the exam. This system

could provide an extractive summary for the student. The student just has to upload the video and the pipeline is building a summary based on a configuration. Then the student downloads the shortened video.

However, since note taking and summarisation are important skills to be learned by students, our system could also be integrated with academic writing support systems to facilitate learning to write good summaries.

7 Summary

We have outlined a research program to build and evaluate a pipeline for extracting multi-modal markers for meaning in lectures. The overall architecture has been described and the dependency on existing tools outlined. Finally, possible evaluation methods have been described.

While implementation of the prototype has started, we are still in the early stages of realisation. The overall architecture has been defined, but e.g. which machine learning methods or tools are to be used primarily has not been finalised. This paper is insofar a concept paper, but it is also an incremental update on our previously published work on the underlying theory. The next steps are implementing the pipeline as a whole and performing the evaluations outlined.

8 Acknowledgements

We would like to thank the two host institutions, the University of Hildesheim and RWTH Aachen University, for making cross disciplinary student participation in this project possible. The first author is currently doing his master thesis on the project, the second author participated in the RWTH Undergraduate Research Opportunities Program (UROP) and did her bachelor thesis on the topic.

References

Françoise Boch and Annie Piolat. Note taking and learning: A summary of research. *The WAC Journal 16*, pages 101–113, 2005.

David Butt, Rebekah Wegener, and Jörg Cassens. Modelling behaviour semantically. In P. Brézillon, P. Blackburn, and R. Dapoigny, editors, *Proceedings of CONTEXT 2013*, number 8175 in LNCS, pages 343–349, Annecy, France, 2013. Springer.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. *arXiv:1611.08050 [cs]*, November 2016. arXiv: 1611.08050.

John Cartmill, Alison Moore, David Butt, and Lyn Squire. Surgical teamwork: systemic functional linguistics and the analysis of verbal and non verbal meaning in surgery. *ANZ Journal of Surgery*, pages 925–929, 2007.

Peter Coad, Eric Lefebvre, and Jeff De Luca. *Java Modeling in Color with UML*. Prentice Hall, Upper Saddle River, NJ, 1999.

M. A. K. Halliday and Christian Matthiessen. *An introduction to functional grammar (4th ed.)*. Routledge, London/ New York, 2014.

Ruqaiya Hasan. Situation and the definition of genre. In Allen Grimshaw, editor, *What's going on here? Complementary Analysis of Professional Talk: volume 2 of the multiple analysis project*. Ablex, Norwood N.J., 1994.

Ruqaiya Hasan. Speaking with reference to context. In Mohsen Ghadessy, editor, *Text and Context in Functional Linguistics*. John Benjamins, Amsterdam, 1999.

Fairouz Hussein, Sari Awwad, and Massimo Piccardi. Joint action recognition and summarization by sub-modular inference. In *ICASSP*, 2016.

Anders Kofod-Petersen, Rebekah Wegener, and Jörg Cassens. Closed doors – modelling intention in behavioural interfaces. In Anders Kofod-Petersen, Helge Langseth, and Odd Erik Gundersen, editors, *Proceedings of the Norwegian Artificial Intelligence Society Symposium (NAIS 2009)*, pages 93–102, Trondheim, Norway, November 2009. Tapir Akademiske Forlag.

Annabelle Lukin, Alison Moore, Maria Herke, Rebekah Wegener, and Canzhong Wu. Halliday's model of register revisited and explored. *Linguistics and the Human sciences*, 2011.

Ron Martinez, Svenja Adolphs, and Ronald Carter. Listening for needles in haystacks: how lecturers introduce key terms. *ELT journal 67(3)*, pages 313–323, 2013.

Sameer Maskey and Julia Hirschberg. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Interspeech 2005*, pages 621–624, 2005.

Sameer Maskey and Julia Hirschberg. Summarizing speech without text using hidden Markov models. In *Proceedings of the NAACL HLT, Companion Volume: Short Papers*, pages 89–92. ACL, 2006.

Sylvain Meignier and Teva Merlin. LIUM SpkDiarization: An open source toolkit for diarization. In *CMU SPUD Workshop*, Dallas, Texas, 2010.

Bianca Schüller. Understanding the identification of important information in academic lectures: Linguistic and cognitive approaches. Bachelor thesis, RWTH Aachen University, 2018.

Rebekah Wegener and Jörg Cassens. Multi-modal markers for meaning: using behavioural, acoustic and textual cues for automatic, context dependent summarization of lectures. In J. Cassens, R. Wegener, and A. Kofod-Petersen, editors, *Proceedings of the Eighth International Workshop on Modelling and Reasoning in Context*, 2016.

Rebekah Wegener, Bianca Schüller, and Jörg Cassens. Needing and wanting in academic lectures: Profiling the academic lecture across context. In Phil Chappell and John S. Knox, editors, *Transforming Contexts: Papers from the 44th International Systemic Functional Congress*, Wollongong, Australia, 2017.

Rebekah Wegener. Instantiation in modelling multimodal communication: Challenges and proposals, part 1. 2017. Presentation at the European Systemic Functional Linguistics Conference in Salamanca, Spain.