Automatic Summarization of Meeting Data: A Feasibility Study.

Conference Paper · January 2004

Source: DBLP

CITATIONS
31

READS
42

3 authors, including:

Wessel Kraaij
Leiden University
215 PUBLICATIONS **5,233** CITATIONS
SEE PROFILE

Stephan Raaijmakers
TNO
42 PUBLICATIONS **354** CITATIONS
SEE PROFILE

Automatic Summarization of Meeting Data: A Feasibility Study

Anne Hendrik Buist, Wessel Kraaij and Stephan Raaijmakers

TNO TPD

Abstract

The disclosure of audio-visual meeting recordings is a new challenging domain studied by several large scale research projects in Europe and the US. Automatic meeting summarization is one of the functionalities studied. In this paper we report the results of a feasibility study on a subtask, namely the summarization of meeting transcripts. A Maximum Entropy based extractive summarization system using a mix of 15 features improved the performance of a baseline system selecting all utterances longer than 10 words with 20% (F-measure). However, stronger contextual awareness seems to be necessary in order to reduce the precision of the summarizer. The study required the creation of reference extractive summaries, which is documented in the paper.

1 Introduction

As speech recognition of broadcast news is becoming more mature, research is moving into types of speech that are more challenging. One such area is conversational speech. Initially telephone conversations were studied but more recently attention moved to meeting recordings. Indeed, interesting applications can be foreseen if automatic speech recognition (ASR) performance of conversational speech could be boosted to reach the same level of accuracy as for broadcast news. In the EU projects M4 (M4 2002b) and AMI (M4 2002a), meetings are recorded in a "smart meeting room" using multiple synchronized cameras and microphones (de Jong 2004). The key application developed in these projects is the "meeting browser", which facilitates users to search and browse meeting recordings. For this purpose the raw data is processed by multimodal analyzers that recognize "meeting actions" (e.g. discussion, presentation etc.) and perform a shallow semantic analysis. During the AMI project, topic segmentation and summarizing functions will be developed.

This paper describes a feasibility study of using machine learning techniques for extractive summarization of meetings. It is clear, that summarization is important for a meeting recording archive, since it will help users to find relevant meetings and (if the summary is linked to the recordings) to locate salient fragments of recordings for viewing. Reading summaries is much more time-efficient than listening to (or viewing) a recording. Also reading/searching the raw transcripts is not desirable, as these contain a lot of backchannels, elaborations and side topics which do not contribute to the content.

The area of summarizing dialogues or meetings has not yet been explored by many researchers. Some groundbreaking work has been done by Klaus Zechner and Alex Waibel, who created a dialogue summarizer DiaSumm (Zechner and Waibel 2000), which features include turn-linking, topic segmentation and information condensation. In (Zechner 2002), Klaus Zechner gives an overview of work done in this area. While one may think that the process of summarizing meetings is similar to the sum-

marization of news articles or broadcast news, it appears to be very different in practice. In contrast to written publications, sentence boundaries are very hard to discern, because conversations contain many disfluencies. Additionally, the information density in meetings is much lower than in news, which in essence is highly condensed information.

Moreover, trivial automatic summarization of news items is often facilitated by the fact that news articles have titles and lead text, and taking just the few initial lines of a news item already yields a good summary. For meeting summarization, such metadata is usually not available.

Our goal is to create a system which can extract all important topics from a recorded meeting and present them in an understandable way, thus creating a readable summary of the meeting. The summary should then be linked to the audio in a multimedia player, which allows for browsing the summarized meeting both audibly and textually. For an initial feasibility study, we investigated whether machine learning techniques that had proved to be successful for the summarization of broadcast news, could be adapted for the meetings domain. A fully functional automatic meeting recording summarization system would be highly complex, since it combines a.o. high quality speech recognition, speaker segmentation, utterance segmentation, dialogue act interpretation, domain knowledge with summarization techniques, each of which components are not sufficiently mature yet. Therefore, we performed a limited study into the effectiveness of structural and lexical properties of utterances as features based on manual meeting transcripts.

The rest of this paper is organized as follows: section 2 describes our approach to summarization using Maximum Entropy models, section 3 describes the corpus and annotation procedure, sections 4 and 6 describe the features used for the ME models and the experiments. The paper is finished with our preliminary conclusions and ideas for future work.

2 Maximum Entropy based summarization

Even though automating abstractive summarization is the goal of summarization research, most practical systems are based on some form of extractive summarization. Extracted sentences can form a valid summary in itself or form a basis for further condensation operations. Furthermore, evaluation of extracted summaries can be automated, since it is essentially a classification task.

During the DUC 2001 and 2002 evaluation workshops, TNO developed a sentence extraction system for multi-document summarization in the news domain. The system was based on a hybrid system using a Naive Bayes classifier and statistical language models for modeling salience. Although the system exhibited good results (Kraaij, Spitters and van der Heijden 2001, Kraaij, Spitters and Hulth 2002), we wanted to explore the effectiveness of a Maximum Entropy (ME) classifier for the meeting summarization task, as ME is known to be robust against feature dependencies. Maximum Entropy has also been applied successfully for summarization in the broadcast news domain (Osborne 2002).

2.1 Maximum Entropy Modeling

The maximum entropy model framework can classify information that comes from many different sources. The data the model trains on can be described as a vector of features $\{f_1, \ldots, f_k\}$. One of those features could be as follows:

$$f_j(b,c) = \begin{cases} 1 & \text{if SegmentLength}(c) = Long \;\&\; b = \texttt{good} \\ 0 & \text{otherwise} \end{cases}$$

This feature teaches the model that a segment that is *long* (for instance, longer than 20 words), is relevant to the summary, and the probability $p(\texttt{good}, c)$ will increase. b can either be `good`, which means it should belong in the summary, or `bad`. These features can be of a complex type, and we can use prior knowledge about what information is important for classification. Each feature that is in the model corresponds to a certain constraint. Then from all the models that satisfy the constraints:

$$\sum p(b,c) f_j(b,c) = \sum \tilde{p}(b,c) f_j(b,c), 1 \leq j \leq k$$

the one that maximizes the entropy $H(p)$ is chosen:

$$H(p) = -\sum p(b,c) \log p(b,c)$$

$\tilde{p}(b,c)$ is the observed distribution of features found in the training data. Choosing this maximum entropy model is a method to preserve as much uncertainty as possible: we want to have as little unjustified constraints of information as possible (Manning and Schütze 1999, Ratnaparkhi 1996): when the model finds a segment to be `good` or `bad`, we know it has found sufficient evidence for this outcome.

This model has proved to be very useful for many natural language processing tasks, including sentence detection, named entity recognition and part of speech tagging. We have used the OpenNLP implementation of the model, which is freely available on the web (Baldridge, Morton and Bierner 2001).

2.2 Applying ME for meeting summarization

As for any application of supervised machine learning techniques, a corpus is required that is annotated with ground truth information. For our experiments, the ICSI Meeting Recorder corpus (we will call this ICSI corpus from now on) was used, available from LDC. The M4 project did not have an extensive corpus available, especially not with manual transcriptions of the audio. At the time of the experiments, no ground truth extractive summarization data was available, so we manually annotated several meetings from the ICSI corpus. The annotation procedure is described in section 3. Subsequently several lexical and structural features were selected (some features that have been successfully applied in the broadcast news domain were evaluated as well). A subset of the annotated data was used for training the ME classifier. Training itself was based on determining (feature=value) pairs for all features for each sentence. The feature selection process is described in more detail in section 5.

3 Annotation procedure

A significant amount of time of the feasibility study was spent on producing training material for the automatic summarizer. Unfortunately, we did not have time nor the people let multiple people annotate the same meetings. We chose to have one annotator instead, who did his best to annotate 6 meetings, approximately an hour each. By doing the annotation in multiple steps, backtracking and correcting, we tried to guarantee that the quality of each summary was decent. Annotating the meetings took about 12 to 14 hours per meeting[1]. Approximately 22000 segments were rated in this way.

3.1 The ICSI Meeting Recorder corpus

The meeting corpus developed at ICSI, Berkeley (USA) consists of about 75 meetings recorded at ICSI. The meetings of several research groups at ICSI were recorded, so conversations have a highly technical focus. For each meeting, transcripts are available that contain both the start and end times, of words and speaker segments. In addition, the original audio recordings are available and several hand-annotated analyses of the corpus, e.g. of dialogue acts (provided in the MRDA corpus) and adjacency pairs[2].

3.2 Extract-based summaries

All segments of the six ICSI meetings were annotated for importance on a ternary scale. A segment in this context is a whole sentence or a part of it, spoken by one person. Sentences in the corpus were automatically cut off at some points, where the speaker would pause for a certain amount of time. Segments rated with 3 are highly relevant for the summary, while a 1 indicated that they are of little or no importance. Rating with a 2 indicates either an (ongoing) elaboration on the subject, or expresses doubt by the annotator regarding the importance of the segment.

3.3 Annotation method

In order to keep structure in the annotation, a few rules were followed while annotating. First of all, the annotator attempted to base summary annotations on just the text and specifically avoided to be biased by his knowledge of NLP techniques and the problems that specific utterances would pose. We chose to annotate the meetings in the MRT format.[3] During the actual annotation the following scheme was followed (for each meeting):

- Part I

[1] Meetings are often unstructured, and the audio can be very hard to perceive

[2] In conversations, many utterances are directed to evoke a natural response. E.g., complaints require apologies or maybe counter-complaints. This is called an adjacency pair.

[3] This is the XML format used by the ICSI corpus. The Meeting Recorder Dialogue Act (MRDA) corpus uses a different format, but that is deprecated.

1. Make a printout of all segments, preceded by the speaker of the utterance. Every segment on a new line.
2. Scan the printout to detect the topic.
3. Sequentially read the printout, rating segments on the fly. Backtrack when encountering a possible flaw (which could be e.g. incoherence, or things which seemed unimportant at first).
4. Listen to the audio when in doubt.
5. Import the handwritten segment ratings into the MRT file using a simple PERL script. While doing this, additional flaws in the summary were sometimes detected and in that case corrected.
6. Correct any mistakes made in importing by manually editing the MRT file.

- Part II

7. Make a printout of the important segments (rated 3). Calculate the percentage of the summarized portion over the whole meeting.
8. Recheck for fluency, understandability and other errors encountered. These were manually corrected again in the MRT file. When the percentage of extracted text (in words) was more than 30%, the summary was also more thoroughly checked for superfluous portions, and normalized. All corrections were also annotated on the original printout.

Accidently we used an older version of the ICSI corpus to annotate, so we had to re-rate all the meetings for the new format, because sentence boundaries had changed between versions (Sentences were cut off at different points). In this process, all segments were checked again quickly and some more corrected.

3.4 Evaluating importance

A crucial part in summarization is how to judge whether a segment is important or not. Other than in news articles or papers, things are said more than once in a meeting. It is hard to decide, when two utterances are almost equal, which one should make it into the summary. It is not good to include both, because the resulting summary would contain redundant parts. When evaluating, the automatic summarizer may prefer one of those two sentences above the other, for some reason, which might be the 'wrong' one. We found no solution for this, except that the annotator chose the most logical one in this case. For example, when the two sentences would be uttered by two different participants, one restating the former, most logical would be the original expression. In other, clearer cases where the segments differed a little, the more elaborate one was chosen, observing that the extra information was relevant.

When a participant starts a long series of utterances, selections were made on very clear points only. Even though a sentence might accidentally be understandable when removing a portion, it is not a wise thing to do.

Sometimes, when a segment consisted of a conjunction only, such as 'And um', these segments were also rated as good, because it would interrupt the flow of words

in the summary, when the segments are 'stitched' back together lateron. An exception is when such a segment contains only a backchannel: removing this does not harm the flow of a sentence so can be safely removed, resulting in a score of 1.

In some cases, a segment is only partly interesting. The whole segment was marked as important (3). This is one of the reasons that an abstract-based summary will probably be able to include the same information in a more condensed form.

What is important regarding the content may also differ on a per audience basis. The annotator assumed the summary to be a recollection of topics discussed in the meeting, trying to preserve as much information as possible.

3.5 Result of corpus summarization

The annotation eventually resulted in a rated corpus of 6 meetings, which incorporates approximately 6 hours of dialogue. The compression-rate in words is about 70%, and 80%-90% on the segment level. This difference is due to the fact that very long sentences are of most importance in a summary, and ultra-short ones are often non-salient. In addition to rating every segment, a topic segmentation was also performed: every meeting was annotated with the topics that are brought up during that meeting. This data was not (yet) actually used in the summarizer.

4 Experiments

4.1 Training and testing

The six hand-annotated meetings were divided into a training (4 meetings) and evaluation set (2 meetings). Training involved two steps: a feature extractor computed feature vectors with key=value pairs for each segment in the training set. Subsequently, the feature vectors (complemented with the truth data) were used as input to train a maximum entropy model. This model was applied to predict the salience value of segments of the test meetings. By counting how many segments in the testdata were correctly labeled `good`, the performance level of the model could be quantified. This performance was measured by the recall (how many of the total relevant segments are recognized correctly) and precision (percentage of relevant segments in relation to the number of segments labeled as `good`). In addition their harmonic mean (the F-measure) was computed (Van Rijsbergen 1979):

$$F - Measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Additionally, 10-fold cross-validation was used because of the relatively small dataset, and to create more stable results. The maximum entropy model tends to perform better when trained on a balanced set of good and bad examples, so the good examples in the training data were randomly oversampled. This process gets random good samples from the training set and adds them at the end. A 40% oversample rate makes sure that the good samples cover 40% of the training data. A 50% rate means a perfect balance between good and bad samples. The initial balance between good and bad

samples is approximately 17% good, 83% bad. Shown samplerates always indicate the percentage of good samples.

5 Feature generation and overview

In order to generate a good model, it is important to find as many strong features as possible. A feature can be something like frequent words or sentence length, or can for instance capture a relation to previous segments. Although we found many features that weighted towards being summary-specific, no strong features could be found. This was actually a bit of a surprise, because we initially expected that recycling the feature set of the TNO summarizing system that was developed for DUC2002 would generate a solid base. However, by finding many mediocre features it is still possible to achieve acceptable results. Table 1 provides a table with all the used features, a short explanation and the abbreviation that is used in the results.

SL	Sentence length: a segment is either ultra-short, short, medium or long.
MIN	The segment contains more than 3 words.
LW	Segment contains long words (more than 10 characters).
TF	Segment contains words that stood out in a TF-IDF approach.
FR	Segment has frequent words (own algorithm, many features).
FR1	Segment has frequent words (own algorithm, one feature).
FRB	Segment has frequent bigrams.
ACK	Segment is an acknowledgment.
CUE	Segment contains cue phrases.
CON	Segment connects with previous sentence (= same speaker and continues the sentence)
EMP	Segment is empty (no words).
SAM	Segment has the same speaker as previous one.
IMP	Segment features important speakers (3 features of the most prominent speakers).
DA	A number of features for some relevant dialogue acts (using the MRDA corpus).
DIG	Digit task.
TUR	A turn change took place: the next speaker is different from the last.
LON	A long turn: a speaker takes a turn over many segments, uninterrupted.
PC	Whether the previous segment is good. 2 features, go back 2 segments.

Table 1: Feature overview

Some features are actually a set of features which belong together: for instance, the SL feature set counts four features, of which only one can be true at the same time. This approach is needed for correct implementation by the Maximum Entropy model. The purpose and usage of some features may speak for themselves, but a few need additional explanation.

5.1 Sentence Length (SL)

This feature class is divided into four parts: ultra-short, which are segments that are only 10 characters long or less. There are many occurances of ultra-short segments which are often of no importance. The boundaries for 'short' are larger than 10 and smaller than 30 characters. 'Medium' is between 30 and 80 characters, and everything longer than 80 is considered 'long'. The measure is in characters instead of words especially for smaller segments: a segment like 'I like it' contains no information, while a sentence with three long words tends to be more important.

5.2 TF-IDF (TF)

This is an implementation of the TF-IDF information retrieval technique (Baeza-Yates and Ribeiro-Neto 1999) where words are highlighted that occur more often in the current meeting than the average in a corpus of similar meetings. Segments that contain document-specific words often contain valuable information for a summary. The feature this generates indicates whether such a word is in a segment. Only the top 20 words are selected.

5.3 Frequent words and bigrams (FR FR1 FRB)

A variation on the TF-IDF algorithm, this implementation uses the mean and standard deviation of word frequencies in a corpus to evaluate their importance. When tested as a single feature, it works slightly better than TF-IDF. The variant implements a feature for the 15 most important words to occur in the document, which results in 15 different features. This method gives the model many more features to train with. The bigram version implements the same for significantly important sets of two words. To ensure a clean list of important words in the unigram variants, a closed word stoplist is used: the use of particular closed words are not a sign of importance, they only depict the style of the speakers.

5.4 Cue phrases

Although this feature is not fully implemented, it captures the idea: people tend to use certain phrases to announce something important. To find some phrases that stood out from the good segments, we experimented with likelihood ratio statistics, which can distinguish specific terms (or phrases) by comparing occurrence frequency with a background corpus. When using bad segments as the background corpus, one would be able to see which phrases are specific for a summary. Unfortunately, this did not work as well as we hoped: the results were words/phrases that were not obviously important. We handpicked a number of cue phrases from the top list. When those occur in a segment, this feature is triggered.

5.5 Linking to the previous segment (PC)

This can be done by doing a run on the generated list of feature vectors, and add new features that determine if the previous segment was rated good. This results in two features, one for the last, and one for the second-last segment. It adds a new layer of uncertainty, because the features are based on the assumption that the segments were correctly labeled in the first run.

5.6 Important speakers (IMP)

By comparing speaker times in the rated (test)corpus, we found that the most frequent speakers also say more important things. This behavior is captured in three features that name the first, second and third longest speakers. This is calculated by summing up the time of speech for each individual speaker.

5.7 Dialogue Acts (DA)

A dialogue act is metadata about a segment that informs about the intention of the speaker. Examples are grabbing the floor[4], making a statement, or asking a question. Every segment can contain multiple dialogue acts. When experimenting with them, we found that interesting segments were mostly statement-only and question segments.

We had the privilege to have access to the MRDA corpus, a set of meetings from the ICSI Meeting Recorder project where dialogue acts were hand-annotated. Unfortunately, because the format of the ICSI corpus was still changing, the MRDA corpus was not completely compatible with the newer ICSI format: segments were wrapped at irregular places, which prevented a clean remapping of the corpus. The eventual mapping is probably accurate for at least 90% of the segments, which makes it quite usable. Eventually an automatic DA tagger will be implemented.

6 Results

The system was trained with different feature combinations, because using the complete feature set rendered suboptimal results. A result was calculated for every single feature, after which the best result was selected. Then every feature combined with that best feature was evaluated again. This was done a few times. In table 2 the initial result per feature is shown. A number of features have no significant results because the model selected all segments to be important. The strength of these features is their combination with others.

We also experimented with oversampling percentages. Unfortunately, due to using random samples for oversampling, results differed between identical runs of the program: the deviance was approximately 1%. As expected, oversampling greatly improves the summarizing process, as can be seen in table 3. Because at 50% the last

[4] *Grabbing the floor* is interrupting the current speaker, in order to make a statement. Cp. *Holding the floor*, in which the speaker tries to keep the attention by connection his statements, while he prepares his words (e.g. with ‘and umm…’)

Feature	TF	FR1	FR	FRB	CUE	LW	ACK	DA	MIN
F-Measure	0.2	0.321	0.31	0.233	0.157	0.392	0.144	0.314	0.269

Table 2: Single feature results

Oversampling	-	20%	30%	40%	50%	60%	70%	80%
F-Measure	0.162	0.283	0.406	0.447	0.503	0.508	0.507	0.496

Table 3: Results for oversampling, the percentage shows the amount of good samples in the set. This is 17% without oversampling.

Feature set	*Recall*	*Precision*	*F-Measure*
Baseline (random)	0.199	0.169	0.182
Baseline (10 words)	0.448	0.362	0.401
All features on	0.641	0.379	0.476
All features - PC	0.653	0.381	0.482
TF FR1 FRB CUE (content features only)	0.228	0.348	0.276
LW SL LON ACK CON EMP SAM IMP DIG TUR DA MIN PC (no contentfeatures)	0.620	0.354	0.451
LW SL LON TF FR ACK CUE CON EMP SAM IMP DIG TUR (optimal feature set)	0.682	0.401	0.505

Table 4: Results with different feature sets, 50% oversampling

great increase was noted, this rate was used in further tests. With the increase of the samplerate, a sharp increase of recall is seen, extracting more segments than wanted from 50% onward.

For comparison, we also tested two baseline approaches. The first is a random extract of 10% of the segments from each test meeting. This is the absolute baseline, as there is no coherence or logic in this method. The second baseline selects those segments that contain more that 10 words. This is motivated by the fact that there are many backchannels and filtering these out is a simple first step to clean up a transcript for extractive summarization.

System	Our system	N-Top	Fung Ngai Chi-Shun	MEAD
Result	50%	41%	69%	59%

Table 5: Results for some document summarization systems. N-Top is a baseline that extract the n top sentences from each document, Fung and MEAD are both multiple document extraction based summarizers. Percentages are the amount of correctly extracted sentences. Single document summarizers achieve even higher percentages.

6.1 Inter annotator agreement

The optimal selection of features renders a F-Measure of 0.505, which is not a very high score. This relatively low performance may have a good reason, that cannot be easily solved: a paper by Mandar Mitra et al. (Mitra, Singhal and Buckley 1997) addresses the issue of human agreement. They let two people make a summary of the same text, where the persons had to choose critical paragraphs that were to be included. The overlap between these persons was only 46%, which means that they agreed only on 46% of the content of the final summary. This is a serious problem, because this means that only half of a summary's content is typical. While manually creating summaries, for every segment a choice has to be made, which in some cases inevitably leads to arbitrariness. For a machine learning model to be effective, it is necessary that the data is somehow consistent. Because we only had summaries made by one individual, we had no possibility of comparison.

6.2 Other issues

Unfortunately, there are no similar systems available that deal with meeting summarization to compare the results with. When compared to summarization of broadcast news or documents, performance is quite low (See table 5 for some results taken from (Fung, Ngai and Cheung 2003)). However, these are completely different types of content. One of the big differences is the nature of a meeting compared to news articles and other written documents: where articles always follow strict guidelines for publishing, meetings do not. A meeting has little structure in topic sentences or any placement of important segments. Where the first sentence of an article often contains the topic, this is rarely the case with meetings. Sometimes a meeting will start right away, other times there will be some chit-chat beforehand. It is very hard to discern between such conversations.

6.3 Screen output in SMIL

When a summary has been generated, it is possible to listen to an audio version using Realplayer. Realplayer supports SMIL (Synchronized Multimedia Integration Language), which is a XML markup language for audiovisual presentations. A SMIL version of the extract was produced, consisting of just the extracted segments in textual form (they were displayed as running text), with synchronous presentation of the

corresponding audio, thus skipping the material marked as unimportant. Segments are colorcoded for each individual, and the format also allows clicking in a topic index to skip certain parts. This system actually works very well and feels quite natural, even though segments are sometimes not completed. Because people involved in conversations often do not finish their sentence either, this does not lead to irritation.

7 Conclusion

The automatic sentence extraction system was able to improve a heuristic baseline system by about 20%, showing the effectiveness of the chosen features. Nevertheless, the absolute performance of the extractor is less than is expected for NLP classification problems. We feel that lower levels of performance (in terms of F-value) are not so surprising for summarization tasks, since it is well known that human annotators have a low level of agreement on a manual task. Overall, the system produces fairly readable summaries, even though no effort has been done to rephrase sentences. The bottleneck of the system is the lack of structure in meetings, and related to this the absence of good features.

Furthermore, the study gave some insight into the structure of meetings, showing some interesting features that could be used in further research. The approach to classify each segment individually, without looking at the context is obviously too naive. Still, our results can function as a reference baseline for comparison with future results.

8 Future work

To continue work on this matter, a new approach using lexical chains is investigated. Lexical chains are capable of pinning down topic hotspots in a document, and connecting the most important sentences. The use of lexical chaining can be implemented as a whole new method, or as a enhancement on the feature set of our current summarization system, e.g. by producing better (context based) estimates of which tokens are topical.

References

Baeza-Yates, R. A. and Ribeiro-Neto, B. A.(1999), *Modern Information Retrieval*, ACM Press / Addison-Wesley.

Baldridge, J., Morton, T. and Bierner, G.(2001), The maximum entropy framework, http://maxent.sourceforge.net/about.html.

de Jong, F.(2004), Disclosure of non-scripted video content: InDiCo and M4/AMI, *Proceedings of CIVR 2004*.

Fung, P., Ngai, G. and Cheung, C.-S.(2003), Combining optimal clustering and hidden markov models for extractive summarization, *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*.

Kraaij, W., Spitters, M. and Hulth, A.(2002), Headline extraction based on a combination of uni- and multidocument summarization techniques, *Proceedings*

of the ACL workshop on Automatic Summarization, Document Understanding Conference (DUC 2002), Philadelphia, USA.

Kraaij, W., Spitters, M. and van der Heijden, M.(2001), Combining a mixture language model and naive bayes for multi-document summarization, *Proceedings of the Document Understanding Conference*, Document Understanding Conference (DUC 2001), New Orleans, USA.

M4(2002a), Augmented multiparty interaction (ami), http://www.m4project.org/overview.html.

M4(2002b), Multi modal meeting manager (m4), IST-2001-34485 http://www.m4project.org/overview.html.

Manning, C. D. and Schütze, H.(1999), *Foundations of Statistical Natural Language Processing*, MA: MIT Press, Cambridge.

Mitra, M., Singhal, A. and Buckley, C.(1997), Automatic text summarization by paragraph extraction, *Mani and Maybury*, MIT Press, Cambridge, Massachusetts.

Osborne, M.(2002), Using maximum entropy for sentence extraction, *ACL 2002 Workshop on Automatic Summarization*.

Ratnaparkhi, A.(1996), A maximum entropy model for part-of-speech tagging, *in* E. Brill and K. Church (eds), *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Somerset, New Jersey, pp. 133–142.

Van Rijsbergen, C. J.(1979), *Information Retrieval, 2nd edition*, Dept. of Computer Science, University of Glasgow.

Zechner, K.(2002), Summarization of spoken language - challenges, methods and prospects.

Zechner, K. and Waibel, A.(2000), Diasumm: Flexible summarization of spontaneous dialogues in unrestricted domains, *Proceedings of COLING*, International Conference on Computational Linguistics (COLING), Saarbrucken, Germany.