

Automatic Speech Recognition on Meeting Recordings for Search and Question Answering

Maurice Silverio
maurice.silverio@student.uva.nl

Submitted in partial fulfillment for the degree of master of science

Master Information Studies
Data Science
Faculty of Science
University of Amsterdam

	Internal Supervisor
Title, Name	Dr. Maarten Marx
Affiliation	UvA
Email	maartenmarx@uva.nl

ABSTRACT

This research aims to improve information retrieval from recordings of Dutch city council meetings by proposing a new archival system that combines Automatic Speech Recognition (ASR), Information Retrieval (IR) and Question and Answering (QA) methodologies. To achieve this, Whisper from OpenAI is used for ASR, converting spoken audio into full transcripts. The IR component extracts relevant information from these transcripts using multiple models that optimize results based on the user's search complexity. Moreover, an interactive question-answering feature is introduced, which enables a more natural and engaging approach when retrieving information from within these recordings. In addition to Dutch city council meetings, the findings of this research illustrates how speech recognition and information search techniques can improve audio data retrieval. Enhancing the accessibility and usability of recorded content across various fields.

KEYWORDS

Automatic Speech Recognition, Information Retrieval, Question and Answering Systems

1 INTRODUCTION

Numerous city councils in the Netherlands utilize meeting software to live stream and archive their sessions [18]. Providing a valuable resource for residents to engage with their local governance. This archiving functionality allows users to access past council meetings, enabling them to stay informed about discussions and decisions that shape their communities. While the current archive feature enables access to past council meetings, it has become evident that certain limitations hinder residents from efficiently finding and extracting the information they seek. As a result, there is an identified need for an enhanced system that significantly improves search capabilities, streamlines navigation, and optimizes information retrieval from these archived meetings.

One significant limitation of the existing system is its dependence on metadata-based searches. Meaning only search using speaker names and the titles of meeting topics is available. Although this method can be helpful in certain situations, it often fails when users are trying to pinpoint specific speeches or discussions. Users commonly struggle to find relevant sections, especially without knowledge of the speaker's associated metadata or precise timestamps for the information they seek. This issue primarily stems from the fact that the search is restricted to metadata, which doesn't encompass the actual content of the recordings.

This research explores a multi-faceted approach to address the limitations of the existing archival system, consisting of three primary components. An initial Automatic Speech Recognition (ASR) component, which is utilized to convert spoken language into written text. By using ASR, audio data from the meeting recordings can be transcribed into a searchable format for information retrieval. The second component focuses on Information Retrieval (IR) techniques. Its main objective is to locate and extract information sections from these recordings, using a range of search methods, including keywords, semantic context, and metadata. Finally, the integration of a Question and Answering (Q&A) component, which can quickly retrieve information from the transcripts generated by the ASR system, detecting and summarizing valuable data that would otherwise be difficult and time-consuming to find.

The purpose of this paper is to introduce a new meeting archival system to enhance the information retrieval from these meeting recordings. While the research areas of ASR, IR and QA have been extensively researched independently, their combined integration could show significant potential [21], [22]. The overarching goal is to enrich the user experience and improve the accessibility of the content of these recordings.

1.1 Research Question

The main research question is:

To what extent can Automatic Speech Recognition (ASR) and Information Retrieval (IR) be integrated to increase relevancy of search results within Dutch meeting recordings?

1.1.1 Sub questions.

- (1) To what extent can an ASR model be used accurately transcribe existing meeting recordings?
 - How do metrics such as Word Error Rate (WER) and Jaccard Index reflect the performance and accuracy of the ASR model?
- (2) What methods or components are needed in the architecture of the proposed Automatic Speech Recognition (ASR) and Information Retrieval (IR) system?

This paper aims to address these questions by following a structured approach. Firstly, a review of related literature is presented to provide an understanding of the relevant terms and techniques. Subsequently, the methodology employed in this research is introduced, along with the corresponding results. Finally, the paper concludes by summarizing the findings, offering points of discussion, and proposing suggestions for future research.

2 BACKGROUND LITERATURE

This chapter discusses developments in automatic speech recognition (ASR) and information retrieval (IR). It outlines the evolution from early models in ASR to advanced solutions, the shift in IR from keyword-matching to semantic search, and the impact of integrating LLMs into IR systems. The overview provides a contextual basis for this research.

2.1 Automatic Speech Recognition (ASR)

Developing robust and human level ASR models has seen some challenges, as they must contend with various scenarios such as noisy input signals, diverse speakers, and distinctive characteristics of speech [49]. Initial work focused on utilizing Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) to effectively model speech acoustics, despite the inherent challenges posed by speech data variability [48], [20]. Over time, research led to the development of hybrid solutions. These systems incorporate multiple machine learning components that are trained independently, demonstrating their superiority over traditional models, particularly when dealing with limited training data [34] [44].

Simultaneously, there has been a rise in the popularity of end-to-end approaches in parallel with the development of hybrid solutions [53] [11]. These models simplify the ASR process by directly mapping speech signals to transcriptions, representing a significant advancement compared to the module-based approach of hybrid systems [44]. More recently, Transformer models originally designed for machine translation tasks have been adapted for ASR applications. They offer solutions to longstanding challenges in sequential calculations and signal correlations [12] [42] [27]. Research has demonstrated the efficiency of Transformers across various tasks, showcasing their successful application in both hybrid

and end-to-end ASR systems [37].

The performance of ASR models is generally assessed through metrics such as the Word Error Rate (WER) and the Jaccard Index. WER measures the difference between the generated output and the reference transcription, taking into account substitutions, deletions and insertions [15]. Accuracy is quantified by dividing the total number of errors by the number of words in the reference. The Jaccard Index assesses the similarity between sets of sentences in generated and reference transcriptions. It handles output length variations and is calculated by comparing the common words between the sets with the total combined words.

2.2 Information Retrieval (IR)

The field of Information Retrieval (IR) has traditionally relied on term-matching algorithms such as TF-IDF and BM25 to identify documents containing specific query keywords. BM25, an information retrieval technique employed in the ranking of documents according to their relevance to a given search query. It is a probabilistic model based on the bag-of-words concept, succeeding the traditional Term Frequency-Inverse Document Frequency (TF-IDF) approach [41]. However, these approaches have had limitations, especially when dealing with terms that have different meanings in the contexts of the overall query [33].

In response to these challenges, research was directed toward semantic search, understanding the intent and contextual implications of search queries. This has led to the development of techniques such as Latent Semantic Indexing (LSI) and Word Embedding models such as Word2Vec [10]. These models are used to improve the understanding of search queries, increasing the relevance of results and taking into account the nuances of language [28].

Comparative studies have shown that there is no one-size-fits-all solution for search algorithms. For example, while semantic models are effective in understanding context and intent, they can sometimes be outperformed by traditional methods such as BM25 when the searches are basic and depend largely on term frequency [30], [25]. It is therefore important to note that none of these methods can be considered the "ideal" search method. Their performance varies depending on the specific requirements of the search task, the complexity of the searches, the nature of the documents being searched, and even the specific implementation details of the search algorithms. Therefore, selecting the most appropriate search algorithm is often a matter of striking a balance between precision and recall, and optimizing based on the specific needs of the task at hand.

2.2.1 Advancements in IR with LLM's. Information retrieval has been significantly enhanced by large language models (LLMs) such as GPT-4 and BERT [17]. Extensively pre-trained and fine-tuned on large text datasets, these LLMs show skills in language comprehension, language generation, interaction and reasoning [52]. As a result, their functionalities are being integrated into information retrieval systems [31], [50], [40]. One direction is conversational search [38]. This approach presents the search system as a knowledgeable agent that conducts multiple conversations with users.

175 Unlike traditional keyword-based or even semantic searches, this
176 dialogue method produces more intuitive and accurate results, sig-
177 nificantly improving the user experience.

178
179 However, LLM integration is not without it's challenges. One no-
180 table problem is the tendency of LLMs to "hallucinate" or generate
181 factually incorrect or contradictory text [31]. To solve this, research
182 has been proposed which uses an information retrieval systems
183 besides an LLM. This approach enables LLMs to draw relevant
184 content from custom private data resources, thus reducing the risk
185 of hallucinations [40] [50]. Moreover, with access to private data,
186 these retrieval systems alongside LLMs can effectively respond to
187 specialized in-domain queries, surpassing the capabilities of LLMs
188 trained solely on public data [31] [52].

189 3 RELATED WORK

190 Archiving previous plenary debates and meeting recordings, as
191 mentioned in the problem statement, is a common feature of sev-
192 eral systems used by different governmental organizations. Some
193 notable examples of such systems include:

- 194 • The "Debat Gemist" website, which provides access to ple-
195 nary debates and committee meetings of the Dutch Tweede
196 Kamer [2]. It's ASR-based feature provides users with accu-
197 rate transcriptions from the recording available for down-
198 load. However, its IR functionality is limited to searches for
199 specific keywords in the transcript and speaker names.
- 200 • Similarly, the Multimedia Centre of the European Parliament
201 offers a system for viewing recorded parliament sessions
202 [3]. The system also makes transcripts available, but it's IR
203 functionality is confined to only specific video title searches.

204 While commercial ASR systems exist that can overcome these
205 mentioned shortcomings, they usually have limited IR capabilities,
206 mostly limited to generating meeting summaries without search
207 capabilities [6] [9]. The few that do offer search features usually
208 rely solely on keyword searches, which do not always provide the
209 information needed. [4] [1].

210 4 METHODOLOGY

211 This section presents the methodology used to evaluate and imple-
212 ment the proposed system. First, the automatic speech recognition
213 (ASR) models used for transcribing Dutch media content are intro-
214 duced. Subsequently, the data pre-processing steps are described,
215 followed by an overview of the Information Retrieval (IR) methods
216 used. The architecture of the proposed system, is presented. Finally,
217 the evaluation metrics for ASR and IR performance are discussed.

218 4.1 ASR

219 To transcribe Dutch media content, three different ASR systems
220 were considered and evaluated based on prior work [37, 42]. The
221 following ASR models are compared:

- 222 - **Whisper**: An ASR model developed by OpenAI [45].
- 223 - **Wav2Vec2**: ASR Model developed by Facebook's AI research
224 lab [13].

- **Kaldi ASR**: Kaldi NL, which is a Dutch trained ASR web service
developed at Radboud University ¹.

227
228 Whisper is a general-purpose speech recognition model. It is
229 trained on a large dataset of diverse audio and is also a multitasking
230 model that can perform multilingual speech recognition, speech
231 translation, and language identification [43]. Five model sizes are
232 available, which provides varying trade-offs between speed and
233 accuracy. The models range from a 'tiny' version with 39 million
234 parameters to a 'large' version comprising 1550 million parameters,
235 the latter necessitating approximately 10GB of VRAM. The 'large'
236 multilingual model was chosen for Dutch language ASR, which
237 showcased a Word Error Rate (WER) of 10.2% on the multilingual
238 Librispeech dataset, and a WER of 7.1% the Common Voice 9 Dataset
239 [45].

240 Wav2Vec2 is specifically designed to facilitate self-supervised learn-
241 ing from raw audio data. In this particular case, the decision was
242 made to utilize the Dutch-language XLSR Wav2Vec2 model, which
243 has been fine-tuned by Jonas Grosman ². This fine-tuned model
244 is specifically trained for the Dutch language and demonstrates a
245 Word Error Rate (WER) of 15.72% on the Common Voice Dutch
246 Dataset ³.

247 Kaldi is a toolkit that helps build custom automatic speech recogni-
248 tion (ASR) systems. In the context of this article, a web service with
249 a Dutch Kaldi version is used. This system was created by Emre
250 Yilmaz, affiliated with the Centre for Language Studies (CLST) at
251 Radboud University Nijmegen ⁴.

252
253 *4.1.1 Data (pre) processing.* To evaluate and compare the men-
254 tioned ASR models, preprocessing is used to identify and extract
255 speech signal features from the audio data. This pre-processed data
256 is further described in 4.2. The primary goal for preprocessing is
257 to improve speech signal clarity while minimizing noise caused by
258 unvoiced signals. This is necessary for achieving a low WER when
259 comparing ASR models [39]. Accurately identifying the start and
260 end points of voiced signals is crucial for distinguishing between
261 voiced and unvoiced signals. In this regard, the segmentation model:
262 pyannote/segmentation [7] is used. This model allows for the detec-
263 tion of areas where there is vocal activity [16], enabling the audio
264 files to be effectively divided into smaller .wav formatted audio
265 clips. Resulting in individual segment containing voiced signals.

266
267 To determine "who speaks when?" after the previous processing
268 step, speaker diarization is implemented using pyannote/diariza-
269 tion [8]. This module identifies the specific speaker associated with
270 each audio segment. Notably, it can detect overlapping speakers by
271 re-scoring the start and end times of each speaker's speech. As a
272 result, the audio files are further broken down into distinct segments,
273 isolating the voice of each speaker.

¹Kaldi NL (https://github.com/opensource-spraakherkenning-nl/asr_nl)

²Dutch Wav2Vec (<https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-dutch>)

³(<https://paperswithcode.com/sota/speech-recognition-on-common-voice-dutch>)

⁴Kaldi NL (https://github.com/opensource-spraakherkenning-nl/asr_nl)

4.2 Data

For the evaluation of the compared ASR models, the Buren dataset and the baseline plenary debate dataset from the Tweede Kamer are described.

4.2.1 Buren Dataset. This first data-set has been created using recordings from the city council Buren [18]. These publicly available recordings comprise multiple archived meeting sessions from the past few months. To represent the structure of the recordings, segments of up to 30 minutes were chosen from these meetings. The complete data-set encompasses approximately 1.5 hours of labeled transcriptions, and is represented in Table 1.

Table 1: Buren Dataset

Metric	Description	Value
Avg. Speech blocks per Meeting	Mean	112.60
	Median	103
	SD	32.28
Avg. Words per Speech	Mean	86.06
	Median	48.00
	SD	87.69
Total Amount of Speeches	338	
Total number of unique speakers	15	
Total Words	29,088	
Unique Words	7,272	
Percentage of Very Short Speeches	11.01%	

In the Buren dataset, about 11.01% of all speeches consists of Very Short Speeches, corresponding to 5 words or less. These brief speeches often include prompt affirmations, denials, or concise responses that punctuate discussions during meetings.

4.2.2 Tweede kamer. The second dataset is derived from the Debat Gemist website, which enables users to view plenary debates from the Tweede Kamer [24]. The website provides transcripts of the recorded sessions, created by the Tweede Kamer annotator. These labeled transcripts form the basis of the baseline data-set. To develop this baseline data set, a similar 1.5-hour transcription is generated by combining these debate recordings. The result of which is displayed in Table 2.

Table 2: Tweede Kamer Dataset

Metric	Description	Value
Avg. Speech blocks per Meeting	Mean	146.38
	Median	133.90
	SD	42.03
Avg. Words per Speech	Mean	111.88
	Median	62.40
	SD	113.78
Total Amount of Speeches	439	
Total number of unique speakers	20	
Total Words	49,144	
Unique Words	12,286	
Percentage of Very Short Speeches	7.31%	

4.3 Search Framework

To apply information retrieval from the raw output of the ASR models, the development of a search pipeline is researched. Several frameworks have emerged to simplify and streamline this information retrieval process.

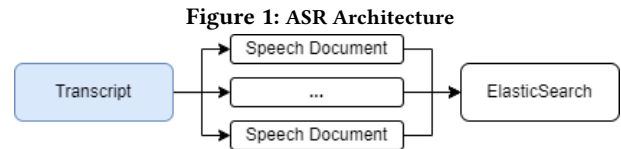
- Haystack by Deepset, is a notable open-source framework capable of implementing end-to-end search pipelines [19].
- Amazon Kendra is search framework developed by Amazon Web Services [26]. Designed to assist in searching across various content repositories using its integrated connectors.

Haystack by Deepset, was chosen as the preferred option for building search pipelines. It offers extensive customization capabilities and maintains compatibility with various libraries of pre-trained models, such as Huggingface⁵. In terms of cost-effectiveness, Haystack proves to be a more viable choice by eliminating the reliance on Kendra’s pay-as-you-go pricing model, resulting in reduced direct costs. For storing the information used in this pipeline, Elasticsearch⁶, a highly scalable open-source storage framework, is utilized.

4.4 Proposed Architecture

The proposed architecture for a new meeting recording archival system consists of three distinct components. The ASR architecture, shown in Figure 1. The information pipeline, shown in Figure 2. And finally, a question-and-answering functionality, described in section 4.4.3.

4.4.1 ASR Architecture. As stated in 4.1.1, the first step of the ASR architecture is pre-processing the meeting recording. The pre-processed data-set, is then transcribed by an ASR model. The resulting transcripts are then segmented into units of 1500 characters, which are regarded as manageable speech blocks. This segmentation process iterates through the transcript, keeping track of speaker information, the start and end time of the speech, and the transcribed text. To maintain the original context and meaning of the speech, speech blocks are divided at sentence boundaries or natural punctuation marks whenever feasible. These segmented speech blocks are converted into Haystack Documents to ensure compatibility with the IR Pipeline, and subsequently stored in Elasticsearch. The architecture of this ASR process is depicted in Figure 1.



4.4.2 IR Pipeline. The second part of the proposed system is the Information Retrieval pipeline. This pipeline integrates two techniques: the BM25 algorithm and embedding methods.

BM25 as stated in 2.2, faces challenges when dealing with complex and context-heavy queries, where the importance lies not

⁵HuggingFace (<https://huggingface.co/>)

⁶Elasticsearch (<https://www.elastic.co/>)

342 in individual search terms but in understanding word order and
343 contextual nature of the query. The models dependency on term
344 frequency can lead to it overlooking documents that contain the
345 required contextual information but do not explicitly match the
346 search terms [35].

347 To improve document retrieval for complex queries, the model
348 "text-embedding-ada-002" from OpenAI [5] is integrated in the IR
349 pipeline besides BM25. This embedding model utilizes word embed-
350 dings to encode transcript documents, preserving their contextual
351 information. The selection of this model is based on it's perfor-
352 mance on the Multilingual Text Embeddings Benchmark (MTEB)⁷
353 and it's proficiency in Dutch language embeddings. For IR, using
354 an embedding model first requires calculating the embedding rep-
355 resentation of the transcript documents. This operation takes place
356 during the ASR process. At this stage, the embeddings are stored
357 alongside the transcription documents in the Elasticsearch index.
358

359 To enable the IR pipeline to either select the BM25 or the embed-
360 ding retriever, it needs to differentiate between short, keyword-
361 based queries and longer, more complex statements/question-based
362 queries. For instance, the system should be able to distinguish be-
363 tween a prompt like 'Rental Housing' and a more complex one
364 such as 'What is the current budget for renovations in 2024'. To
365 accomplish this, a query classification model is employed, namely,
366 'shahrukhx01/bert-mini-finetune-question-detection'⁸. By analyz-
367 ing the structure and attributes of the query, this model can classify
368 the user's query type as either a keyword or a statement/question.
369

370 After the search query is classified, it's processed by either the
371 BM25 retriever or the embedding retriever. This dynamic approach
372 ensures that the retrieval model best suited to the nature and com-
373 plexity of the query is utilized, optimizing information retrieval.
374 The architecture of this system can be seen in Figure 2.
375

376 Additionally, several studies [23], [47], [46], [32], have explored
377 the effectiveness of extending an embedding retriever with an ad-
378 ditional cross-encoder model to improve the relevance of informa-
379 tion retrieval. Unlike embedding retrievers, cross-encoders employ
380 a Transformer network to process both the query and the docu-
381 ment simultaneously. Instead of generating an embedding, a cross-
382 encoder produces a matching score ranging from 0 to 1, indicating
383 the level of similarity between the query and the document [54].
384 A pre-trained cross-encoder was chosen as the preferred model,
385 guided by performance measures on two benchmark datasets: TREC
386 Deep Learning 2019 and MS Marco Passage Re-ranking⁹. From
387 these benchmarks, the multilingual model 'cross-encoder/mmarco-
388 mMiniLMv2-L12-H384-v1'¹⁰ was selected as the cross-encoder for
389 the pipeline.
390

⁷MTEB (<https://huggingface.co/spaces/mteb/leaderboard>)

⁸Query Classifier (<https://huggingface.co/shahrukhx01/bert-mini-finetune-question-detection>)

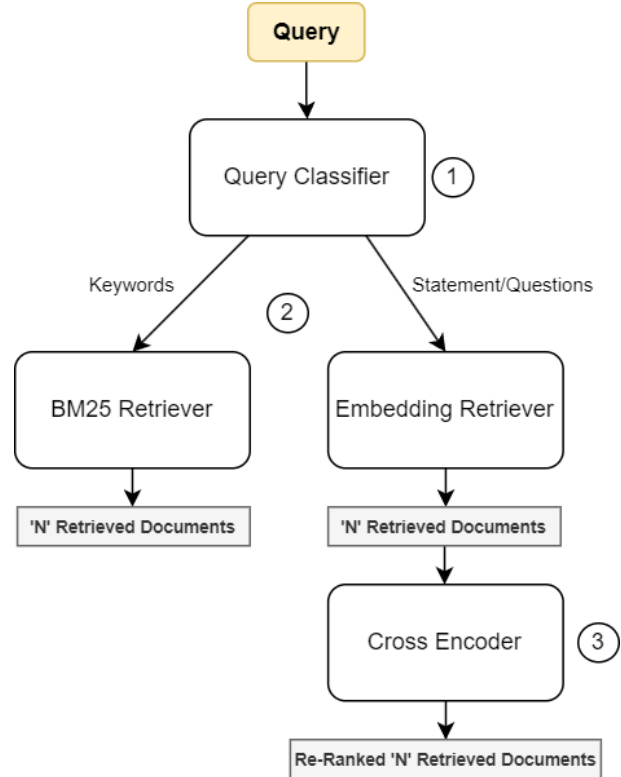
⁹<https://www.sbert.net/docs/pretrained-models/ce-msmarco.html>

¹⁰Cross Encoder (<https://huggingface.co/cross-encoder/mmarco-mMiniLMv2-L12-H384-v1>)

392 However, the high computational demands of cross-encoders, espe-
393 cially in processing each query-document pair, is a limitation. This
394 problem increases as the number of documents increases. To miti-
395 gate the computational limitations of employing the cross-encoder
396 in the IR pipeline, the embedding retriever is initially used to re-
397 trieve a top 'n'¹¹ subset.

398 The cross-encoder compares each query-document pair in this
399 subset, individually assessing the relevance of each document to
400 the query. The process involves comparing the semantic context
401 of the query with the document content to determine a match-
402 ing score. This score is used to re-rank the documents, ensuring
403 that the ones most relevant to the query appear at the top of the
404 search results. This method of applying the cross-encoder is used
405 for improved relevance of the search results but also to provide a
406 more nuanced and context-aware ranking compared to the initial
407 embedding retriever. The role of the cross-encoder is highlighted
408 as step 3, as indicated in Figure 2.
409

Figure 2: IR Pipeline



410 4.4.3 **Question and Answering.** Lastly, the system integrates
411 a question-answering (Q&A) feature to increase the accessibility
412 of meeting recordings. This feature uses a Large Language Model
413 (LLM) to generate responses to user queries. As mentioned in 2.2.1,
414 the potential issue of 'hallucinations' from LLMs can be mitigated
415 by integrating them with an information retrieval system. Because

¹¹'n' is a predetermined value used to reduce the computational cost of each model. In the system, BM25 uses 'n' = 15 and Embedding uses 'n' = 10.

of this, during Q&A, the Information Retrieval (IR) pipeline from Figure 2 is used.

The IR pipeline serves as a means to verify and validate the LLM’s responses. It acts as a reference layer and provides a contextual basis for the responses generated by the LLM. By extracting relevant content from existing documents, it acts as a safeguard against possible "hallucinations" of the LLM and ensures that the responses generated are factual and relevant. In addition, it can provide the LLM’s responses with links to resources in these documents, thus providing a more comprehensive response to user questions.

Several options were evaluated in the search for a language model (LLM). After testing and evaluating their performance in benchmark tests [56], OpenAI’s GPT-4 emerged as the most suitable choice to serve as an LLM for the conversational retrieval.

5 RESULTS

5.1 ASR Evaluation

This section presents a comparison between the chosen ASR models, namely Whisper, a fine-tuned version of Wav2Vec, and a Dutch adapted version of Kaldi ASR. The section begins with an overview comparison which focuses on evaluating the word error rate (WER) and Jaccard Index for each model. It’s noteworthy that a lower WER indicates better performance, while a higher Jaccard Index is preferable. Performance results with the Buren dataset are presented in Table 3, while Table 4 illustrates performance with the baseline dataset derived from the Tweede Kamer recordings.

Table 3: Performance ASR systems on Buren Dataset | n = 120

ASR System	WER	Jaccard Index
Whisper	13.8%	68.2%
Wav2Vec2	18.4%	61.6%
Kaldi ASR	23.2%	56.8%

Table 4: Performance ASR systems on baseline Tweede Kamer dataset | n = 140

ASR System	WER	Jaccard Index
Whisper	11.9%	70.1%
Wav2Vec2	16.5%	63.5%
Kaldi ASR	21.3%	57.7%

5.1.1 Kaldi NL. According to the results, Kaldi NL demonstrated suboptimal performance on both datasets. It’s best performance was seen on the Buren dataset, where it achieved a WER of 23.2% and a Jaccard Index of 56.8%. However, on the baseline dataset derived from Tweede Kamer recordings, the WER decreased slightly to 21.3% and the Jaccard Index increased marginally to 57.7%.

5.1.2 XLSR Wav2Vec2 Fine-tuned Dutch. The XLSR Wav2Vec2 model, which has been fine-tuned for Dutch, demonstrated an improvement over Kaldi NL but didn’t outperform Whisper. On the Buren

dataset, Wav2Vec2 achieved a WER of 18.4% and a Jaccard Index of 61.6%. It’s performance on the Tweede Kamer dataset was slightly better, with a WER of 16.5% and a Jaccard Index of 63.5%.

5.1.3 Whisper Large. Among the three models, Whisper emerged as the best performer. It delivered predictions with the lowest WERs and highest Jaccard Index scores across both datasets. On the Buren dataset, Whisper achieved a WER of 13.8% and a Jaccard Index of 68.2%. The performance improved further on the Tweede Kamer dataset, where it achieved a WER of 11.9% and a Jaccard Index of 70.1%.

5.2 Information Retrieval Quality

In assessing the information search pipeline, attention was paid to the quality of the retrieved data. A user study was conducted to measure the perceived relevance of the retrieved documents. The metric Normalized Discounted Cumulative Gain (NDCG) was used to quantify the evaluation of quality. This metric takes into account not only the relevance of the retrieved documents, but also the ranking of the documents in the results list. As a result, highly relevant documents that are earlier in the list contribute more to the overall NDCG score than documents that are lower in the returned list. The NDCG is computed in two stages. First, the Discounted Cumulative Gain (DCG) at a position p is calculated as follows:

$$DCG_p = \text{relevance}_1 + \sum_{i=2}^p \frac{\text{relevance}_i}{\log_2(i+1)} \quad (1)$$

In this equation, relevance_i represents the relevance score of the document at position i in the results list. The relevance scores reflect how well the document matches the user’s search query or information needs. The second stage involves calculating the NDCG at position p using the formula:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (2)$$

Here, $IDCG_p$ represents the DCG score of the perfect ranking of documents. That is, it denotes the ideal ordering of documents based on their relevance scores. The purpose of using the perfect ranking as a reference is to compare the actual ranking with the best possible ranking. The results of the NDCG evaluation can be seen in Figure 3 and Figure 4

To maintain standardization during data collection, each user was assigned a set of five prompts to test the the different retrieval methods of the system. The list of prompts used in this test can be found in Appendix A. The first two prompts were keyword-based, while the remaining three prompts consisted of longer questions. For each prompt, the models returned the top 5 documents. Subsequently, users were asked to rate the relevance of these retrieved documents on a scale of 0 to 5, with 5 indicating the highest level of relevance to the input prompt, according to the user’s judgment. The purpose of using the same prompts for all models was to minimize potential variations in responses that could arise from using different prompts.

5.2.1 BM25 Results. When it comes to keyword prompts, the BM25 model demonstrated effective performance notably with an NDCG

Figure 3: User test results | NDCG

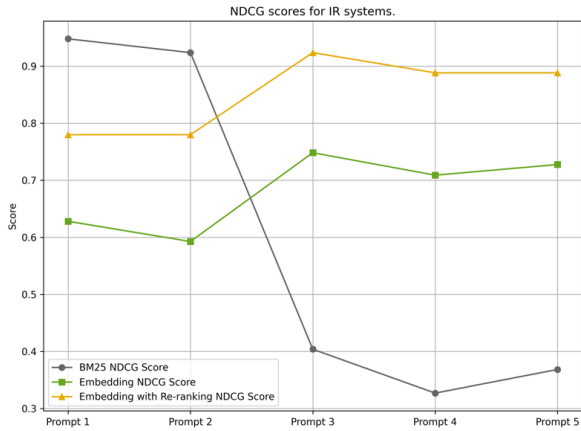
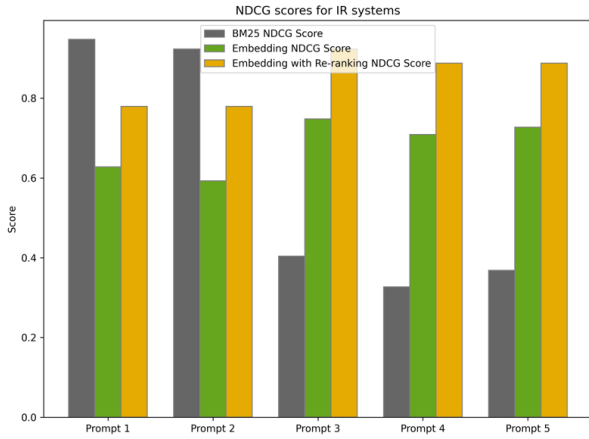


Figure 4: User test results | NDCG



score on the first keyword prompt of 96.43%. However, its effectiveness decreased notably when dealing with longer more complex prompts. This can be attributed to the design of BM25, which relies on term frequency to determine the relevance of documents. This design may not be as capable of understanding the nuanced semantic understanding needed to interpret the intent behind a query.

5.2.2 *Embedding Retriever Performance.* The Embedding Retriever model showcased improved competence in retrieving relevant documents when presented with question prompts. Achieving a NDCG score of 75.23%. This progress comes from a greater ability to understand and interpret the intent of complex searches, where context is more important than term frequency. As a result, more relevant documents for these types of queries are returned.

5.2.3 *Combined Approach: Embedding and Ranking.* Integrating the Embedding Retriever with the Cross Encoder Ranker resulted in the most consistent performance for both keyword queries and more complex question queries. The Cross Encoder Ranker improves the initial results of the Embedding Retriever by re-ranking

the top-n documents. Resulting in a highest NDCG score of 92.13% on the question queries.

5.3 Information Retrieval Speed

To evaluate the trade-off between quality and speed of different retrieval methods, this section examines the retrieval speed of BM25, Embedding Retriever and the embedding retriever + re-ranker. In these results, the same 5 prompts from the quality evaluation were used.

Table 5: Exec. Time (sec) for Each Stage in BM25 retrieval

Stage	P1	P2	P3	P4	P5
Initialization BM25	0.0	0.0	0.0	0.0	0.0
Retrieval	0.0070	0.0071	0.0075	0.0076	0.0077
Full/Query Execution	0.0090	0.0091	0.0094	0.0095	0.0096

5.3.1 *Bm25 Retrieval Speed.* Table 5 reveals the performance of the BM25 method. It showed an instant initialization and a fast document retrieval speed, averaging 0.0072 seconds across the five prompts (P1 to P5). The full execution, including the query, averaged at 0.0092 seconds.

Table 6: Exec. Time (sec) for Each Stage in Embedding retrieval

Stage	P1	P2	P3	P4	P5
Init. Embedding Retriever	0.270	0.271	0.273	0.275	0.275
Retrieval of Embedding Docs	0.750	0.752	0.756	0.759	0.760
Full/Query Exec.	1.020	1.023	1.029	1.034	1.035

5.3.2 *Embedding Retrieval Speed.* Contrasting with BM25, the Embedding Retrieval method exhibits a more complex process, as shown in Table 6. The initialization averages at 0.2728 seconds, requiring a notably longer duration, being about 30 times slower in the initialization phase than the BM25 method. Document retrieval in this approach is slower, averaging 0.7554 seconds. Furthermore, the full execution of this retrieval method, including the query execution, takes significantly longer than BM25, clocking at about 110 times the speed of BM25. With an averaging of 1.0282 seconds across all phases.

Table 7: Exec. Time (sec) for Each Stage in Embedding retrieval with Re-ranking

Stage	P1	P2	P3	P4	P5
Init. Embedding Retriever	0.270	0.271	0.273	0.275	0.275
Retrieval of Embedding Docs	0.750	0.752	0.756	0.759	0.760
Document Ranking	1.610	1.613	1.618	1.621	1.623
Full/Query Exec.	2.380	2.383	2.390	2.394	2.395

5.3.3 *Embedding + Re-Ranking Retrieval Speed.* Embedding Retrieval with Re-ranking, shows significant execution times being 260 times slower than the BM25, as presented in Table 7. The initialization phase remains constant, similar to the standard Embedding

Retrieval method. Document retrieval also aligns with its counterpart, averaging at 0.7554 seconds. The introduction of the document ranking phase is the primary contributor to the overall increase in execution times, taking an average of 1.617 seconds across all phases. Consequently, the overall full execution time, including the query, averages at 2.3884 seconds, significantly longer than both the BM25 and standard Embedding Retrieval methods.

The comparison between the three models (BM25, Embedding Retrieval, and Embedding Retrieval with Re-ranking) is visualized in Table 8. The table presents the average execution time for each stage in the three models.

Table 8: Mean Exec. Time (sec) for Each Stage

Stage	BM25	Embedding	Embedding + re-ranking
Initialization	0.0	0.2728	0.2728
Retrieval	0.00738	0.7554	0.7554
Document Ranking	N/A	N/A	1.617
Full/Query Execution	0.00932	1.0282	2.3884

6 DISCUSSION

6.1 ASR Models

An evaluation was carried out on three ASR systems: Whisper, Dutch fine-tuned Wav2Vec2, and Kaldi NL. For each, testing was conducted on both the Buren ($n = 120$) and Tweede Kamer ($n = 140$) datasets. Whisper demonstrated superior performance compared to the other models on both datasets, with a Word Error Rate (WER) of 13.8% and 11.9%, respectively. The mean WER for Whisper across both datasets was 12.85%, with a standard deviation of ± 0.95 on the Buren data-set and ± 1.19 on the baseline dataset, indicating a low variability in performance across the datasets. These results align with existing research [45], [55] that also demonstrated superior performance of Whisper in terms of WER on English as well as multilingual datasets.

However, a limitation was exhibited with all three models when handling specific segments of the recordings. In particular, each system demonstrated difficulty in decoding sections of speeches marked by overlapping voices and unclear audio signals. A potential strategy to address this issue could be to test the systems under a range of different audio conditions to determine the extent of these limitations. Lastly, as noted by [14], implementing further in-domain fine-tuning could potentially improve the WER performance across all models.

6.2 IR Systems

NDCG evaluations revealed BM25's proficiency with short, keyword-based prompts, achieving an average user score of 96%. For more complex statement/question prompts, incorporating a cross-encoder for re-scoring after using the Embedding Retriever raised this average relevance score from 75% to 92%. In terms of speed, BM25 excelled, averaging 0.0092 seconds for document retrieval, while the more complex Embedding Retriever averaged 1.0282 seconds. Including re-ranking prolonged the process to 2.3884 seconds. These results emphasize the quality-speed trade-off. While BM25 may

provide speed benefits, the relevance for longer prompts degrade, highlighting the necessity of this balanced approach.

It should be noted that with the current corpus size for a transcribed meeting (about 120 documents), the retrieval speed is relatively quick. However, scaling to a larger document set may expose limitations, particularly affecting the embedding retriever. This is due to the inefficiency of real-time embedding matching in a larger corpus to find documents similar to a given query. The use of Hierarchical Navigable Small World graphs (HNSW) could potentially speed up this retrieval process [36]. This method would allow efficient approximation of nearest neighbors for embeddings in high-dimensional spaces.

6.3 QA

After conducting an initial user testing to assess the effectiveness of the IR system, participants were also asked to test the integrated Question and Answering feature. This additional evaluation was done to gather qualitative feedback.

Users responded positively to this feature, as it allowed for a more natural and conversational approach to information retrieval. In addition, the system's ability to summarize key points from discussions proved useful to users. However, while evaluating the QA system, it was noted that the quality of the answers generated depended heavily on the construction of the query or prompt. In fact, when users asked specific and detailed questions, the system provided more accurate and valuable answers. This observation is in line with studies by [57], [51], and [29], which explored the impact of prompt engineering on various generative Question Answering systems' performance. In these studies, prompt engineering was found to play a significant role in enhancing the system's output comprehensibility, relevance, and overall quality.

7 CONCLUSION

This research addresses the limitations in the current archival system for meeting recordings used by city councils in the Netherlands. The current system's reliance on metadata-based searches often hinders users from efficiently finding and extracting the information they seek. To overcome these limitations, a multi-faceted approach has been proposed, consisting of Automatic Speech Recognition (ASR), Information Retrieval (IR), and Question and Answering (QA) components.

This research aims to answer the main research question: "To what extent can Automatic Speech Recognition (ASR) and Information Retrieval (IR) be integrated to increase the relevancy of search results within Dutch meeting recordings?" A comparative analysis of three ASR systems - Whisper, Wav2Vec2, and Kaldi NL - showed that OpenAI's Whisper model outperformed the other two. The Whisper model recorded the lowest Word Error Rate (WER), with 13.8% on the test set and 11.9% on the baseline dataset.

To further improve the system, integration of information retrieval was proposed, using a hybrid method consisting of three components. BM25 was used for short, keyword-driven queries, an

646 embedding retriever was applied for more complex and contex- 693
647 tually dependent queries. Lastly, a combination of an embedding 694
648 retriever with a cross-encoder was introduced to refine the rele- 695
649 vance of returned information by re-scoring search results. The 696
697 effectiveness of the proposed method, particularly its impact on
698 the relevance of search results, was quantified using Normalized
699 Discounted Cumulative Gain (NDCG). The results showed that for
700 short, keyword-based prompts, users found an average relevance
701 score of 96%. For more complex, context heavy queries, incorporat-
702 ing a cross-encoder for re-scoring results improved average rele-
703 vance score, from 75% to 92%. However, the use of a cross-encoder
704 for improved relevance came at the cost of increased execution time,
705 highlighting the speed/quality trade-off the IR system. Additionally,
706 in an effort to improve the overall user experience and broaden
707 search, a question and answer (QA) system was introduced. This
708 component allowed users to receive responses generated by Ope-
709 nAI’s GPT-4, which directly referenced the content of the meeting
710 recordings. This feature received positive feedback from users, who
711 appreciated the ability to facilitate a more natural and interactive
712 approach to searching for information.
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734

667 However, these results should be considered in light of certain
668 limitations. Despite the superiority of the Whisper model, all three
669 ASR models revealed challenges in decoding overlapping and un-
670 clear speech segments. Moreover, although the hybrid IR approach
671 enhanced search relevancy, it revealed a potential drawback in
672 terms of retrieval speed when applied to larger document collec-
673 tions. Additionally, it was observed that the quality of answers
674 generated by the QA component heavily depended on how the user
675 queries were formulated.
676

677 The research demonstrates the potential of integrating ASR and IR
678 technologies to improve search relevance and streamline naviga-
679 tion within meeting recordings. These findings have implications
680 beyond Dutch city council meetings. Offering new opportunities for
681 leveraging speech recognition and information retrieval techniques
682 in various fields. With the goal to enhance the accessibility and
683 usability of information retrieval from audio data.
684

685 Opportunities for future work could address the scalability lim-
686 itations identified in this study. Key areas of interest include the
687 integration of scalable retrieval techniques, especially in the context
688 of processing complex and contextually important queries. Addi-
689 tional, in domain fine-tuning could also improve the performance
690 of the ASR model. Lastly, a better understanding of the prompts used
691 with prompt engineering could improve the generated responses
692 of the QA system.

REFERENCES

- [1] 2023. *Converteer audio en video automatisch naar tekst* | Sonix. <https://sonix.ai/nl>
- [2] 2023. *Debat Gemist | Via Debat Gemist kunt u plenaire debatten en commissievergaderingen van de Tweede Kamer terugkijken*. <https://debatgemist.tweedekamer.nl/>
- [3] 2023. *European Parliament Multimedia Centre*. <https://multimedia.europarl.europa.eu/nl>
- [4] 2023. *MeetGeek | Record, Transcribe Share Meeting Notes*. <https://meetgeek.ai/>
- [5] OpenAI 2023. *New and improved embedding model*. OpenAI. <https://openai.com/blog/new-and-improved-embedding-model>
- [6] 2023. *Otter.ai - Voice Meeting Notes Real-time Transcription*. <https://otter.ai/>
- [7] pyannote 2023. *pyannote/segmentation · Hugging Face*. pyannote. <https://huggingface.co/pyannote/segmentation>
- [8] pyannote 2023. *pyannote/speaker-diarization · Hugging Face*. pyannote. <https://huggingface.co/pyannote/speaker-diarization>
- [9] Rev.com 2023. *Transcribe Speech to Text | Rev. Rev.com*. <https://www.rev.com/>
- [10] Edgar Altszyler, Sidarta Ribeiro, Mariano Sigman, and Diego Fernández Slezak. 2017. The interpretation of dream meaning: Resolving ambiguity using Latent Semantic Analysis in a small corpus of text. *Consciousness and Cognition* 56 (11 2017), 178–187. <https://doi.org/10.1016/j.concog.2017.09.004>
- [11] Keyu An, Hongyu Xiang, and Zhijian Ou. 2020. CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency. *arXiv preprint arXiv:2005.13326* (2020).
- [12] Junyi Ao, Ziqiang Zhang, Long Zhou, Shujie Liu, Haizhou Li, Tom Ko, Lirong Dai, Jinyu Li, Yao Qian, and Furu Wei. 2022. Pre-training transformer decoder for end-to-end asr model with unpaired speech data. *arXiv preprint arXiv:2203.17113* (2022).
- [13] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. (2020). arXiv:2006.11477 [cs.CL]
- [14] Jayadev Billa. 2021. Improving low-resource ASR performance with untranscribed out-of-domain data. (2021). arXiv:2106.01227 [cs.CL]
- [15] Gamal Bohouta and Veton Këpuska. 2017. Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *Int. Journal of Engineering Research and Application* 2248-9622 (03 2017), 20–24. <https://doi.org/10.9790/9622-0703022024>
- [16] Hervé Bredin. 2021. End-to-end speaker segmentation for overlap-aware resegmentation. (4 2021). <https://arxiv.org/abs/2104.04045>
- [17] Tom B. Brown. 2020. Language Models are Few-Shot Learners. (5 2020). <https://arxiv.org/abs/2005.14165>
- [18] Royalcast Buren. 2023. *Vergaderingen Buren*. <https://channel.royalcast.com/buren/#!/archived>
- [19] Haystack by Deepset. 2023. *Haystack | Haystack*. <https://haystack.deepset.ai/>
- [20] Akshay Madhav Deshmukh. 2020. Comparison of hidden markov model and recurrent neural network in automatic speech recognition. *European Journal of Engineering and Technology Research* 5, 8 (2020), 958–965.
- [21] Souraya Ezzat, Neamat El Gayar, and Moustafa M. Ghanem. 2010. Investigating analysis of speech content through text classification. (2010), 105–110. <https://doi.org/10.1109/SOCPAR.2010.5686000>
- [22] Xuandi Fu, Feng-Ju Chang, Martin Radfar, Kai Wei, Jing Liu, Grant P. Strimel, and Kanthashree Mysore Sathyendra. 2022. Multi-Task RNN-T with Semantic Decoder for Streamable Spoken Language Understanding. (2022), 7507–7511. <https://doi.org/10.1109/ICASSP43922.2022.9746104>
- [23] Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulić, and Iryna Gurevych. 2022. Retrieve Fast, Rerank Smart: Cooperative and Joint Approaches for Improved Cross-Modal Retrieval. *Transactions of the Association for Computational Linguistics* 10 (05 2022), 503–521. https://doi.org/10.1162/tacl_a_00473
- [24] Tweede Kamer Debat Gemist. 2023. *Debat Gemist | Via Debat Gemist kunt u plenaire debatten en commissievergaderingen van de Tweede Kamer terugkijken*. <https://debatgemist.tweedekamer.nl/>
- [25] Vladimir Karpukhin, Barlas Ögüz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).
- [26] Amazon Kendra. 2023. *Enterprise Search Engine - Amazon Kendra - AWS*. <https://aws.amazon.com/kendra/>
- [27] Santosh Kesiraju, Marek Sarvas, Tomas Pavlicek, Cecile Macaire, and Alejandro Ciuba. 2023. Strategies for improving low resource speech to text translation relying on pre-trained ASR models. *arXiv preprint arXiv:2306.00208* (2023).
- [28] et al Khadija. 2012. Evaluating Semantic Search Systems to Identify Future Directions of Research. (2012), 148–162. https://doi.org/10.1007/978-3-662-46641-4_11
- [29] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. (2021). arXiv:2101.00190 [cs.CL]
- [30] Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270* (2020).

- 769 [31] Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen. 2023. RETA-LLM: A Retrieval-Augmented Large Language Model Toolkit. 770 (2023). arXiv:2306.05212 [cs.IR] 847
- 772 [32] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. 773 2022. ERNIE-Search: Bridging Cross-Encoder with Dual-Encoder via Self On-the-fly Distillation for Dense Passage Retrieval. (2022). arXiv:2205.09153 [cs.CL] 848
- 774 [33] Man Luo, Arindam Mitra, Tejas Gokhale, and Chitta Baral. 2022. Improving Biomedical Information Retrieval with Neural Retrievers. (2022). 849
- 778 [34] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. RWTH ASR Systems for LibriSpeech: Hybrid vs Attention-w/o Data Augmentation. *arXiv preprint arXiv:1905.03072* (2019). 850
- 782 [35] Yuanhua Lv and ChengXiang Zhai. 2011. When documents are very long, BM25 fails! (07 2011), 1103–1104. <https://doi.org/10.1145/2009916.2010070> 851
- 784 [36] Yu. A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. (2018). arXiv:1603.09320 [cs.DS]
- 787 [37] Orken Mamyrbayev, Oralbekova Dina, Alimhan Keylan, Turdalykyzy Tolganay, and Othman Mohamed. 2022. A study of transformer-based end-to-end speech recognition system for Kazakh language. *Scientific Reports* 12, 1 (5 2022). <https://doi.org/10.1038/s41598-022-12260-y>
- 791 [38] Kelong Mao, Zhicheng Dou, Haonan Chen, Fengran Mo, and Hongjin Qian. 2023. Large Language Models Know Your Contextual Search Intent: A Prompting Framework for Conversational Search. (2023). arXiv:2303.06573 [cs.IR]
- 793 [39] Tobias Menne, Ilya Sklyar, Ralf Schlüter, and Hermann Ney. 2019. Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech. *arXiv preprint arXiv:1905.03500* (2019).
- 797 [40] Reichihiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. WebGPT: Browser-assisted question-answering with human feedback. (2022). arXiv:2112.09332 [cs.CL]
- 800 [41] Jianmo Ni. 2021. Large Dual Encoders Are Generalizable Retrievers. (12 2021). <https://arxiv.org/abs/2112.07899>
- 802 [42] Raphael Olivier and Bhiksha Raj. 2022. Recent improvements of ASR models in the face of adversarial attacks. *arXiv preprint arXiv:2203.16536* (2022).
- 804 [43] Openai. 2023. *openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision*. <https://github.com/openai/whisper>
- 806 [44] Juan M Perero-Codocero, Fernando M Espinoza-Cuadros, and Luis A Hernández-Gómez. 2022. A comparison of hybrid and end-to-end ASR systems for the IberSpeech-RTVE 2020 speech-to-text transcription challenge. *Applied Sciences* 12, 2 (2022), 903.
- 808 [45] Alec Radford. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. (12 2022). <https://arxiv.org/abs/2212.04356>
- 810 [46] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. (2019). arXiv:1908.10084 [cs.CL]
- 812 [47] Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. In Defense of Cross-Encoders for Zero-Shot Retrieval. (2022). arXiv:2212.06121 [cs.IR]
- 814 [48] Fei Sha and Lawrence Saul. 2006. Large margin hidden Markov models for automatic speech recognition. *Advances in neural information processing systems* 19 (2006).
- 816 [49] Davinder Sharma and Jamin Atkins. 2014. Automatic speech recognition systems: Challenges and recent implementation trends. *International Journal of Signal and Imaging Systems Engineering* 7 (12 2014), 220–234. <https://doi.org/10.1504/IJSISE.2014.066600>
- 818 [50] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. REPLUG: Retrieval-Augmented Black-Box Language Models. (2023). arXiv:2301.12652 [cs.CL]
- 820 [51] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. (2020). arXiv:2010.15980 [cs.CL]
- 822 [52] Weiwei Sun. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. (4 2023). <https://arxiv.org/abs/2304.09542>
- 824 [53] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. 2019. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460* (2019).
- 826 [54] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. (2021). arXiv:2010.08240 [cs.CL]
- 828 [55] Juan Camilo Vásquez-Correa and Aitor Alvarez. 2023. Novel Speech Recognition Systems Applied to Forensics within Child Exploitation: Wav2vec2.0 vs. Whisper. *Sensors* 23, 4 (2 2023), 1843. <https://doi.org/10.3390/s23041843>
- 830 [56] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. (2023). arXiv:2303.18223 [cs.CL]
- 832 [57] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitus, Harris Chan, and Jimmy Ba. 2023. Large Language Models Are Human-Level Prompt Engineers. (2023). arXiv:2211.01910 [cs.LG]

852 A APPENDIX

853 A.1 NDCG Test Prompts

854 In the section is a list given of the prompts used in the NDCG user
855 test to test the quality of the relevant returned documents by the
856 system:

- 857 – **(Prompt 1 - Keyword):** ZBB.
- 858 – **(Prompt 2 - Keyword):** Hybride Gebouw.
- 859 – **(Prompt 3 - Statement):** De voorstellen van het Zero Based
860 Budgeting (ZBB) project.
- 861 – **(Prompt 4 - Question):** Wat zijn de voorgestelde beleid-
862 skaders voor de plaatsing van kleine windmolens?
- 863 – **(Prompt 5 - Question):** Wat zijn de lokale ambities en doel-
864 stellingen van de gemeente Buren met betrekking tot de
865 circulaire economie?

866 A.2 QA Example

867 This section provides some examples of the responses produced by
868 the question and answer system during the qualitative user test:
869

Input: Wat zijn de voorstellen van het Zero Based Budgeting (ZBB) project?

Response: De voorstellen van het Zero Based Budgeting (ZBB) project zijn gericht op het ombouwen en bezuinigen op verschillende posten binnen de gemeenten. Het doel is om terug te gaan naar de wettelijke taken en benchmarking speelt hierbij een rol. De berekeningen zijn destijds gemaakt en vanaf 2023 moeten er bezuinigingen worden doorgevoerd op de exploitatielasten van de huisvesting. Het gemeentehuis, dat jaarlijks 200.000 euro kost, moet vanaf 2024 een bedrag van 250.000 euro bezuinigen. Het ZBB project heeft dus als aanleiding gediend om te kijken naar voorstellen om deze bezuinigingen te realiseren.

870 **Input:** Wat zijn de voorgestelde beleidskaders voor de plaatsing van kleine windmolens?

Response: De voorgestelde beleidskaders voor de plaatsing van kleine windmolens zijn dat initiatiefnemers een positief advies van de Commissie Ruimtelijke Kwaliteit moeten verkrijgen en hun omgeving moeten betrekken bij de plannen en hiervan verslag moeten doen bij de indiening van hun verzoek. Verder wordt voorgesteld dat de tiphoogte van de windmolens 25 meter moet zijn en dat er een maximum van 20 kleine windmolens geplaatst mogen worden. Er wordt geen maximale hoogte voorgesteld door de provincie, maar de gemeente moet wel strikte voorwaarden hanteren over waar de windmolens geplaatst mogen worden, bijvoorbeeld niet in weidevogelgebieden.