Spoken Question Answering on Municipal Council Meetings

Pepijn van Wijk and Maarten Marx[0000−0003−3255−3729]

IRLab, Informatics Institute, University of Amsterdam
maartenmarx@uva.nl pepijn.van.wijk@student.uva.nl

Abstract. We demo an IR system for municipal council meetings only available in video format. The system uses Whisper to turn speech into text. Three IR systems were compared in a user study: BM25/TF-IDF, KNN ranking on MPNet embeddings, and a chatbot prompted with the query and the top k highly ranked passages. Users solved complex information need slightly faster using the chatbot, but did not find the correct answer in 23%, whereas these were always found with the BM25 and KNN rankers. The benefits of the IR system compared to no system were huge: problem solving time dropped from around 2 minutes to half a minute and the miss rate from 69% to 0%.

Keywords: Spoken Question Answering, Council Meeting Notes

1 Introduction

We present an IR solution to the following type of information need:

> What was the standpoint of political party X on topic Y during yesterday evening's municipal council meeting?

As the notes of such meetings are only available as a video recording, this information need belongs to the IR field of Open Domain Spoken Question Answering [8]. A desirable interface would be a chatbot which generates a short summary together with one or more entry points in the video to which the answer can be attributed [10]. The main challenge already mentioned in Allen 2002 [1] is present in this scenario: both the data and the information need frequently contain named entities and out of vocabulary words [12]. Even with reasonable Automatic Speech Recognition (ASR) quality, this is problematic for the often used QA pipeline consisting of a fast short passage retrieval using a traditional or learned sparse index, followed by an expensive top N passage reranker, followed by an answer generation step by an LLM prompted with the query and a few top ranked passages [7].

We were curious whether the reported very high ASR quality of Whisper [11], even for a relatively small language as Dutch, could make this sketched QA pipeline based on passage retrieval from ASRed text feasible. We built a complete system including a well working interface in order to evaluate every module in the pipeline. In the user evaluation, we compared the problem solving speed of various search systems ranging

from BM25 to a chatbot. This involved both the time to find, and to verify the correct answer.

Our main findings show that Whisper, even on the Dutch language, generates fairly accurate transcriptions with a word error rate of 10%. The speaker segmentation model used had a diarization error rate of 7%. Both these models are easily run on consumer grade GPUs, with a real-time processing factor of around 30.

In the user evaluation, we found that the search engine significantly reduced average task completion times, decreasing from 113 to 30 seconds. The chatbot achieved an average response time of just 14 seconds. However, in almost a quarter of the tests, the user did not find the right answer with the chatbot, whereas With BM25 and KNN vector search, the success rate was 100%.

A working prototype is available at `https://videotulen.wooverheid.nl/#/gemeente/hoekschewaard/vergaderingen/2023/1068571` (the chatbot is switched off because of the high costs of the GPU, but will be working in the demo), and a video demonstrating the system at `https://surfdrive.surf.nl/files/index.php/s/5xufxXWiFJ99c1M`. All code and manually annotated test-data is available at `https://github.com/deboradum/videotulen`.

2 Description of the system

The input data consists of over 3000 hours of automatic video recordings of municipal council meetings in six different municipalities. All meeting archives were obtained from municipality-managed websites specifically designated for hosting council meeting recordings, in compliance with Dutch legal requirements. In our corpus, these had a mean length of 2 hours, 10 minutes (N=1.020, $\sigma = 69$). The audio quality is high, as each speaker has a separate microphone. The video automatically zooms in on the person that speaks and this works rather well. The videos are divided into "chapters" corresponding to the agendapoints of the meeting ($\mu = 11.5,\ \sigma = 8.6$). The agendapoints were available as text consisting of just a few (often very general) terms.

2.1 Automatic speech and speaker recognition

We use OpenAI's Whisper ASR [11] to generate transcriptions of the input videos. The speaker segmentation model pyannote.audio [3] extracts which unique speakers are speaking at what moment in the video. The data obtained from both models is then combined to define speaker turns. These *speeches* will be the indexed units in the search system. The voice profiles of all unique speakers are embedded using pyannote.audio as well. Profiles are linked to speaker names in a one-time manual process. These named voice profiles allow users to filter their search to specific speakers.

2.2 Search engine

The units of retrieval, the speeches, are indexed using TF-IDF weighting and are also stored as dense vectors after embedding them using multilingual MPNet [14]. We used Weaviate [5] as both a vector database and a search engine. The search engine allows for three distinct search methods: keyword search with BM25 ranking; top k nearest

neighbors measured by the cosine similarity between the embedding of the query and those of the documents; and a tunable hybrid between these two. The hybrid search score is given by a weighted linear combination of the ranks of both search methods using the reciprocal rank fusion formula from [4].

2.3 Chatbot

Given a natural language question Q, the chatbot is prompted with Q and the top 3 highest ranked speeches given Q. We experimented with two LLMs: a locally running instance of Meta's Llama3 and OpenAI's GPT4o. In order to prevent hallucinations, the LLMs are specifically instructed to not answer questions if the answer is not in the provided speeches, unless the user asks for a question explaining a concept mentioned in the video. The following prompt (in Dutch) was used:

> Je bent een behulpzame assistent die vragen over gemeente vergaderingen beantwoord. Je krijgt bij elke vraag context meegestuurd waar je je antwoord op moet baseren. Als het antwoord niet in de context staat, laat dit weten en verzin geen antwoorden. Generieke vragen kan je beantwoorden met je eigen kennis.

Which translates to:

> You are a helpful assistant that answers questions about municipal meetings. You will receive context with each question, and your answers must be based on this context. If the answer is not in the context, indicate this and do not fabricate responses. You may answer generic questions with your own knowledge.

2.4 Interface

Figure 1 shows the interface; area (1) in the figure contains a clickable scrollbox containing the hierarchical "table of contents" consisting of agenda-points divided into speeches which are displayed on mouse hover. All available meeting topics can also be found in area (3). Area (2) and (4) respectively show the search and chatbot areas. In the search area, one can filter search results for specific topics and speakers. Users can also choose between the different search methods.

3 Evaluation

The entire analysis preprocessing phase is easily ran on consumer-grade GPUs. The Whisper large v3 model requires roughly 10GB of VRAM, with a real-time factor 30-40x, depending on the GPU. pyannote.audio only requires about 8GB of VRAM and also has has a real-time factor of about 30-40x.

To test the performance of Whisper on the dataset, we manually transcribed 40 minutes (3.349 words) of random audio chunks across seven different meetings from five years and three municipalities. After standard pre-processing of the data, we recorded an average (over the 7 meetings) word error rate (WER)[1] of about 10%, with a median of 9%.

[1] Recordings of Zoom meetings during the Covid-era had a significantly worse WER. Without these, average WER is 8%.

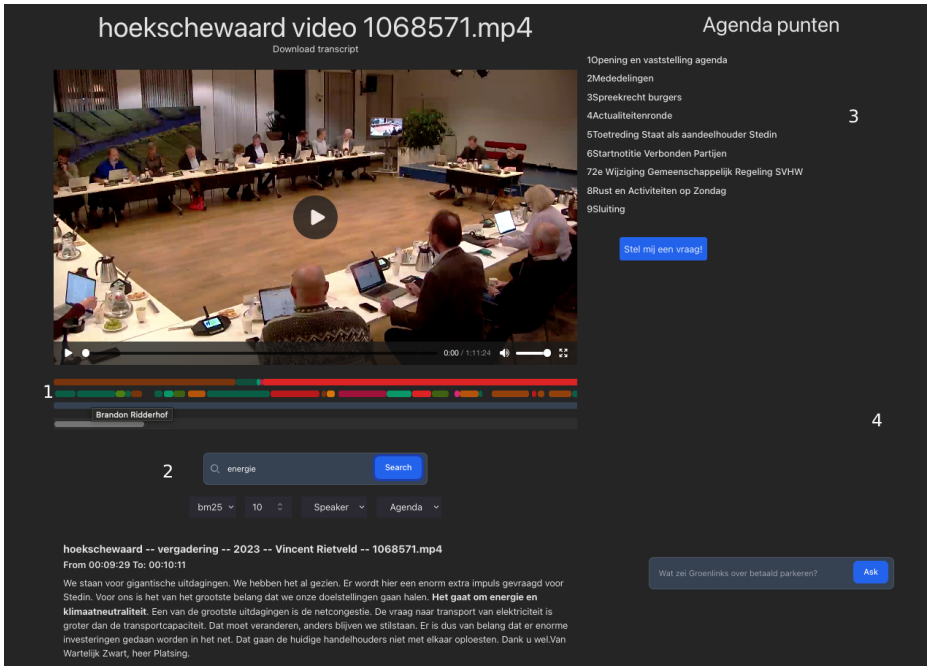

Fig. 1. Interface of the council meeting video spoken QA system.

Speaker recognition performance is generally benchmarked by four different metrics: *confusion rate*, the duration of speaking with an incorrect speaker, *false alarm rate*, the duration of non-speech incorrectly classified as speech, *missed detection rate*, the duration of speech that is incorrectly classified as non-speech and *diarisation error rate*, which is the ratio of the sum of these three and the duration of the ground truth. We manually annotated the above mentioned 40 minutes with speaker and speech ground truth. To account for systematic differences in the beginning and ending of speeches, we performed 3 tests using collars [9]: without collar and with a 250- and 500-millisecond collar. A 250 millisecond collar means that the 125 milliseconds before and after each speaker change is excluded from the evaluation. With a 500ms collar, the average (over the 7 meetings) confusion, alarm and missed detection rates were 0.9%, 3.74%, and 11.54%, respectively, resulting in an average diarisation error rate of 7%.

To determine the effectiveness of the developed search system, we gave a test group of 13 participants the task to answer a factoid question with an answer contained in the recorded council meeting. One of the questions was: *What are the expected advantages of the installation of green areas surrounding underground trash containers?* Each participant got five randomly selected questions combined with a unique search method or the traditional method of manually scrolling through the video file. If no answer could be found within three minutes, the search counts as a failure.

We found that without any search help, the average time to find the answer was 113 seconds, with a 69% miss rate. Vector search, BM25 and hybrid search led, on average, to an answer in 32, 31 and 41 seconds, respectively. Unlike answering the questions by

watching the video, all searches led to the correct answer, a miss rate of 0%. The chat functionality led to an answer even faster; in 14 seconds on average, but with a miss rate of 23%. Note that this average time includes time to formulate and type the question, as well as scanning the LLM's response.

4 Related work

IR on spoken documents is harder than IR on text because of two main reasons: imperfect ASR quality and the difficulty of browsing audio and video [1]. For ad hoc search, the first reason turned out not a problem after 4 years of TREC spoken document retrieval [6], basically because documents are long enough to have correctly recognized keyword matches. The second reason can be overcome with systems which do entry point retrieval, but these, just as QA systems, index much smaller passages, and thus the query-document mismatch plays up again [2,13]. Neural end to end systems starting with a spoken question bypass ASR and achieve good results on passage retrieval [8]. Because of the reported high ASR quality of OpenAI's Whisper [11], also for smaller languages, we wanted to see how well proven cheap non-neural IR methods faired with QA on spoken documents.

Acknowledgements This research was supported in part by the Netherlands Organization for Scientific Research (NWO) through the ACCESS project grant CISC.CC.016 and an Open Science Fund grant no. 01607400. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-5788.

References

1. Allan, J.: Perspectives on information retrieval and speech. In: Information Retrieval Techniques for Speech Applications. pp. 1–10. Springer (2002)
2. Amato, G., Bolettieri, P., Carrara, F., Falchi, F., Gennaro, C., Messina, N., Vadicamo, L., Vairo, C.: VISIONE: a large-scale video retrieval system with advanced search functionalities. In: Proc. ICMR '23. p. 649–653 (2023). https://doi.org/10.1145/3591106.3592226
3. Bredin, H., Yin, R., Coria, J.M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., Gill, M.P.: Pyannote.audio: neural building blocks for speaker diarization. In: Proc. ICASSP '20. pp. 7124–7128. IEEE (2020)
4. Cormack, G.V., Clarke, C.L.A., Buettcher, S.: Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In: Proc. SIGIR '09. p. 758–759. ACM (2009). https://doi.org/10.1145/1571941.1572114
5. Dilocker, E., van Luijt, B., Voorbach, B., Hasan, M.S., Rodriguez, A., Kulawiak, D.A., Antas, M., Duckworth, P.: Weaviate, https://github.com/weaviate/weaviate
6. Garofolo, J.S., Auzanne, C.G., Voorhees, E.M., et al.: The TREC spoken document retrieval track: A success story. In: Proc. TREC-8. NIST special publication 500-246 (2000)
7. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282 (2020)
8. Lin, C.J., Lin, G.T., Chuang, Y.S., Wu, W.L., Li, S.W., Mohamed, A., Lee, H.Y., Lee, L.S.: SpeechDPR: End-to-end spoken passage retrieval for open-domain spoken question answering. In: Proc. ICASSP '24. pp. 12476–12480 (2024). https://doi.org/10.1109/ICASSP48485.2024.10448210

9. McKnight, S.W., Hogg, A.O.T., Naylor, P.A.: Analysis of phonetic dependence of segmentation errors in speaker diarization. In: Proc. EUSIPCO '21. pp. 381–385 (2021). https://doi.org/10.23919/Eusipco47968.2020.9287552
10. Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., et al.: Teaching language models to support answers with verified quotes. arXiv preprint arXiv:2203.11147 (2022)
11. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: Proc ICML '23. pp. 28492–28518 (2023)
12. Sidiropoulos, G., Vakulenko, S., Kanoulas, E.: On the impact of speech recognition errors in passage retrieval for spoken question answering. In: Proc. CIKM '22. p. 4485–4489 (2022). https://doi.org/10.1145/3511808.3557662
13. Snoek, C.G.M., Worring, M.: Concept-based video retrieval. Foundations and Trends in Information Retrieval **2**(4), 215–322 (2009). https://doi.org/10.1561/1500000014
14. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: MPNet: Masked and permuted pre-training for language understanding (2020), https://arxiv.org/abs/2004.09297